UNI GRAZ
TU Graz
vetmeduni
FH JOANNEUM University of Applied Sciences

# Scientific Computing 2023

## Conference Proceedings

| | |
|---|---|
| Herausgeber | Wolfgang Granigg / FH JOANNEUM Gesellschaft mbH |
| Layout | Alina Konopasek, Sophia Kristl und Klaus Lichtenegger |
| Layout Umschlag | Sarah Bernard, Alina Konopasek und Sophia Kristl |
| Umschlagbilder | pixabay (https://pixabay.com/) |
| Druck | Druckerei Niegelhell GmbH |

Verlag der FH JOANNEUM Gesellschaft mbH

Alte Poststraße 149

A-8020 Graz

www.fh-joanneum.at

| | |
|---|---|
| ISBN Print: | 978-3-903318-19-9 |
| ISBN e-Book: | 978-3-903318-20-5 |

# Scientific Computing 2023 – Vorwort und Danksagung

**Scientific Computing** spielt eine **immer größere Rolle** in verschiedenen wissenschaftlichen Disziplinen, sei es beim Lösen komplexer Fragestellungen, beim Analysieren großer Datenmengen oder bei der Simulation von Experimenten.

Daher findet heuer zum ersten Mal eine **Konferenz**, die sich diesem immer relevanteren Thema widmet, statt: die **Scientific Computing 2023** ermöglicht es Nachwuchswissenschaftler:innen, die Ergebnisse ihrer Arbeiten in Form eines Papers oder Posters einzureichen und zu präsentieren. Die **Hauptthemen** der Konferenz umfassen:

- Computer Science & Big Data,
- Modellbildung & Simulation,
- Statistik & Machine Learning sowie
- Artificial Intelligence.

Wir haben uns über **viele Einreichungen** gefreut, die ein breites Spektrum von **physikalischen** Anwendungen bis hin zu **gesundheitlichen** und **technischen Themen** abdecken. Die eingereichten Beiträge wurden von einer **Fachjury** gewissenhaft und sorgfältig begutachtet.

Die Veranstaltung Scientific Computing 2023 findet an der **FH JOANNEUM** statt. Unser Dank gilt daher dem Management der FH JOANNEUM für das Zurverfügungstellen der **Räumlichkeiten** und **Ressourcen**. Die **Organisation** der Konferenz wurde von **vier Institutionen** übernommen. Wir bedanken uns für die unkomplizierte, pragmatische und fokussierte Zusammenarbeit bei den Verantwortlichen der **TU Graz**, der **Universität Graz** und der **Vetmeduni Wien**.

Konkret möchten wir uns insbesondere auch bei den folgenden Personen bedanken, die einerseits in der **Vernetzung der beteiligten Hochschulen** organisatorisch unterstützt haben und sich andererseits auch in die **Fachjury** der Scientific Computing 2023 unter dem Vorsitz von **Wolfgang Granigg** (FH JOANNEUM) eingebracht haben: **Horst Possegger** (TU Graz), **Georg Jäger** (Universität Graz) und **Peter Roth** (Vetmeduni Wien). Ein großer Dank gilt zudem auch den **übrigen Mitgliedern der Fachjury**: Rupert Beinhauer, Birgit Burböck, Sten Hanke, Elmar Krainz, Klaus Lichtenegger, Michael Melcher, Robert Mischak, Bernhard Neumayer, Joachim Schauer, Andreas Tramposch, Andreas Windisch und Wilhelm Zugaj.

Im Weiteren bedanken wir uns für die hochprofessionelle und stringente **Organisation** bei **Lisa Zimmermann** und **Alina Konopasek**. Hinsichtlich der Zusammenstellung des vorliegenden **Sammelbands** möchten wir uns insbesondere bei **Sophia Kristl** bedanken. Zusätzlich möchten wir ein großes **Danke** für die **Unterstützung** an folgende Personen aussprechen: Sarah Bernard, Wolfgang Dautermann, Werner Fritz, Manfred Füllsack, Gabriele Holosch, Markus Hopfer, Klaus Kral und Klaus Lichtenegger, sowie an alle, die sonst an der Organisation beteiligt waren.

Außerdem wäre die Veranstaltung ohne unsere **Sponsoren** nicht möglich gewesen, ein **herzliches Dankeschön** an: bitmedia, BOOM Software, dynatrace, Informationstechnik Graz, Merkur Innovation Lab, PASO Solutions, Raiffeisen Landesbank Steiermark, sappi und öh joanneum.

Genießen Sie **die Beiträge dieses Sammelbands** und sehen Sie wie spannend, vielfältig und faszinierend **Scientific Computing** sein kann!

MMMMag. DDr. Wolfgang Granigg
FH JOANNEUM University of Applied Sciences

Dr. Georg Jäger
Universität Graz

Dipl.-Ing. Dr. Horst Possegger
Technische Universität Graz

Univ.-Prof. Dipl.-Ing. Dr. Peter Roth
Veterinärmedizinische Universität Wien

# Scientific Computing 2023

## Conference Proceedings

# Inhaltsverzeichnis

## Paper-Track: Chemie ........ 2

Bringing Chemistry to Scale: Loss Weight Adjustment for Multivariate Regression in Deep Learning of Thermochemical Processes ........ 4

Classification of single cells by Raman spectroscopy and machine learning: comparison of common algorithms ........ 12

## Paper-Track: Gaming ........ 22

Development of an educational escape game for Virtual Reality, Augmented Reality and Mobile phone 24

Performance of Generative Adversarial Imitation Learning in Various Minihack Environments ........ 32

## Paper-Track: Gesundheit ........ 42

An Agent-based Framework for COVID-19 Simulation to Predict Outcome for the Federal State of Tyrol and Province South Tyrol ........ 44

Brain Tumor Detection in MRI Images with Transfer Learning from Pre-Trained CNN Models ........ 50

Machine learning in human healthcare: Optimizing computer-aided detection and diagnosis systems using an x-ray data set of pediatric wrist fractures ........ 60

## Paper-Track: Luftfahrt ........ 70

Design, Construction and Fluid Flow Calculation of a Centrifugal Compressor ........ 72

Development of a calculation algorithm for the power design of a vertical icing wind tunnel (IWT) ........ 82

Implementation and comparison of various algorithms for attitude and heading reference systems of drones in MATLAB® ........ 88

Numerical investigation of the effects of a winglet configuration on the lift and drag coefficients of the SD7062 wing profile ........ 100

Numerische Simulation von Tragflächenvereisungen ........ 110

Three-dimensional simulation of in-flight icing on an aerofoil using Ansys FENSAP-ICE ........ 114

## Paper-Track: Mathematik ........ 124

Exploring complex contagions on progressively sparse networks: A study of non-linear graph reduction ........ 126

## Paper-Track: Mobilität ........ 134

Can you trust in me? A qualitative literature review on validation techniques for multi-sensory environments in autonomous driving ........ 136

CFD based exploration of parametric 3D swirler ........ 142

Entwicklung und Evaluation einer virtuellen Verkehrssimulation in Kombination mit einem Expertensystem ........ 152

Finding the Optimum Design of Large Gas Engines Prechambers using CFD and Bayesian Optimization ........ 160

First development steps towards autonomous driving ........ 170

Using AI to transform Traffic Models into Mobility Models ........ 178

## Paper-Track: Wirtschaft ........ 188

Comparing different Sales Forecast approaches on real-world data using Machine Learning ........ 190

Image classification in the real world: Methodologies for hierarchical classification of customer provided images in e-commerce ........ 198

Tracking the Evolution of Climate Protection Discourse in Austrian Newspapers: A Comparative Study of BERTopic and Dynamic Top Modeling ........ 208

## Poster ........ 220

A Pony Farm's Combinatorial Use Case: Finding a Suitable Assignment by Using a SAT Solver ........ 222

Energiespeicher im privaten Sektor - eine nachhaltige Energielösung für Österreich? ........ 223

Extended Virtual Acrophobia Treatment ........ 224

Generating Satellite Deepfakes with AI: Using Handmade Sandscapes as Input and Projection Mapping for Output ........ 225

Hydrofoil Boat Design ........ 226

In-cabin Gaze Detection using Neural Networks on Mobile Devices ........ 227

Interactive Sandscape - explore the interactivity of height modeling by shaping your own landscape with kinetic sand ........ 229

Investigation of Process Influences on the Steel Composition and Inclusion Modification in Secondary Metallurgy using Diverse Statistical Learning and Data Mining Methods ........ 230

Landform Detection on Mars Using a U-Net Architecture ........ 231

Modeling Rebound Effects from a Shift to Electric Vehicles in Private Mobility in Styria ........ 232

Vorhersage von Radsatz-Reprofilierungen durch künstliche Intelligenz ........ 233

# Digitalisierung

## von **Schulen** und **Bildungseinrichtungen**

Wir schaffen Innovation und gleichzeitig Standardisierung unter dem Einsatz modernster Technologien, um digitale Produkte und ganzheitliche Lösungen für den gesamten Bildungssektor anzubieten.

Wir stehen für Education Technology und der Digitalisierung von Bildung. **Wir sind bit media.**

Schul-verwaltung

E-Testing

E-Learning

Mehr erfahren

bit media
member of eee group

**bit media education solutions GmbH** | Kärntner Straße 337, 8054 Graz
www.bitmedia.at | office@bitmedia.at | +43 316 28 66 60

# Digitale Lernlösungen

## für **Unternehmen** und **öffentliche Organisationen**

Wir sind einer der führenden E-Learning Full-Service-Anbieter im deutschsprachigen Raum. Mit unserer Erfahrung, Know-How und Kreativtät begleiten wir Sie bei der Digitalisierung Ihrer Aus- und Weiterbildung - praxisnah, ansprechend und auf Ihre Bedürfnisse anpassbar.

**Wir sind Ihr Partner für digitale Lernlösungen.**

Standard-Content

Individual-Content

Learning Management System

Mehr erfahren

:MIT

**M.I.T e-Solutions GmbH** | Kärntner Straße 337, 8054 Graz
office@mit-esolutions.at | www.mit-esolutions.at

# Scientific Computing 2023

## Track: Chemie

*Franz M. Rohrhofer, Stefan Posch, Clemens Gößnitzer, José M. García-Oliver und Bernhard C. Geiger:* **Bringing Chemistry to Scale: Loss Weight Adjustment for Multivariate Regression in Deep Learning of Thermochemical Processes** ........ 4

*Daniel Zimmermann, David Lilek, Nicholas Posch, Daniel-Ralph Hermann, Norbert Pytel, Birgit Herbinger und Katerina Prohaska:* **Classification of single cells by Raman spectroscopy and machine learning: comparison of common algorithms** ........ 12

# Bringing Chemistry to Scale: Loss Weight Adjustment for Multivariate Regression in Deep Learning of Thermochemical Processes

Franz M. Rohrhofer[1], Stefan Posch[2], Clemens Gößnitzer[2], José M. García-Oliver[3], and Bernhard C. Geiger[1]

[1]Know-Center GmbH – Research Center for Data-Driven Business & Big Data Analytics, Sandgasse 36/4, 8010 Graz, Austria
[2]LEC GmbH – Large Engines Competence Center, Inffeldgasse 19, 8010 Graz, Austria
[3]CMT Motores Térmicos - Universitat Politècnica de València, Camí de Vera, 46022 València, Spain

## Abstract

Flamelet models are widely used in computational fluid dynamics to simulate thermochemical processes in turbulent combustion. These models typically employ memory-expensive lookup tables that are predetermined and represent the combustion process to be simulated. Artificial neural networks (ANNs) offer a deep learning approach that can store this tabular data using a small number of network weights, potentially reducing the memory demands of complex simulations by orders of magnitude. However, ANNs with standard training losses often struggle with underrepresented targets in multivariate regression tasks, e.g., when learning minor species mass fractions as part of lookup tables. This paper seeks to improve the accuracy of an ANN when learning multiple species mass fractions of a hydrogen ($H_2$) combustion lookup table. We assess a simple, yet effective loss weight adjustment that outperforms the standard mean-squared error optimization and enables accurate learning of all species mass fractions, even of minor species where the standard optimization completely fails. Furthermore, we find that the loss weight adjustment leads to more balanced gradients in the network training, which explains its effectiveness.

## 1 Introduction

Driven by the urgent need for more efficient and less polluting combustion engines, understanding and modeling of thermochemical processes through computational fluid dynamics (CFD) has become an essential part in engineering. To cope with the huge computational demands of including detailed chemical reaction mechanisms in three-dimensional simulations, several approaches for simulating turbulent combustion have been developed over the past decades. Flamelet models are widely used in CFD and belong to a category of approaches which may utilize indexing of a predetermined lookup table to accelerate calculations during simulation runtime. These lookup tables contain all the necessary information of the thermochemical process and can take up several gigabytes of memory space, potentially hindering their applicability for complex simulations when resources are scarce.

With the recent upsurge of scientific deep learning (DL), the use of ANNs has become popular, since ANNs can store the information of lookup tables with a finite, possibly small, number of network weights. Several approaches have been introduced, and differ in the choice of input (control) variables and predicted (thermochemical) scalars of the lookup table. Recently, [MJDB23] has successfully demonstrated the use of an ANN-based approach in the simulation of a turbulent premixed flame, while reducing the memory size of the flamelet table by around 92%. Furthermore, [HKH+21] also employed an ANN model for simulating water-sprayed turbulent combustion and reported reduced memory demands, while achieving a comparable accuracy to that of a conventional flamelet-generated manifold (FGM) approach.

However, a common challenge in DL for tabulating lookup tables is simultaneously learning and predicting multiple species mass fractions with a single ANN. ANNs performing a multivariate regression task often struggle to accurately predicting minor species, whose target mass fractions are much smaller than those of major species. This issue was explicitly reported in [OKA+20] and [DRJ22], and is usually addressed by learning smaller groups of species with similar orders of magnitude of mass fraction. Such clustering approach is followed, e.g., in [DRRJ21] by the multiple multilayer perceptron (MMP) approach, in [OKP21] by the mixture of experts (MoE) approach or in [RLLE21] by self-organizing maps (SOM). Overall, these methods are effective in learning species mass fractions of lookup tables, but they often come with an increased computational expense since multiple ANN instances have to be trained.

This work aims at improving the overall accuracy of a single ANN which is trained on multiple species mass fractions of thermochemical processes. To cope with the different ranges of target values, we employ a simple loss weight adjustment and test this approach in learning a lookup table of $H_2$ combustion which involves nine species. This system is chosen due to the current interest for combustion applications. We compare the accuracy of the loss weight adjustment with the use of standard mean-squared error (MSE) optimization. We study backpropagated gradients in the optimization and observe that the loss weight adjustment leads to more balanced gradients, which explains its effectiveness in learning multiple species with different target mass fractions.

# 2 Background

## 2.1 Flamelet Model

The flamelet model, initially derived by Peters [Pet88], assumes that a turbulent flame is an ensemble of laminar one-dimensional reacting structures, referred to as 'flamelets'. Flamelets are solved externally in ad-hoc solvers, from which the thermochemical state in terms of temperature $T$ and species mass fraction $Y_j$ is retrieved and later fed into the CFD code. This approach overcomes the need to solve a transport equation for each species in every cell of the computational domain. Some flamelet methods are tabulated to allow for accelerated CFD-flamelet interactions during simulation run-time. Depending on the flame configuration, different approaches may be followed; for example, in the current approach an Unsteady Flamelet Progress Variable (UFPV) tabulation depending on mixture fraction $Z$ (local fuel-air ratio), progress variable $C$ (chemical evolution from inert to steady conditions) and scalar dissipation rate $\chi$ (related to the spatial gradient along the 1D flamelet domain) is followed [NNPW15].

## 2.2 Artificial Neural Networks

ANNs are the 'work horse' of deep learning, due to their ability of approximating any continuous function given sufficient expressive power (universal approximation theorem). Their flexibility has rendered ANNs suitable for substituting lookup tables, i.e., to learn the nonlinear mapping $\boldsymbol{X} \mapsto \boldsymbol{Y}$, with $\boldsymbol{X} \in \mathbb{R}^D$ denoting the input control variables and $\boldsymbol{Y} \in \mathbb{R}^d$ the approximated thermochemical scalars, in this work the species mass fractions $Y_j$.

### 2.2.1 Deep Multilayer Perceptron

Multilayer perceptrons (MLPs) are a simple class of feedforward ANNs with multiple layers of perceptrons (or neurons) which are fully-connected. For a fully-connected MLP with $L$ layers, the network's output is given by the recursive application of activations

$$\boldsymbol{Y} = f_L(\boldsymbol{W}_L, \cdots, f_2(\boldsymbol{W_2}, f_1(\boldsymbol{W}_1, \boldsymbol{X}))\cdots), \quad (1)$$

where $\boldsymbol{W}_l$ contains the weights and biases $\theta_{i,l}$ of layer $l$, and $f_l$ denotes the neuron's activation function of that layer. The activation functions are (except for the final layer) nonlinear where common choices for regression tasks are the hyperbolic tangent (tanh) or Sigmoid linear unit (swish). For simplicity, we denote the weights and biases of the full network function with a single weight vector $\boldsymbol{\theta}$.

### 2.2.2 Loss Function & Optimization

To train the ANN on a specific task, the selection of an appropriate loss function is necessary. A common choice for regression tasks in DL is the well-known MSE loss

$$\mathcal{L}(\boldsymbol{\theta}) = \frac{1}{N}\sum_{i=1}^{N}\left(\boldsymbol{Y}^{(i)} - \boldsymbol{Y}_{\boldsymbol{\theta}}^{(i)}\right)^2, \quad (2)$$

with training examples $\left\{(\boldsymbol{X}^{(i)}, \boldsymbol{Y}^{(i)})\right\}_{i=1}^{N}$ and $\boldsymbol{Y_\theta}$ denoting the network's output. The optimization, i.e., finding the network weights that minimize the loss function, is performed in an iterative (gradient-based) process and according to the general update rule

$$\boldsymbol{\theta}_{n+1} \leftarrow \boldsymbol{\theta}_n - \alpha\frac{\partial\mathcal{L}}{\partial\boldsymbol{\theta}}, \quad (3)$$

where $\alpha$ denotes the learning rate. The gradients of the loss function w.r.t. the network weights can be calculated via the backpropagation algorithm that is part of modern deep learning libraries.

## 3 Methods

This section provides details on the loss weight adjustment that will be used in our experiments. We start by splitting the multivariate MSE loss from Equation (2) into the individual species mass fractions $\boldsymbol{Y} = (Y_0, Y_1, \ldots, Y_d)^T$

$$\mathcal{L}(\boldsymbol{\theta}) = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{d}\left(Y_j^{(i)} - Y_{j,\boldsymbol{\theta}}^{(i)}\right)^2, \quad (4)$$

where $Y_{j,\boldsymbol{\theta}}$ is the network's output of the $j$-th species mass fraction. Furthermore, we can write

$$\mathcal{L}(\boldsymbol{\theta}) = \sum_{j=1}^{d}\mathcal{L}_j(\boldsymbol{\theta}), \quad (5)$$

where $\mathcal{L}_j$ is the univariate MSE, cf. Equation (2), for each individual species mass fraction $Y_j$. From this perspective, the use of the MSE optimization in multivariate regression can be seen as a single objective by (linear) scalarization of the multiple objectives, where each objective is given by the optimization of a single target species mass fraction with a weight equal to one. Equations (2), (4) and (5) denote the same loss function which we will refer to as *standard* MSE for the later course of this work. According to Equation (5), the gradient update rule can be written in the form

$$\boldsymbol{\theta}_{n+1} \leftarrow \boldsymbol{\theta}_n - \alpha\sum_{j=1}^{d}\frac{\partial\mathcal{L}_j}{\partial\boldsymbol{\theta}}, \quad (6)$$

which implies that different orders of loss magnitudes yield different contributions to the gradient update. This can have serious consequences for the final training outcome as it will be demonstrated in the later part of this work.

### 3.1 Loss Weight Adjustment

In general, loss weights are commonly used in DL, ranging from Lagrange multipliers that adjust the strength of $L_1$ (lasso) or $L_2$ (ridge) regularization, to hyperparameters that adjust the importance of different objectives in physics-informed neural networks [RPG21] or variational autoencoders [KW$^+$19]. For our porpose, we note that the MSE loss is bound to absolute ranges of targets. Hence, by assuming that the ANN induces equal relative errors across the different species mass fractions, the standard MSE and respective gradients are potentially dominated by species with larger mass fractions. To compensate this, we consider weighting the individual components in Equation (5) according to

$$\mathcal{L}(\boldsymbol{\theta}) = \sum_{j=1}^{d} \omega_j \mathcal{L}_j(\boldsymbol{\theta}), \tag{7}$$

where $\omega_j$ is a scalar weight. Thus, individual gradients in the update rule are also scaled by

$$\boldsymbol{\theta}_{n+1} \leftarrow \boldsymbol{\theta}_n - \alpha \sum_{j=1}^{d} \omega_j \frac{\partial \mathcal{L}_j}{\partial \boldsymbol{\theta}}. \tag{8}$$

The choice of appropriate loss weights $\omega_j$ is crucial for the success of this approach. In this work, we choose the variance as a measure of dispersion, to determine how far individual mass fractions $Y_j$ are spread out from their average value $\bar{Y}_j$

$$\omega_j := \frac{1}{\mathrm{Var}(Y_j)}, \tag{9}$$

with $\mathrm{Var}(Y_j) = \sum_{i=1}^{N}(Y_j^{(i)} - \bar{Y}_j^{(i)})^2/N$. This effectively assigns greater loss weights to minor species with smaller mass fractions compared to those with larger mass fractions. Furthermore, determining the variance of species mass fractions has to be performed only once prior network training which is a computationally cheap and robust method for selecting adequate loss weights. The weights can be applied either directly at the loss computation in Equation (7) or in the gradient update rule in Equation (8) using the standard MSE. Both methods effectively scale the gradients in the final optimization step and will be later on referred to as *weighted* MSE optimization.

# 4 Experimental Setting

## 4.1 Datasets

For our experiments we use a single databases, representing the lookup table for $H_2$ combustion, as computed with the flamelet method discussed in Section 2.1. The database contains 6.7 million data points with nine species mass fractions. Details on the involved species can be taken from Table 1. We choose as input the progress variable $C$, the mixture fraction $Z$ and scalar dissipation rate $\chi$ and approximate $\boldsymbol{Y}(C, Z, \chi)$ with a single ANN where $\boldsymbol{Y} \in \mathbb{R}^9$. A 80/10/10 dataset split is used where 80% of the database is taken for training, and 10% for validation and testing, respectively.

## 4.2 Network Architecture & Optimization

We use fully-connected ANNs with four hidden layers, 50 neurons per layer and tanh activation function. For the output neurons, we use a softmax logistic layer to account for mass conservation in the species mass fractions, since $\sum_j Y_j = 1$. The network weights are initialized using Glorot initialization and inputs are scaled using MinMax feature scaling. The ANNs are optimized with stochastic gradient-descent on mini-batches of size 1024, i.e., the loss function for the standard or weighted optimization is computed over 1024 consecutive training points. We choose a learning rate of $\alpha = 0.001$ and perform 50 iterations (epochs) through the entire training dataset.

# 5 Results

Our experiments are arranged as follows: first, in Section 5.1 we train and test a single ANN on the $H_2$ database by using the standard MSE optimization as discussed in Section 2.2.2. Next, the loss weight adjustment, as introduced in Section 3.1, is applied and results are presented in Section 5.2. Finally, in Section 5.3 we extend and compare the two optimization approaches on different splits of training and validation set. We measure the final accuracy in terms of coefficient of determination

$$R^2 = 1 - \frac{\sum_i^N \left(Y_j^{(i)} - Y_{j,\boldsymbol{\theta}}^{(i)}\right)^2}{\sum_i^N \left(Y_j^{(i)} - \bar{Y}_j\right)^2}, \tag{10}$$

for each species mass fraction $Y_j$ on the test set, and use the validation set to record the MSE of each individual species and backpropagated gradients during optimization.

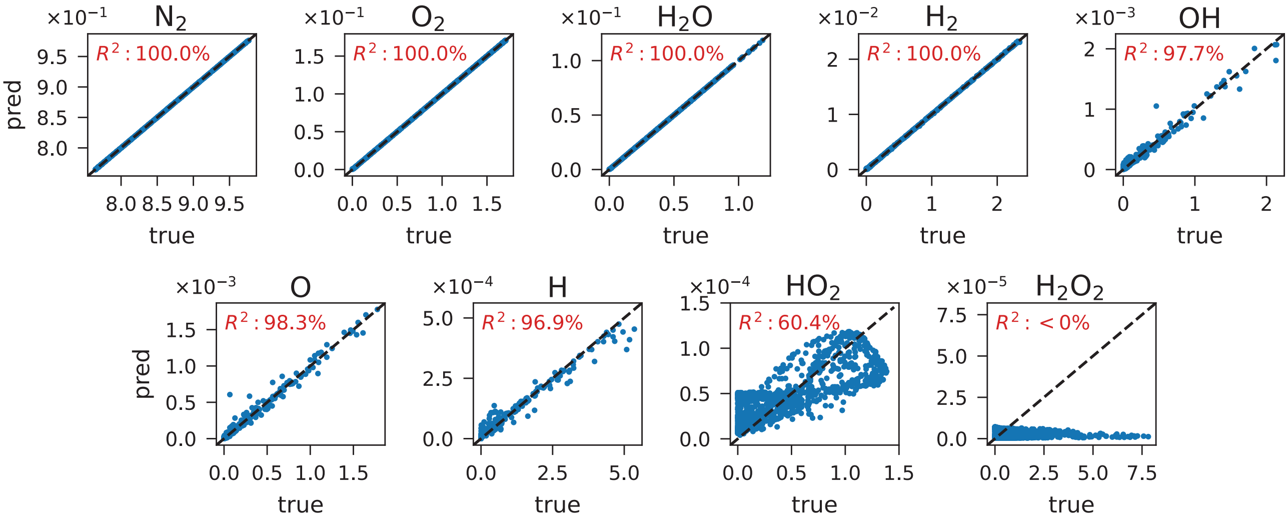


Figure 1: **Standard MSE Optimization.** True vs. predicted species mass fractions for the nine species in the $H_2$ test set. A perfect fit is indicated by the diagonal, dashed line. The accuracy in terms of the $R^2$ score is shown in red. Minor species, such as $HO_2$ or $H_2O_2$, are not learned accurately.

## 5.1 Standard MSE Optimization

For the first experiment, we train a single ANN on the $H_2$ database using the standard MSE optimization. Figure 1 shows the true versus predicted species mass fractions of the nine species in the $H_2$ database, evaluated on the test set. A perfect fit is indicated by the diagonal, dashed line and the accuracy (in terms of $R^2$) is given in the upper left corner. We note that the order of species is based on absolute ranges of species mass fractions, decreasing from left to right and top to bottom. From the figure it is evident that, while major species such as $N_2$, $O_2$, $H_2O$ and $H_2$ are predicted well ($R^2 > 99\%$), the accuracy drops considerably for minor species, in particular for $H_2O_2$ with a $R^2$ value below zero. To investigate this issue further, we refer to Figure 2 which shows the individual losses and backpropagated gradients, evaluated on the validation set during the optimization. For simplicity, we use the standard deviation of backpropagated gradients as a general measure of gradient information that updates the network weights. In Figure 2b, we observe unbalanced gradients across the different species, with minor species having significant lower gradients than major species. The discrepancy originates from the spread in the individual losses where, as already discussed in Section 3, the contribution from major species to the loss and gradient update is significantly larger. This explains why mass fractions of major species in Figure 1 are learned more accurately than that of minor species, even when the network size or learning rate is varied (also tested but not shown).

## 5.2 Weighted MSE Optimization

We now perform the same training and testing procedure as before, but this time apply the loss weight adjustment explained in Section 3.1. Figure 4 shows the true versus predicted species mass fractions of the nine species in the $H_2$ database, evaluated on the test set. We observe a nearly perfect fit ($R^2 \approx 100\%$) of all species mass fractions, even for minor species. Figure 3 shows the individual losses and backpropagated gradients, evaluated on the validation set during the optimization. By looking at the individual losses (Figure 3a), it is evident that, in comparison to the previous results, there is an equal and

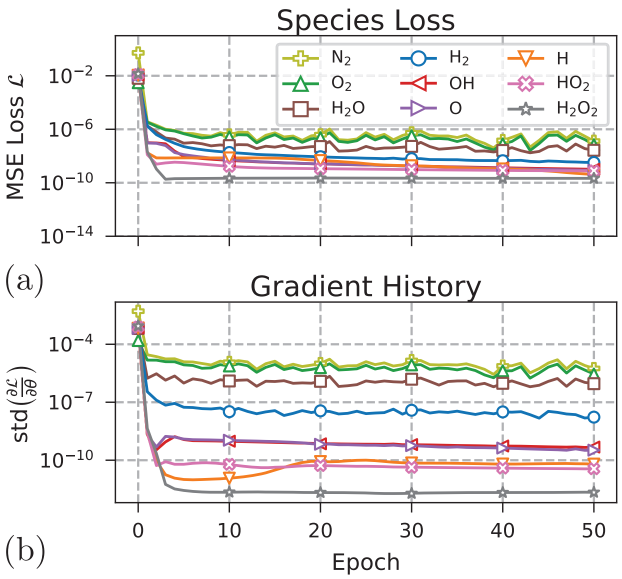


Figure 2: **Standard MSE Optimization.** Recorded (a) species loss and (b) backpropagated gradients for each of the nine species in the $H_2$ database. The unbalanced gradients during optimization explain the issue in learning the minor species, in particular $H_2O_2$ (cf Figure 1).

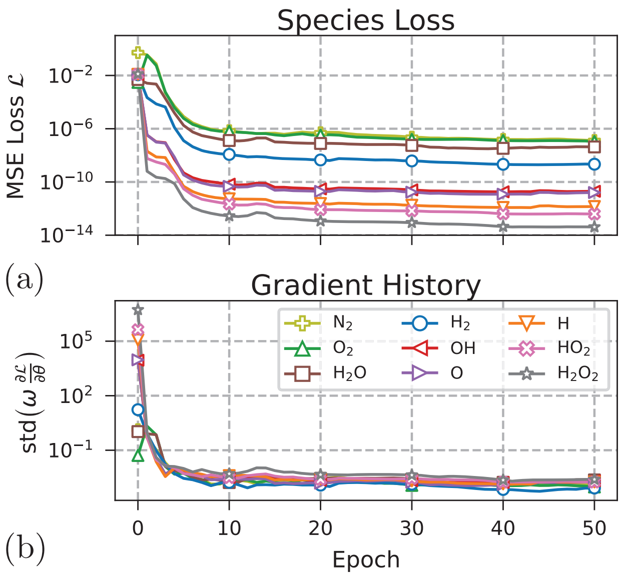


Figure 3: **Weighted MSE Optimization.** Recorded (a) species loss and (b) backpropagated gradients for each of the nine species in the $H_2$ database. The balanced gradients during optimization explain the success in a equal and continuous learning of all species (cf Figure 4).

continuous improvement across all species. This can be explained by Figure 3b which shows that backpropagated gradients are mostly in balance during optimization and, hence, account for an equal and continuous learning of all species mass fractions. We note that although gradients start at a much different range when the loss weight adjustment is applied, the same initial learning rate of $\alpha = 0.001$ still works well as a decent step size that results in balanced gradients during optimization.

### 5.3 Comparison & Summary

While the previous results have demonstrated the effectiveness of the loss weight adjustment on a single experimental setup, it remains unclear whether this can be extended to the general case, especially on different dataset splits. To this end, we train and test multiple networks. We record the average accuracy over ten different ANN instances, each uses a unique seed for the initialization of network weights, and split of training and validation set. We optimize them with and without the loss weight adjustment.

The results can be found in Table 1. In the table header, each species and the respective variance of its mass fraction $\mathrm{Var}(Y_j)$ is listed. The main table shows the mean and standard deviation of the $R^2$ score over the ten individual runs as determined on the test set. $R^2$ scores below zero are listed as $< 0$. From the table it is evident that the use of the loss weight adjustment vastly outperforms the standard optimization in accurately learning all species, in particular minor species mass fractions where the standard optimization completely fails. In summary, these results demonstrate the effectiveness of the loss weight adjustment which resolves issues in learning species mass fractions on different scales.

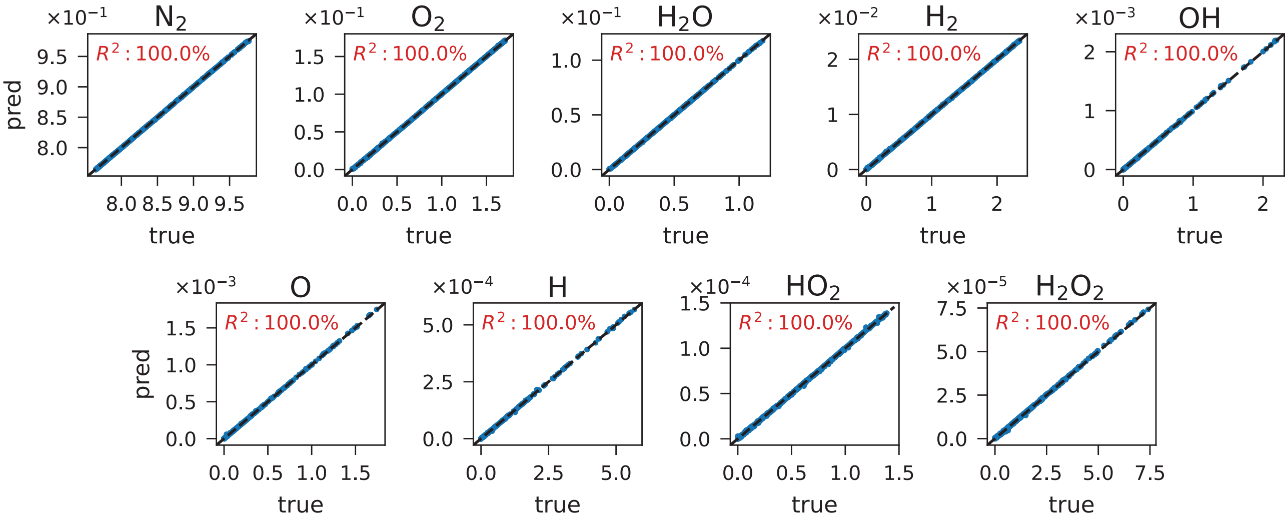


Figure 4: **Weighted MSE Optimization.** True vs. predicted species mass fractions for the nine species in the $H_2$ test set. A perfect fit is indicated by the diagonal, dashed line. The accuracy in terms of the $R^2$ score is shown in red. All species in the database are learned perfectly ($R^2 \approx 100$), even minor ones.

Table 1: ANN performance ($R^2$, mean and standard deviation over ten uniquely trained instances) on the nine species of the $H_2$ database. $R^2$ scores below zero are listed as $< 0$. Significant improvement on minor species is achieved when the loss weight adjustment is applied.

| **Specie** | $\mathbf{N_2}$ | $\mathbf{O_2}$ | $\mathbf{H_2O}$ | $\mathbf{H_2}$ | **OH** | **O** | **H** | $\mathbf{HO_2}$ | $\mathbf{H_2O_2}$ |
|---|---|---|---|---|---|---|---|---|---|
| **Var($\boldsymbol{Y_j}$)** | $6.22 \cdot 10^{-2}$ | $5.48 \cdot 10^{-2}$ | $2.18 \cdot 10^{-2}$ | $6.73 \cdot 10^{-3}$ | $2.97 \cdot 10^{-4}$ | $2.87 \cdot 10^{-4}$ | $8.07 \cdot 10^{-5}$ | $4.33 \cdot 10^{-5}$ | $1.24 \cdot 10^{-5}$ |
| **standard** | $100.00 \pm 0.01$ | $100.00 \pm 0.01$ | $99.99 \pm 0.01$ | $99.99 \pm 0.01$ | $98.85 \pm 0.30$ | $99.00 \pm 0.29$ | $93.59 \pm 5.69$ | $58.42 \pm 2.98$ | $< 0$ |
| **weighted** | $100.00 \pm 0.01$ | $100.00 \pm 0.01$ | $99.99 \pm 0.01$ | $100.00 \pm 0.01$ | $99.98 \pm 0.02$ | $99.98 \pm 0.02$ | $99.98 \pm 0.01$ | $99.98 \pm 0.01$ | $99.97 \pm 0.01$ |

# 6 Conclusion

The use of ANNs for replacing lookup tables of flamelet models has become a prominent tool to cope with the huge computational expenses of turbulent combustion modeling. Training ANNs on such tabular data usually involves learning multiple species mass fractions with different characteristics. This work discussed a simple, yet effective loss weight adjustment that balances the loss gradients of different species and brings the ANN optimization to scale. This significantly improved the overall accuracy of a single ANN, performing a multivariate regression task on several species with different mass fractions. Future work will be devoted to extending this approach to more complex systems. In general, our results provide valuable insights into using ANNs for multivariate regression tasks and may find further use in several disciplines beyond thermochemical processes, such as multiscale and multiphysics modelling.

# Acknowledgments

The authors acknowledge the financial support of the Austrian COMET — Competence Centers for Excellent Technologies — Programme of the Austrian Federal Ministry for Climate Action, Environment, Energy, Mobility, Innovation and Technology, the Austrian Federal Ministry for Digital and Economic Affairs, and the States of Styria, Upper Austria, Tyrol, and Vienna for the COMET Centers Know-Center and LEC EvoLET, respectively. The COMET Programme is managed by the Austrian Research Promotion Agency (FFG). Research activity by José M García-Oliver was partially funded by ORIONe project (PDC2021-121066-C22) from the Agencia Estatal de Investigación of the Spanish Government.

## References

[DRJ22] Tianjie Ding, Stelios Rigopoulos, and WP Jones. Machine learning tabulation of thermochemistry of fuel blends. *Applications in Energy and Combustion Science*, 12:100086, 2022.

[DRRJ21] Tianjie Ding, Thomas Readshaw, Stelios Rigopoulos, and WP Jones. Machine learning tabulation of thermochemistry in turbulent combustion: An approach based on hybrid flamelet/random data and multiple multilayer perceptrons. *Combustion and Flame*, 231:111493, 2021.

[HKH+21] Takafumi Honzawa, Reo Kai, Kotaro Hori, Makoto Seino, Takayuki Nishiie, and Ryoichi Kurose. Experimental and numerical study of water sprayed turbulent combustion: Proposal of a neural network modeling for five-dimensional flamelet approach. *Energy and AI*, 5:100076, 2021.

[KW+19] Diederik P Kingma, Max Welling, et al. An introduction to variational autoencoders. *Foundations and Trends® in Machine Learning*, 12(4):307–392, 2019.

[MJDB23] Arash Mousemi, Mehdi Jadidi, Seth B Dworkin, and W Kendal Bushe. Application of machine learning in low-order manifold representation of chemistry in turbulent flames. *Combustion Theory and Modelling*, 27(1):83–102, 2023.

[NNPW15] Bertrand Naud, Ricardo Novella, José Manuel Pastor, and Johannes F Winklinger. Rans modelling of a lifted h2/n2 flame using an unsteady flamelet progress variable approach with presumed pdf. *Combustion and Flame*, 162(4):893–906, 2015.

[OKA+20] Opeoluwa Owoyele, Prithwish Kundu, Muhsin M Ameen, Tarek Echekki, and Sibendu Som. Application of deep artificial neural networks to multi-dimensional flamelet libraries and spray flames. *International Journal of Engine Research*, 21(1):151–168, 2020.

[OKP21] Opeoluwa Owoyele, Prithwish Kundu, and Pinaki Pal. Efficient bifurcation and tabulation of multi-dimensional combustion manifolds using deep mixture of experts: An a priori study. *Proceedings of the Combustion Institute*, 38(4):5889–5896, 2021.

[Pet88] Norbert Peters. Laminar flamelet concepts in turbulent combustion. In *Symposium (International) on combustion*, volume 21, pages 1231–1250. Elsevier, 1988.

[RLLE21] Rishikesh Ranade, Genong Li, Shaoping Li, and Tarek Echekki. An efficient machine-learning approach for pdf tabulation in turbulent combustion closure. *Combustion Science and Technology*, 193(7):1258–1277, 2021.

[RPG21] Franz M Rohrhofer, Stefan Posch, and Bernhard C Geiger. On the pareto front of physics-informed neural networks. *arXiv preprint arXiv:2105.00862*, 2021.

# Classification of single cells by Raman spectroscopy and machine learning: comparison of common algorithms

Daniel Zimmermann, David Lilek, Nicholas Posch, Daniel-Ralph Hermann, Norbert Pytel, Birgit Herbinger, and Katerina Prohaska

Austrian Biotech University of Applied Sciences, Konrad Lorenz-Straße 10, 3430 Tulln, Austria.
E-mail: prohaska@tulln.fhwn.ac.at

15.02.2023

## Abstract

Surface-enhanced Raman spectroscopy (SERS) with in situ or colloidal produced metallic nanoparticles is a powerful tool for fast and reliable measurement of pro- and eukaryotic cells. Recently, much research has been focused on the use of a wide range of machine learning methods to detect minute differences in the spectra of various groups of cells. One aspect which is often neglected in current literature is the *No Free Lunch Theorem*, which states that one cannot assume *a priori* that a specific method is appropriate for a given task. Accordingly, we compare a wide range of commonly used linear and nonlinear classification methods applied to three biological datasets (prokaryotic cells, cyanobacteria, and Hodgkin lymphoma cells) and assess their performance and interpretability. Results demonstrate that while a pre-selection based on the size of the dataset and the complexity of the classification task at hand, comparing several algorithms is still crucial to achieving optimal performance. For this purpose, the Python tool developed herein was made to be easily adaptable for different data and classification methods.

## 1 Introduction

Over the last two decades, Raman spectroscopy has gained significant popularity as a fast and nondestructive analytical tool in many areas of research, ranging from material science and environmental analysis to food safety and medicine [Pro16]. In conjunction with the amplification of the intrinsically weak Raman signals through surface-enhanced Raman spectroscopy (SERS), chemical fingerprinting data can be collected for a wide range of samples [Sch14]. Furthermore, the relatively short excitation wavelengths used in Raman spectroscopy enable spatially resolved measurements when coupled with an optical microscope, for instance of individuals cells.

Previously, SERS has been investigated as a monitoring tool for polyhydroxy butyrate (PHB) production processes in cyanobacteria [HLD+20]. More subtle differences caused by chemotherapeutic treatment of Hodgkin lymphoma cells were also identified by SERS [Zim22].

Nevertheless, SERS also has some significant limitations, especially the low level of reproducibility of the SERS enhancement. Furthermore, the overwhelming dominance of amino and nucleic acids in spectra of biological samples creates additional challenges for the implementation of SERS as a routine method [CPS+14]. Another common issue when analyzing biological samples is background fluorescence which can often drown out the weak Raman signal, although this is less of a problem in SERS measurements. As a result of these challenges, extracting useful information from SERS spectra remains a difficult task, especially for complex biological samples. To detect the often minute differences between spectra of different classes, a wide range of machine learning (ML) methods has been applied in recent years [RL20].

Despite the numerous advantages provided by these techniques, it is crucial to acknowledge the fundamental principle of the *No Free Lunch Theorem*, which asserts that there can be no single optimal algorithm for all possible learning tasks [Wol96]. As a result, comparing different methods is necessary for achieving optimal classification performance. Such a selection process is however often neglected – or at least not explicitly discussed – in current Raman spectroscopic literature [TLY$^+$21].

A wide range of classification algorithms are available and regularly used for Raman spectroscopic data. In many cases, relatively simple linear models are entirely sufficient. Linear Discriminant Analysis (LDA) – usually combined with Principal Component Analysis (PCA) for dimensionality reduction – is commonly used, although other linear models such as logistic regression and (linear) support vector machines (SVMs) are also employed [PZD$^+$21]. In recent years, nonlinear methods including random forests, kernel SVMs and artificial neural networks (ANNs) have been increasingly applied to Raman spectroscopic data to identify minute differences in complex biological samples [LTC$^+$20].

As one of the most commonly used techniques for linear classification of Raman spectra, we selected PCA-LDA as the starting point for this comparison. We first explored other linear methods, starting with alternative dimensionality reduction techniques such as non-negative matrix factorization (NMF) [LS99], feature agglomeration/clustering (FA), and feature selection via peak detection. With logistic regression and SVMs we then examined other linear classifiers which do not require a separate dimensionality reduction step and compared different regularization strategies (using the $\ell_1$ or $\ell_2$ norms) to optimize performance and interpretability. For nonlinear models, we focused on decision trees and two types of tree ensembles, random forests [FGE14] and gradient-boosted decision trees (GBDT) [NK13]. Other nonlinear techniques such as SVMs and ANNs were considered but ultimately not used due to their poor interpretability and (especially for ANNs) large number of samples required.

Three biological datasets were analyzed using the presented workflow, the first two of which have already undergone simple exploratory data analysis in previous works:

- Prokaryote dataset: differentiation between cells of *E. coli* and *Paenibacillus polymyxa*, with the aim of detecting bacterial species in hospital sewage [Ste18]
- Cyanobacteria: detection of stored polyhydroxy butyrate (PHB), which is of interest for the biopolymer industry as biodegradable plastic [RGD$^+$20, HLD$^+$20]
- Hodgkin lymphoma (HL) cells: determining the effect of cytostatic treatment compared to an untreated control [Zim22]

These datasets follow a gradient of increasing complexity. As such, the classification of bacterial species is expected to be relatively easily achievable due to differences in metabolism and cell structure, while PHB production in a single cyanobacterial species may be more difficult to accomplish. In contrast, human HL cells undergoing cytostatic treatment present a more complex analysis environment, and the differences introduced by the treatment are likely more subtle.

# 2 Methods

## 2.1 Cell culture

**Prokaryotes dataset** *E. coli* (strain: DSM498) and *Paenibacillus polymyxa* (strain: CCI-25CAIT) were cultivated overnight on LB/M9 and GSC medium respectively, as described in [Ste18]. Cultures were incubated at 37 °C for *E. coli* and 30 °C for *P. polymyxa*. After washing with deionized water, the samples were spotted onto $CaF_2$ slides and allowed to dry at room temperature. 149 spectra of *E. coli* and 77 spectra of *P. polymyxa* were collected.

**Cyanobacteria dataset** *Synechocystis* sp. PCC6803 was grown in variations of the common BG11 medium. For a low intracellular PHB level the full medium was used, while nitrogen- and phosphorous-limited conditions resulted in PHB being stored as an energy reserve. Signal enhancement was achieved with *in situ*-reduced Ag-colloids as described in [HLD+20]. The dataset consists of 50 spectra per PHB level (low/high).

**HL dataset** Three biological replicates of the HL cell line L-540 [DKS+81] were cultivated using 80 % RPMI-1640 and 20 % fetal calf serum (FCS). Additionally, 1 % each of sodium pyruvate and Penicillin/Streptomycin solutions were added. After 72 h initial incubation, 25 µg/mL Etoposide was added to the treatment group. The control group was cultivated without Etoposide. After a further 24 h the cells were harvested and measured using colloidal gold nanoparticles as described in [BHCR+16]. For each replicate at least 100 spectra were recorded, giving a total of 300 spectra per group. [Zim22]

SERS measurements were carried out on the Senterra I Raman micro-spectroscope (Bruker, Billerica) using a 50x objective and $CaF_2$ slides (Crystran unlimited, Poole). Measurement parameters for each dataset are listed in table 1.

Table 1: Main parameters for Raman data acquisition of each dataset using the Senterra I Raman micro-spectroscope.

| | Prok. | Cyano. | HL |
|---|---|---|---|
| Laser [nm] | 532 | 785 | 785 |
| Laser power [mW] | 20 | 10 | 25 |
| Aperture [µm] | 50 × 1000 | 25 × 1000 | 50 × 1000 |
| Resolution [$cm^{-1}$] | 9–15 | 9–15 | 3–5 |
| Spectral range [$cm^{-1}$] | 60–4450 | 90–3500 | 440–1810 |
| Integration time [s] | 10 | 1 | 5 |
| Coadditions | 2 | 5 | 4 |

## 2.2 Data preprocessing

After data acquisition, the first step to ensure data quality is to remove spectra of low intensity and peak count. We used a self-developed, automated tool to assess the quality of the Raman spectra [ZLPH21]. After the quality control step 50 spectra from the Cyanobacteria and Prokaryotes datasets and 300 spectra from the HL dataset were retained per class. Background fluorescence was removed by baseline correction via the asymmetric least squares algorithm, after which measurement noise was smoothed using a $3^{rd}$ order Savitzky-Golay filter with a window size of 15. Finally, vector normalization was applied to compensate variations in overall intensity [Zim22].

## 2.3 Cross-validation

To ensure an unbiased estimate of model performance while still making full use of the limited available data, we employed a nested 5-fold cross-validation (CV) scheme. The outer CV was used to obtain performance estimates on independent test sets. On the other hand, the inner CV optimized the various hyperparameters which control model complexity and therefore affect over- and underfitting. This nested CV was applied to each of the examined classification methods, which are shown in figure 1.

For PCA-LDA and NMF-LDA, complexity was controlled via the number of components that was passed on to LDA. Similarly, FA-LDA utilized the allowed number of clusters for complexity control. In all three cases, a higher value resulted in a more complex model. In contrast, for feature selection via peak detection (Peak-LDA) the minimum distance between neighboring peaks was optimized, with a larger distance leading to a less complex model.

Logistic regression and SVMs are both controlled by the regularization parameter $C$, which governs the relative importance of the selected norm ($\ell_1$ or $\ell_2$) as a penalty term in the cost function. A higher value for $C$ leads to stronger regularization, resulting in a less complex model.

Table 2: Model hyperparameters for each tested classification method and dataset. The minimum and maximum values are given. C: regularization strength of $\ell_1$ and $\ell_2$ regularization; $\alpha$: complexity parameter of minimal cost-complexity-pruning in the decision tree model; Feature Subsample: Randomly chosen fraction of features used for each tree in a random forest model

| Model | Parameter | Prokaryotes | Cyanobacteria | Hodgkin Lymphoma |
|---|---|---|---|---|
| PCA–LDA | No. Components | 1–20 | 1– 20 | 1–50 |
| NMF–LDA | No. Components | 2–20 | 2– 20 | 5–40 |
| FA–LDA | No. Clusters | 5–40 | 5– 40 | 5–40 |
| PeakPicking–LDA | Min. Distance | 10–150 | 10– 150 | 10–150 |
| LogReg $\ell_1$ | C | $10^{-1}$–$10^{2}$ | $10^{-1}$– $10^{2}$ | $10^{-2}$–$10^{1}$ |
| LogReg $\ell_2$ | C | $10^{-4}$–$10^{1}$ | $10^{-4}$– $10^{1}$ | $10^{-5}$–$10^{-1}$ |
| Linear SVM $\ell_1$ | C | $10^{-2}$–$10^{1}$ | $10^{-2}$– $10^{1}$ | $10^{-3}$–$10^{0}$ |
| Linear SVM $\ell_2$ | C | $10^{-3}$–$10^{-1}$ | $10^{-5}$– $10^{-1}$ | $10^{-5}$–$10^{-1}$ |
| Decision Tree | $\alpha$ | $10^{-2}$–$10^{0}$ | $10^{-2}$– $10^{0}$ | $10^{-3}$–$10^{-1}$ |
| Random Forest | Feature Subsample | 0.005–0.05 | 0.005– 0.05 | 0.01–0.2 |
| GBDT | Learning Rate | 0.01–0.1 | 0.01– 0.1 | 0.01–0.2 |

Finally, the three decision-tree-based models we examined all used different strategies. For the basic decision tree we used cost-complexity pruning, where the complexity parameter $\alpha$ controls how far the tree is trimmed/simplified. In the case of the random forest model, the feature subsampling rate was optimized, which determines the fraction of all data features (i.e., wavenumbers) which is randomly selected for each individual tree of the forest. A larger fraction results in a more complex model. Lastly, the learning rate of GBDT models acts as a scaling factor for the influence of each successive tree on the ensemble, increasing complexity when the learning rate is high, and vice-versa.

Refer to table 2 for a detailed overview of the ranges of hyperparameter values tested for each classification method and dataset.

### 2.4 Model evaluation

The cross-validated classification of Raman spectra using the aforementioned methods was packaged into an open-source Python tool to simplify the collection and interpretation of performance metrics and model parameters. Classifier performance was evaluated based on generalization accuracy, p-values of McNemar tests, confusion matrices, and ROC curves, while interpretability was assessed from model coefficients or Shapley values, if applicable. The distribution of optimal (hyper)parameter values and the time required for fitting and prediction was also monitored [Zim22].

## 3 Results

### 3.1 Model Optimization

For each classification method, optimal values for hyperparameters were found as part of the inner 5-fold CV. The resulting hyperparameter values were then used to train models on the full training set and estimate performance on the independent test set.

Overall, we observed behavior that was in line with what was expected based on the relative complexity of the three datasets. Accurate classification of prokaryotic and cyanobacterial Raman spectra was generally possible with low-complexity models, or there was no significant variation in performance over the evaluated range of hyperparameters. On the other hand,

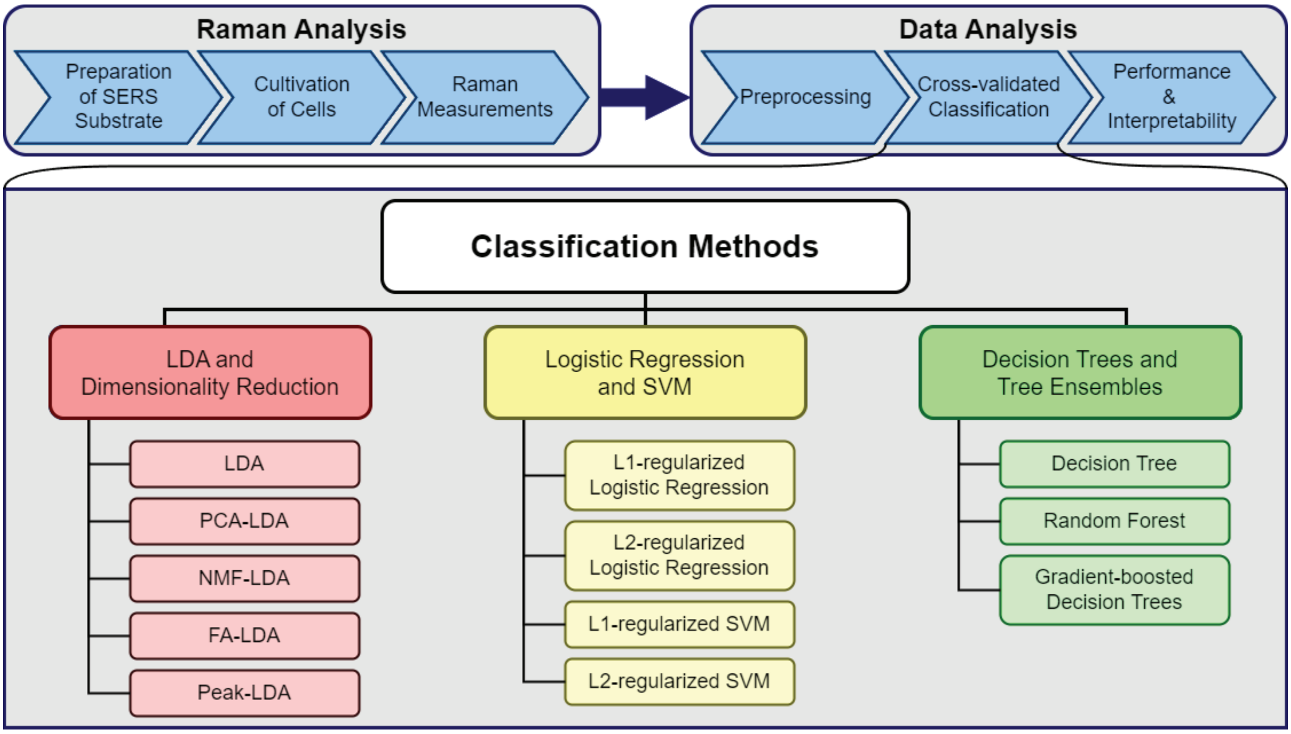

Raman Analysis
Preparation of SERS Substrate
Cultivation of Cells
Raman Measurements
Data Analysis
Preprocessing
Cross-validated Classification
Performance & Interpretability
Classification Methods
LDA and Dimensionality Reduction
LDA
PCA-LDA
NMF-LDA
FA-LDA
Peak-LDA
Logistic Regression and SVM
L1-regularized Logistic Regression
L2-regularized Logistic Regression
L1-regularized SVM
L2-regularized SVM
Decision Trees and Tree Ensembles
Decision Tree
Random Forest
Gradient-boosted Decision Trees

Figure 1: Graphical overview of the workflow ranging from data generation to analysis with a focus on the tested and compared classification methods.

the HL dataset required hyperparameter values which permitted more complex models.

### 3.2 Comparison of classification methods

Based on the generalization accuracy determined on an independent test set, we compared the performance of different classifiers both within a dataset and between datasets of increasing complexity. McNemar tests and ROC curves presented an identical picture, and are therefore omitted here.

For the **prokaryotic dataset**, a generalization accuracy close to or equal to 1 was achieved for most models, as can be seen in figure 2 (left). A notable exception with a median accuracy of 0.98 is the basic decision tree model, which is known to be prone to overfitting, especially on high-dimensional data. This behavior could be expected from the initial exploration of the data using unsupervised PCA, which already showed that the two species of bacteria could be easily differentiated. As *E. coli* is gram-negative and *P. polymyxa* is a gram-positive bacterium, their significantly different cell structures – especially the radically different cell wall compositions – likely contribute to the high classification performance.

A similar trend – although with slightly lower performance – is seen in the **cyanobacterial dataset** (Figure 2 center), which could be classified with 99 % accuracy by several methods, including NMF-LDA, $\ell_1$-regularized logistic regression and SVM, as well as gradient-boosted decision trees.

In contrast, the differently treated **HL cells** are clearly more difficult to classify, with generalization accuracies between 0.85 and 0.9 (Figure 2 right). Several possible reasons exist for this drastic change compared to the previous two datasets. For one, eukaryotic cells present a much more complex sample matrix compared to relatively simple bacteria. The larger size of the cells also creates a challenge for collect-

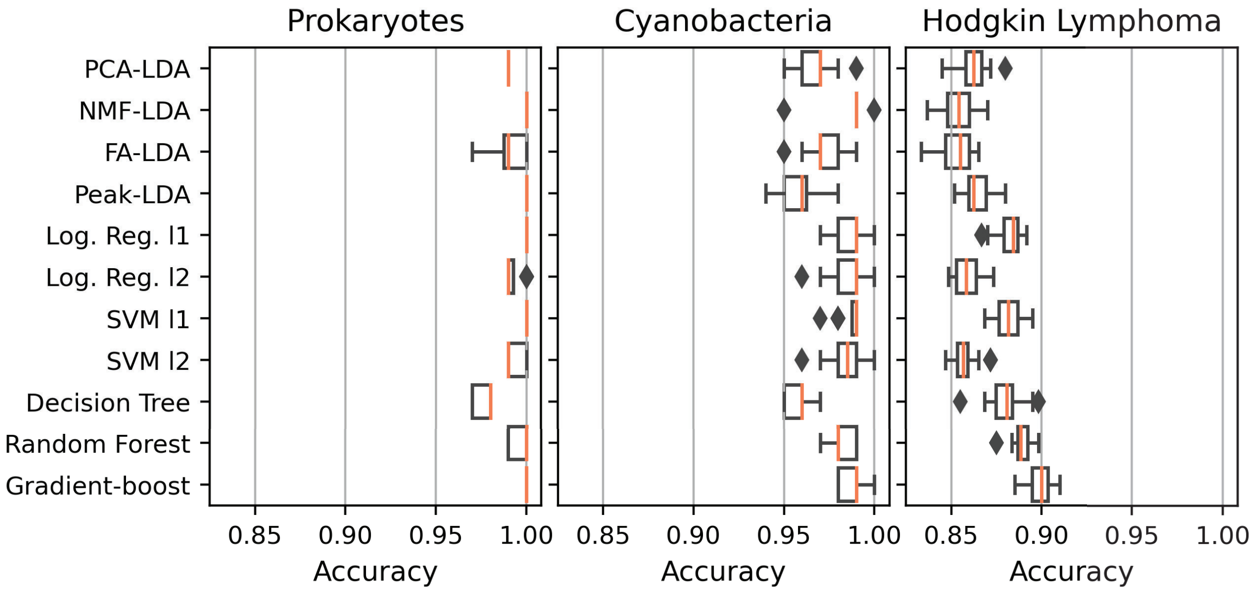


Figure 2: Comparison of the accuracy of various classification methods on three distinct datasets.

ing representative spectra of the complete cell and introduces random variations in the spectra based on the exact measurement position, hindering accurate classification further. Lastly, the changes introduced by cytostatic treatment are likely to by much smaller than the difference between completely separate species or the effects of PHB-production in a nutrient-limited environment. Gradient-boosted decision trees achieved the highest performance on this dataset, classifying 90 % of spectra correctly, followed by random forest and $\ell_1$-regularized logistic regression with accuracies of 0.888 and 0.884, respectively.

### 3.3 Impact of data complexity

Nonlinear methods showed improved performance compared to linear models for the HL dataset only, with the gradient-boosted trees model clearly standing out. On the other hand, they only reached the same accuracy as linear models in both bacterial datasets, which reflects the different sample size. The prokaryotic and cyanobacterial datasets only consist of 50 spectra per class, which appears to be insufficient for nonlinear methods to have an advantage over linear models. In contrast, 300 spectra of HL cells were collected for each treatment group, which highlights the need for a sufficiently large number of samples when analyzing high-dimensional data, especially for more advanced ML techniques.

### 3.4 Effect of regularization

For both logistic regression and SVM, regularization was achieved using the $\ell_1$ or $\ell_2$ vector norm as a penalty term. The $\ell_2$ norm penalizes the sum of squares of the model coefficients, the $\ell_1$ norm does so for the sum of absolute values, which results in a sparse model in which the coefficients for most features are zero. For both methods, the $\ell_1$ norm generally led to higher performance compared to models regularized using the $\ell_2$ norm. Due to the high dimensionality of the spectral data (≈ 2600 features), sparse coefficients in the $\ell_1$-regularized models reduce the likelihood of overfitting, as only a small number of features is actually used by the model. On the other hand, while reducing overfitting to an extent, $\ell_2$-regularized models still consider all features, making it less effective for such high-dimensional data.

# 4 Summary and Outlook

Here we successfully introduced a comprehensive Python tool, which combines quality control, preprocessing and cross-validated classification for Raman spectroscopic data using a wide range of commonly used classification methods. Applying this tool to three sets of Raman spectra from biological samples highlights the need for a careful selection process when choosing ML algorithms. Such comparisons are unfortunately lacking – or at least not explicitly mentioned – in current literature on Raman spectroscopic analysis. Instead, a small number of methods dominate current research, most commonly PCA-LDA and neural networks which represent two extremes in the scale of complexity and may not always be optimal.

Besides the raw performance that we have discussed here, a wide range of other factors influences the choice of model. To gain insight into the biochemical basis of a model, interpretability is equally important. Interpretability and explainability are the focus of the growing field of explainable artificial intelligence (XAI) [XUD+19]. Other important factors when choosing ML methods include time constraints and available computing power, which may limit the use of more complex models.

For the datasets presented in this study, investigation of the underlying biological basis for classification is currently ongoing. The next step will be to extend the binary classification to multi-class problems and test the current tool with additional datasets, primarily those which are not easily distinguished using simple linear models. The classification workflow will also be further optimized in terms of usability and automatization to further the application of ML techniques in Raman spectroscopy research.

# Data and Code Availability

The Raman spectral data analyzed during this study is available from the corresponding authors upon reasonable request.

All Python code used for the analysis of Raman spectra in this study is available freely under a *GNU General Public License* at `https://github.com/FHWNTulln/Zimmermann_RamanMachineLearning`

# References

[BHCR+16] R. Britto Hurtado, *et al.* Instant synthesis of gold nanoparticles at room temperature and SERS applications. *Physics Letters A*, 380(34):2658–2663, 2016.

[CPS+14] Dana Cialla, *et al.* SERS-based detection of biomolecules. *Nanophotonics*, 3(6):383–411, 2014.

[DKS+81] V. Diehl, *et al.* Hodgkin's disease: Establishment and characterization of four in vitro cell lines. *Journal of Cancer Research and Clinical Oncology*, 101(1):111–124, 1981.

[FGE14] Khaled Fawagreh, *et al.* Random forests: From early developments to recent advancements. *Systems Science & Control Engineering*, 2(1):602–609, 2014.

[HLD+20] Daniel-Ralph Hermann, *et al.* *In Situ* based surface-enhanced Raman spectroscopy (SERS) for the fast and reproducible identification of PHB producers in cyanobacterial cultures. *Analyst*, 145(15):5242–5251, 2020.

[LS99] Daniel D. Lee and H. Sebastian Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, 1999.

[LTC+20] Félix Lussier, *et al.* Deep learning and artificial intelligence methods for Raman and surface-enhanced Raman scattering. *TrAC*

- *Trends in Analytical Chemistry*, 124:115796, 2020.

[NK13] Alexey Natekin and Alois Knoll. Gradient boosting machines, a tutorial. *Frontiers in Neurorobotics*, 7, 2013.

[Pro16] Marek Prochazka. Basics of Raman Scattering (RS) Spectroscopy. In *Surface-Enhanced Raman Spectroscopy*, pp. 7–19. Springer, Cham, 2016.

[PZD$^+$21] Liangrui Pan, *et al.* A review of artificial intelligence methods combined with Raman spectroscopy to identify the composition of substances. *Journal of Raman Spectroscopy*, 2021.

[RGD$^+$20] Estel Rueda, *et al.* Polyhydroxybutyrate and glycogen production in photobioreactors inoculated with wastewater borne cyanobacteria monocultures. *Bioresource Technology*, 295:122233, 2020.

[RL20] Nicole M. Ralbovsky and Igor K. Lednev. Towards development of a novel universal medical diagnostic method: Raman spectroscopy and machine learning. *Chemical Society Reviews*, 49(20):7428–7453, 2020.

[Sch14] Sebastian Schlücker. Surface-enhanced raman spectroscopy: Concepts and chemical applications. *Angewandte Chemie - International Edition*, 53(19):4756–4795, 2014.

[Ste18] Theresa Stelzer. *Establisment of Raman microspectroscopy for cell-system investigations.* Master's thesis, FH Wiener Neustadt, Biotech Campus Tulln, Tulln, Austria, 2018.

[TLY$^+$21] Jia-Wei Tang, *et al.* Comparative Analysis of Machine Learning Algorithms on Surface Enhanced Raman Spectra of Clinical Staphylococcus Species. *Frontiers in Microbiology*, 12, 2021.

[Wol96] David H. Wolpert. The Lack of A Priori Distinctions Between Learning Algorithms. *Neural Computation*, 8(7):1341–1390, 1996.

[XUD$^+$19] Feiyu Xu, *et al.* Explainable AI: A Brief Survey on History, Research Areas, Approaches and Challenges. In Jie Tang, *et al.*, editors, *Natural Language Processing and Chinese Computing*, Lecture Notes in Computer Science, pp. 563–574. Springer International Publishing, Cham, 2019.

[Zim22] Daniel Zimmermann. *Classification of Cell Systems Using Raman Spectroscopy and Machine Learning.* Master's thesis, FH Wiener Neustadt, Biotech Campus Tulln, Tulln, Austria, 2022.

[ZLPH21] Daniel Zimmermann, *et al.* A tool for improving SERS reproducibility using spectral quality assessment. In *European Biosensor Symposium.* 2021.

# Scientific Computing 2023

## Track: Gaming

*Pia Somogyi, Alexander Nischelwitzer, Sandra Schadenbauer und Gerhard Zeisler:* **Development of an educational escape game for Virtual Reality, Augmented Reality and Mobile phone** ........ 24

*Carina Grillitsch:* **Performance of Generative Adversarial Imitation Learning in Various Minihack Environments** ........ 32

# Development of an educational escape game for Virtual Reality, Augmented Reality and Mobile phone

DI Pia Somogyi MA[1], Prof. DI Dr. Alexander Nischelwitzer[1], DI (FH) Sandra Schadenbauer[1], and DI Gerhard Zeisler BSc[1]

[1]FH JOANNEUM – University of Applied Sciences, Institute Business Informatics and Data Science, Eckertstraße 30i, 8020 Graz, Austria

## Abstract

Escape games provide an environment where the players must accomplish a specific goal while solving puzzles along the way within a time limit. Students from the University of Applied Sciences FH JOANNEUM developed an educational escape game, which has the goal to transfer knowledge in a playful and immersive way to test new possibilities for teaching and learning. Furthermore, technologies like Virtual and Augmented Reality were used to increase immersion and make the game more location independent.

## 1 Introduction

For developing escape games multiple factors such as atmosphere, puzzle design and interaction possibilities need to be considered. Integration of Virtual Reality (VR) and Augmented Reality (AR) into escape games bring new challenges, especially in design, concept and implementation. Regarding VR and AR it is even more important to examine different features for the collaborative work. Furthermore, the sequence of puzzles brings new difficulties and challenges in developing the application.

The aim of this research is to take a closer look at the factors that need to be considered for an cross-media escape game, which can convey content via VR, AR and mobile phone in the context of knowledge transfer and serious gaming.

The educational escape game *escape YOUR brain* was developed at FH JOANNEUM to mediate knowledge of various human senses and the human brain via VR, AR and mobile phone. The players slip into the role of a comatose patient as well as into the roles of the treating and consulting doctors. All three work together and try to wake up the patient by solving puzzles. The various parts of *escape YOUR brain* were analysed and examined more closely during usability tests with a focus on appearance, knowledge transfer, usability and application design. Furthermore, special attention was on finding the right device for the implementation of the AR application as problems occurred during the usability tests that were carried out.

## 2 Problem statement

Escape games have become increasingly popular in recent years. Many different concepts and implementation options have been developed so far, however, mainly for entertainment. Nevertheless, escape games can be used as gamification method to transfer knowledge and to make it easier for users to access new technologies. Gamification methods are inserted in developments of various applications to increase the involvement of users and to enable interaction in a playful way. That is the reason why educational escape games, which use the logic of serious gaming, are becoming more and more important in the educational field.

The basic principle of an escape game is that one or more people are in a closed room and they have to leave it within a specified time limit. This can only be done with the help of solving tasks and puzzles with clues, which can be found in this room. Through logical and creative thinking as well as cooperation, the players have to master the challenges and escape from the room [WEC15].

Gamification is the design of IT-based services for motivational support and behavioural change of addressed users. Examples of applications can be found in the corporate context, in knowledge acquisition or in influencing personal behaviour. The introduction and use of gamification elements pose challenges and require a systematic and prudent approach. Gamification can have positive effects in different areas. This includes, for example, active participation, the performance of annoying standard tasks or acquisition of new knowledge [Sti15].

One of the most formative advantages is the achievement of higher learning effects through virtual or simulated *learning by doing* with the help of the visualisation possibilities of VR and AR compared to conventional teaching methods [ZWMT18]. With the digitisation of education universities are called upon to provide new teaching and learning formats enriched with digital technologies. To date, however, digital infrastructures have only been used to a limited extent for the innovative design of teaching and learning scenarios.

Therefore, students of the Master degree program *Software and Digital Experience Engineering* at FH JOANNEUM carried out a project as well as a master thesis with the aim of creating an educational escape game for the knowledge transfer at the university.

## 3 Project *escape YOUR brain*

With the project *escape YOUR brain* the learning methods at FH JOANNEUM in the degree program Physiotherapy can be extended and the concept of gamification and its pedagogical potential for university teaching should be presented.

*Escape YOUR brain* is a cross-media escape game, in which two doctors have to help a comatose patient to regain their senses and thus their consciousness before time runs out (60min). Special focus is on the collaboration possibilities between VR, AR and mobile interfaces as well as the specific interaction possibilities of each device and the transfer of knowledge via gamification.

The escape game has to be played by three people, as all roles and devices must be occupied at all times. The VR player plays via Oculus Quest 2, the mobile player via smartphone and the AR player via smartphone (old version) or HoloLens 2 (new version). It was necessary to improve the AR mobile application and convert it into a playable version for the HoloLens 2 as well as optimising the usability and interaction possibilities in order to find out if the device was more suitable for a minimum running time of 45 minutes than a smartphone.

In addition, there is a gamemaster who supervises the game, controls progress and offers help. The escape game consists of five puzzles that must be solved in series. In VR and AR one works primarily with virtual objects or holograms. Nevertheless, physical objects can also be implemented in the puzzle design. These so-called tangibles are objects with 2D markers or physical 3D objects, which can increase immersion [PLK20]. In *escape YOUR brain* the AR player and the mobile player have a doctor's kit at their disposal, which contains physical objects and documents needed to solve the puzzles. The VR player receives one physical object - a 3D printed cube - which is placed in a bag and hung around the player's neck before the game starts. The players and the gamemaster are connected via voice chat.

Even if the puzzle design is well figured out and tested several times, players may need help to solve the puzzle. Then it is the role of the gamemaster to give the appropriate hints. It is important that direct hints are never given. Rather, they should be wrapped in riddles to give the players room to think themselves and not to take away the fun of the game[1].

Storytelling and hints are conducted through voice guidance. Intro and outro audio as well as short sequences of the two doctors' dialogues are played once a puzzle is completed. These dialogues give guidance which sense and puzzle must be completed next. Furthermore, if more hints are needed, the gamemaster can give auditive feedback.

## 4 Puzzle design

There are three different groups of puzzles that can be distinguished. On the one hand interactive or physical puzzles are solved by interacting with objects to get a certain result. On the other hand, non-interactive or mental puzzles are solved by critical thinking or combining clues. Furthermore, there are meta-puzzles which usually need to be solved at the end of the game to complete the story [RFDA$^+$21]. All three puzzle types are implemented in *escape YOUR brain*. Each of the five puzzles refer to a human sense, regarding the senses seeing, hearing, feeling, moving and smelling/tasting. It is important that the players work in a team because most of the puzzles can only be solved collectively.

### 4.1 Puzzle 1 - Hearing

The mobile player learns about human hearing by audio recordings on a dictation device. Therefore, the mobile player must find the dictation device in their doctor's kit and has to listen to the lecture, until the correct frequency occurs. The AR player sets the frequencies at a holographic medical device. Depending on these settings, the voice quality changes for the VR player. At the beginning, the VR player can hear only very indistinctly. The higher the frequency the better the VR player understands the other players. The VR player can operate a button in the virtual world should they notice a change. This button triggers a green light in the AR world, which is directly located near the frequency medical device. The task is completed when the correct frequency has been found and set at the medical device in the AR world.

### 4.2 Puzzle 2 - Motorics

The players learn about balance and postural control. The mobile player gets clues about postural control on his smartphone and through physical documents from the doctor's kit and passes these clues to the other players. The AR player has to move the device (head movement with the HoloLens 2), thereby a ball maze (platform) moves in VR. The VR player sees a brain rolling within the maze and must guide the AR player by giving information about the brain's position. Once the brain has reached its destination the puzzle is completed.

### 4.3 Puzzle 3 – Smelling/Tasting

The players learn about chemical messengers in connection with the tasting sense. The mobile player receives a physical document with transmitter name and additional information about the sense of taste and smell as well as a cloze with missing substances and has to find the missing terms. Furthermore, there are sniffing sticks with a recognisable odour. The AR player discovers the transmitters' names via QR code on different boxes from the doctor's kit. These boxes include jellybeans, which must be tasted to find the correct flavour. The VR player has tubes and fruits in the virtual world and must throw the right objects, which are revealed with the mobile player's cloze, into a cauldron. With the righ combination the task is completed.

### 4.4 Puzzle 4 - Feeling

The players learn about the two-point discrimination. The AR player sees four cubes with different symbols on each side and a numerical code on one side in the virtual room. The correct code must be passed on to the mobile player, who has

[1] 5 Merkmale, die einen großartigen Escape Room ausmachen! 2020 https://www.nowayout-escape.at/de/blog/5-merkmale-die-einen-grossartigen-escape-room-ausmachen/, accessed 2021-10-31

information about two-point discrimination and an input field for a three-digit number code. The VR player wears a bag around their neck with a 3D-printed cube. This cube has to be felt by the VR player (looking at the cube is not allowed). The cube of the VR player matches with one of the holographic cubes of the AR player. Through communication the matching cube as well as the right code has to be found.

### 4.5 Puzzle 5 - Seeing

The players learn about the location of the brain's visual centre. The AR player sees a patient lying on a bed. The part of the brain responsible for vision must be stimulated with an injection. This can be done by placing a holograph syringe over the brain areal. The mobile player assists the AR player by identifying the vision area of the brain via medical database. If the treatment is successful, VR view switches to *eagle eye*. Then the VR world is immersed in other colours and the player finds a door and buttons. The VR player has to press the correct button to open the door and to wake up.

## 5 Implementation

*Escape YOUR brain* was developed with Unity and C#. The AR Foundation framework was used for the initial implementation of the AR application [Zwi22]. The Mixed Reality Toolkit (MRTK) was used for porting this application to the HoloLens 2. The overall concept utilizes four main methods of data transmission. First the Photon Unity Network (PUN) 2 asset is used for synchronizing movement (e.g. head movement between AR and VR), game events (e.g. level completion) and voice (by using the Photon Voice extension) over a cloud based solution. Between the AR, VR and admin application (gamemaster) as well as the web dashboard (gamemaster) the WebSocket protocol is used to stream the player's view (FMETP Unity asset). Furthermore, the WebSocket protocol is also used between the dashboard and a backend server for transmitting the current time and other game information. This information is obtained through REST calls originating from the mobile application to the backend server. This backend server forwards this information via the fourth way of data transmission, MQTT, to the admin application. By leveraging this protocol, the overall architecture supports the integration of IoT devices to further enhance the players' immersion.

A diagram of the underlying communication can be seen in Figure 1.

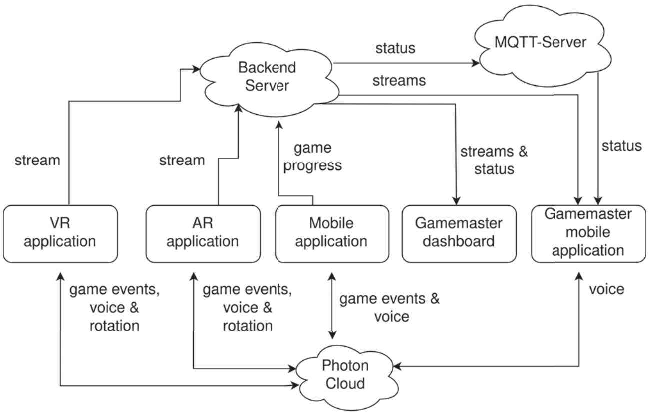


Figure 1: Communication between the system components

## 6 Design concept

The knowledge transfer via escape games can be carried out well, as each escape game is assigned to a specific topic. By dividing the content into the different puzzles, it can be structured and passed on in instalments. Escape games follow clear rules, such as the overall structure, the setup and the execution, which allows a certain security of the framework conditions. Nevertheless, there is enough freedom to design and adapt the educational escape game according to one's own needs. Any topic can be conveyed with the help of this gamification method, no matter if it is complex or shallow content, the knowledge can be prepared appropriately. The players are part of a story, are given a role and are thrown into another world. With this immersion, players remain more engaged and are motivated to follow the story, solve the puzzles and acquire knowledge along the way.

Escape games can be created in many ways and adapted to new platforms. In order to guarantee an optimal game experience the design and rules must be adapted to the specific platform. Therefore, specific interaction methods connected to the platform must be considered. The user interface as well as the user experience must be adapted to the target group since not all players might be familiar with them.

The player must not feel lost in the room and be aware of interaction possibilities. Furthermore, it is also important to show and display the boundaries of the game world so that the world cannot be misused. In particular experienced players like to test the limits of the possibilities, therefore it is important to define the boundaries beforehand. This can be done, for example, by limiting the game area or by making objects available one after the other. This can counteract the possibility of skipping puzzles.

It is important not to lose the player, especially in the puzzle transitions. Therefore, the player must be given auditory or visual feedback to symbolise when a puzzle has been successfully completed and when a new task starts. In *escape YOUR brain*, this was solved with the help of a progress bar and auditory feedback. Each time a puzzle is completed the progress bar fills up and a sensory icon appears. Furthermore, a success sound and a puzzle intro for the newly started puzzle is played. Then new areas or objects are unlocked for the player.

It is also important that each player has a specific task in a puzzle. No player should ever be idle, instead always have a task or the opportunity to participate and share important information. Although a distinction can be made between an active and a passive part in the distribution of roles, each player should also be able to play along. In addition to that it must also be ensured that the passive and active parts are distributed fairly so that each player bears an equal amount of responsibility. The active part of each player naturally promotes motivation because it conveys that they play an important role in the overall concept of the game.

Regarding the implementation of new technologies, like AR, VR and mobile, the immersion and the various interaction possibilities contribute to a good gaming experience. It is particularly exciting when the interaction possibilities of the devices are designed in a diverse way and thus the advantages and possibilities of the technologies can be presented, which is particularly appealing to the players. While VR makes it possible to create an especially high level of immersion by placing the player completely in a virtual world and thus intensifying the visual and auditory effects, AR makes it possible to combine virtual and real worlds. It is important to include objects from the real world in the game, as this combination of the two worlds is the goal of AR. Including a mobile device in an escape game makes it possible to display or supplement digital texts, which emphasises the textually bound character of this device. This role can then be used especially for players who are not yet well versed in dealing with VR and AR technologies. Thereby they can actively participate in the game, as they take on the role of the informant with an easier-to-use device.

Nevertheless, it is important not only to implement new interaction possibilities but also to test them for their usefulness. For example, during the modification process of the AR application, the interaction option via QR codes for the appearance of cubes had to be changed because they could not been rotated completely and therefore not all surfaces were visible for the player. The revision of the interaction option and the testing of the functionality is therefore essential in order to avoid interaction problems.

In general, this shows that the optimal factors for an escape game depend very much on the mediation device. Depending on the device, other factors are decisive and important. Furthermore, the puzzle design is the cornerstone of a successful escape game and the gameplay as well. Moreover, the transitions and role distribution must be planned well and precisely.

## 7 Usability testing

The test phase lasted from the beginning of October to the end of December 2022. The search for test persons and the organisation of the test dates were particularly time-consuming, which is why the active test could only start in November. A total of nine tests with each three test persons were carried out, which makes a sum of twenty-five test persons in total. In the case of the tests with only two test persons available, the mobile player was taken over by an internal colleague so that VR and AR could be fully tested. 13 males and 12 females between 18 and over 46 took part in the tests, where as the focus was on people between 18 and 35, as this also matches the target group of students. The test persons had different professional and technical backgrounds, which is why a broad spectrum of test persons could be examined in the tests. The subjects included professors and students of the FH JOANNEUM from different fields of study. The fields of study ranged from physiotherapy, radiology, social work, aircraft engineering to software development. In addition, external persons also took part, who work in the education or health sectors. As a result, the test persons showed different levels of prior knowledge in dealing with technologies like VR and AR and also had different levels of prior knowledge in the medical field. Due to the great variety of test persons, it was possible to represent a broad spectrum of possible future users.

All teams managed to complete the game, however, the time periods needed to complete the puzzles differ. The time available after completing the game varies between 39 minutes and one minute. Interestingly, the teams with more technical knowledge usually had more time left at the end of the game and were generally able to solve the puzzles faster, while the teams with less technical knowledge needed more time to complete all the puzzles. This can probably be attributed to the fact that the teams with more technical knowledge were also experienced in handling the technical devices and the general interaction possibilities, while the teams with less technical knowledge first had to get used to controlling the devices. Furthermore, the analytical way of thinking, which is common practice in the technical field, could also have been helpful in solving the puzzles more quickly. It could also be noted that the teams that already had experience with escape games were quicker in understanding the individual puzzles and tasks. Furthermore, communication also worked better in these cases, because people actively asked what the other person saw or had available. In general, it can be said that the teams with more previous experience were quicker and generally solved puzzles more easily.

A diagram of the completion of the puzzles by team and time can be seen in Figure 2.

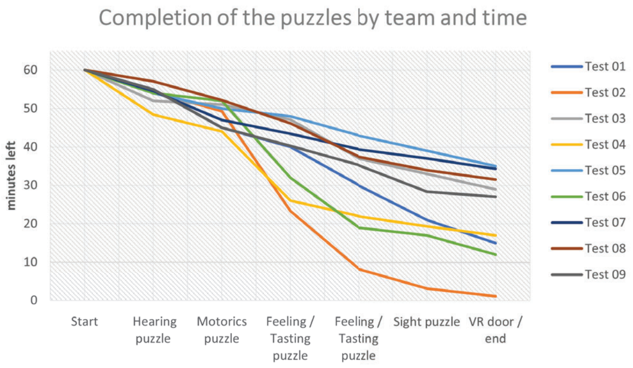


Figure 2: Completion of the puzzles by team and time

## 8 Technical results

The hypothesis "A HoloLens 2 application is more suitable for longer AR implementations with a minimum running time of 45min than a pure AR mobile application" could be confirmed and validated by conducting usability tests.

Compared to the tests carried out with the mobile AR application, the tests with the HoloLens 2 showed no problems in execution and handling. The game ran smoothly, all functionalities were executable during tests and the game could be played until the end. The usual time span for running the game was between 45 and 60 minutes, which is no problem for the HoloLens 2.

Only one game interruption and overheating of the device was recorded, although this only occurred after 90 minutes and thus exceeded the maximum time of the game anyway. However, it is possible to carry out several playthroughs in one day, whereby it should be noted that the HoloLens 2 then requires some recovery time between playthroughs. In addition to that the problems that occurred in the mobile AR application from *escape YOUR brain* could be largely solved.

The results of the multiple tests showed that major problems occurred especially with the mobile AR application, which prevented the game from running optimally. Interactable objects were not well displayed, that is why players could not distinguish between interactable objects and room decoration. Room recognition and object tracking did not run very smoothly. The position of the objects was not optimal and players could not reach the objects well for interaction. It would be advantageous if the objects could be repositioned or moved around the room to counteract space problems and limited room to maneuver. Furthermore, QR codes were not always well recognised by the device. The main problem was that the mobile device tended to overheat which led to crashes of the application. These problems have had a significant impact on the game experience so it was decided to optimise the AR application for HoloLens 2.

Thus, the HoloLens 2 is more powerful and error-free in application and use. Moreover, there are no problems regarding ground detection. The objects are placed safely in the room and anchor themselves to it after the start of the game. This is probably due to the fact that the new version of the AR application does not require the room to be scanned before objects can be instantiated, as they are projected directly into the player's view via holographic lens of the HoloLens 2. Furthermore, there is the additional option of moving the projected objects in space. This is to ensure that the objects can be reached in every game situation. The QR code recognition also works more smoothly using the supported QR code recognition (using Microsoft.MixedReality.QR asset) from HoloLens 2. A crash of the game or similar problems and error messages were not experienced during development and testing with a running time of 60 minutes. In general, it can be said that there were hardly any or no major problems in development during the optimisation process and execution of the new AR application. Minor technical problems during testing were quickly solved with a restart and only occurred with an unstable network. The only major technical problem did not occur with the HoloLens 2 but with the Oculus Quest 2, which no longer displayed the game correctly. However, this error was triggered by the device itself, as no changes were made to the software. The problem was solved by using a new device.

However, it should be noted that the controls of the HoloLens 2 were not always easy to apply for everyone, especially when users had no experience with the device. Then it was more difficult for the player to execute the hand gestures precisely and thus enable error-free interaction with the objects. The interaction was sometimes executed unclearly, which is why it took longer to pick up certain objects or interact with them. Furthermore, the home menu of the HoloLens 2 was often called up unintentionally. This was less common with more experienced users. Therefore, an introduction at the beginning of the game is essential, in which the interaction possibilities are explained well in order to keep the frustration in the game low. One possibility is a specially developed tutorial level, which can be played before the start of the game to learn the general controls.

## 9 Conclusion

In general positive feedback was received from players of all the game devices, both verbally and in written form. The players had fun completing the puzzles and were enthusiastic about the various interaction options and puzzle tasks.

The test persons gave feedback that all in all the game was fun to play and that they were excited about progress in the game. Furthermore, they found the game to be innovatively designed and can imagine using it for knowledge transfer. The players felt comfortable in the game world and the interaction possibilities were not too complex for them. The puzzles were not too easy but challenging, so each team managed to complete all the puzzles in the given time of 60 minutes. Especially cooperation with the other team members and the use of new technologies motivated the players. The test persons valued the usage of new technologies in the form of educational escape games rather positively and can imagine that knowledge can be conveyed through this form of gamification.

Overall it can be noted that educational escape games can be a new possibility to transfer knowledge in a playful and immersive way. The use of educational escape games at universities could improve the involvement of students and can provide the opportunity to create a simulation for specific scenarios where students can test and apply their knowledge or gain new information through a playful approach.

## Acknowledgments

This work would not have been possible without Fuchs Thomas BSc, Weiland Sebastian BSc, DI Zeisler Gerhard BSc and Zwitkovits Gerhard BSc, who have contributed to the *escape YOUR brain* project and thus made the basis for the scientific research possible.

I would also like to thank FH-Prof. Dipl.-Ing. Dr. Alexander Nischelwitzer for his advice and assistance during this research. Moreover, I extend my thanks to DI (FH) Sandra Schadenbauer for attendance during testing.

I would also want to thank the employees and professors of the physiotherapy and radiology programs for their shared knowledge and assistance in providing the team with the resources needed.

## References

[PLK20] David Plecher, Maximilian Ludl, and Gudrun Klinker. Designing an ar-escape-room with competitive and cooperative mode. In Benjamin Weyers, Christoph Lürig, and Daniel Zielasko, editors, *GI VR / AR Workshop*. Gesellschaft für Informatik e.V., 2020.

[RFDA$^+$21] Jessica Reuter, Marta Ferreira Dias, Marlene Amorim, Claudia Figueiredo, Claudia Veloso, and Claudia Figueiredo. How to create educational escape rooms? strategies for creation and design. In *Eighth International Conference on Technological Ecosystems for Enhancing Multiculturality*, TEEM'20. Association for Computing Machinery, 2021.

[Sti15] Stefan Stieglitz. Gamification – vorgehen und anwendung. *HMD Praxis der Wirtschaftsinformatik*, 52, 12 2015.

[WEC15] Markus Wiemker, Errol Elumir, and Adam Clare. Escape room games: „can you transfer an unpleasant situation into a pleasant one?". *Game based learning*, 55:55–75, 2015.

[Zwi22] Gerhard Franz Zwitkovits. Ar collaboration & interaction: Augmented reality interaction development and cross-platform multiplayer. Bachelor's thesis, FH JOANNEUM, 2022.

[ZWMT18] Benedikt Zobel, Sebastian Werning, Dirk Metzger, and Oliver Thomas. *Augmented und Virtual Reality: Stand der Technik, Nutzenpotenziale und Einsatzgebiete*, pages 123–140. Springer Fachmedien Wiesbaden, Wiesbaden, 2018.

# Performance of Generative Adversarial Imitation Learning in Various Minihack Environments

Carina Grillitsch[1]

[1]FH JOANNEUM – University of Applied Sciences, Master's Degree Programme "Data and Information Science", Eckertstraße 30i, 8020 Graz, Austria

## Abstract

Deep Reinforcement Learning (DRL) methods require to periodically reward or punish the agent during its interaction with the environment. However, in some applications it may be easier to demonstrate the desired behaviour rather than specifying a reward function. Learning from demonstrations of an expert is called Imitation Learning (IL). Applying the concept of Generative Adversarial Networks (GANs) to imitation learning in such a way, that a discriminator learns to distinguish between expert and agent data and provides this feedback as a reward to the agent, is called Generative Adversarial Imitation Learning (GAIL).

The aim of this thesis is to provide an understanding of how deep reinforcement learning and imitation learning work, to implement methods from both areas and to evaluate the performance of the DRL method Proximal Policy Optimization (PPO) and GAIL in different MiniHack environments. For this purpose, the theoretical basics of reinforcement learning in general and actor-critic methods in particular, and the basics of imitation learning, GANs, and finally GAIL were researched. This was followed by an implementation tailored to the MiniHack framework, which is a derivative of the popular game NetHack and was developed to allow designing environments for RL at different levels of difficulty. One of the first results of this work was a program that extracts the expert data from the recorded games in a useful format. After the selection of hyperparameter values, the performance of the PPO implementation was finally compared with that of GAIL and some surprising results were achieved. At first it seemed that in the environments used, PPO was superior, but as the complexity of the tasks increased, the potential of GAIL emerged.

Further research may apply these results to tasks of greater complexity and proof, whether the agents can apply what they have learned in a generalized way. The highest goal would be an agent who manages to play through NetHack.

## 1 Introduction

NetHack, a turn-based dungeon-crawler game, first published in the 1980s, has been a playground for developing Reinforcement Learning (RL) strategies in the last years, and it is also a suitable challenge for Imitation Learning algorithms, as there are millions of recorded games of the still active community available, for example on the NetHack alt.org servers. This master's thesis applies the GAIL algorithm, proposed by [HE16], on various environments of MiniHack, a RL framework derivated from the NetHack Learning Environment [KNM+20] to answer the following research question.

### 1.1 Research Question

How does an agent trained by GAIL perform in various MiniHack environments compared to an agent trained by PPO?

This question leads to the following objectives:

- Research of theoretical foundations of PPO and GAIL.
- Implementation of PPO and GAIL in Python.
- Extract expert data out of recorded games.
- Train agents with both methods in various MiniHack environments.
- Evaluate the results.

### 1.2 Code Repository

The data and Python code of this thesis are available on `https://gitlab.com/data-and-information-science-fh-joanneum/mth-dat20-generative-adversarial-imitation-learning-in-minihack` under the MIT license.

## 2 Generative Adversarial Imitation Learning

To put it briefly, GAIL combines the principles of Generative Adversarial Networks with Reinforcement Learning in a way, that the generator of a GAN [GPAM$^+$14] is replaced by a PPO [SWD$^+$17] agent (or TRPO [SLA$^+$15] in the original paper) and the discriminator has to distinguish between state-action pairs of an expert and of the agent. The feedback of the discriminator is the reward for the agent, so there is no reward needed from the environment, which is one of the main characteristics of imitation learning algorithms. The algorithm is shown in Alg. 1.

**Algorithm 1:** Generative adversarial imitation learning [HE16]

```
Input: Expert trajectories $\tau_E \sim \pi_E$, initial policy
       and discriminator parameters $\theta_0$, $\omega_0$
for $i$ = 0,1,2, ... do
    Sample trajectories $\tau_i \sim \pi_{\theta_i}$
    Update the discriminator parameters from $\omega_i$ to
     $\omega_{i+1}$ with the gradient

     $\hat{\mathbb{E}}_{\tau_i}\left[\nabla_\omega \log\left(D_\omega\left(s,a\right)\right)\right]+\hat{\mathbb{E}}_{\tau_E}\left[\nabla_\omega \log\left(1-D_\omega\left(s,a\right)\right)\right]$

    Take a policy step from $\theta_i$ to $\theta_{i+1}$, using the
     TRPO rule with cost function $\log\left(D_{\omega_{i+1}}\left(s,a\right)\right)$.
     Specifically, take a KL-constrained natural
     gradient step with

       $\hat{\mathbb{E}}_{\tau_i}\left[\nabla_\theta \log \pi_\theta\left(a|s\right) Q\left(s,a\right)\right]-\lambda\nabla_\theta H\left(\pi_\theta\right),$

     where $Q\left(\bar{s},\bar{a}\right)=\hat{\mathbb{E}}_{\tau_i}\left[\log\left(D_{\omega_{i+1}}\left(s,a\right)\right)|s_0=\bar{s},a_0=\bar{a}\right]$

end for
```

## 3 The MiniHack Environment

MiniHack [SKK$^+$21] is a derivation of NetHack, a still popular game of the 1980s. NetHack is hard to solve even for human players because it contains hundreds of enemy and object types and every game is unique because of the procedurally generated levels. Nevertheless, basically each level has the same objective: the hero starts at the entry point and has to find the exit with increasing difficulty level. What makes NetHack suitable also for Imitation Learning is the availability of over six million played games on the public NetHack server at alt.org [NAO].

MiniHack is a sandbox framework for designing environments for RL and is built around the NLE, so it also uses the Gym interface for interactions between the agent and the environment. It comes with many predesigned tasks and environments, from which the following five were used for the evaluation of the performance of GAIL in comparison to an equivalent PPO agent in the scope of this thesis.

- MiniHack-Room-Random-5x5-v0
- MiniHack-Room-Random-15x15-v0
- MiniHack-MazeWalk-Mapped-15x15-v0
- MiniHack-River-Narrow-v0
- MiniHack-Corridor-R2-v0

The Room environments are simple squared rooms with either a side length of 5 or 15, where the agent has to find the exit (as in all MiniHack games). MazeWalk is also a squared room with the side length 15, but with walls in between. In the River environment the agent has to cross a river by first pushing boulders into it to make a passable area in the river. In the Corridor environment are two rooms connected with a corridor, but entrance and exit are not necessarily in different rooms. The action space in these environments is reduced to navigational actions.

## 4 Training Data

NetHack games on the alt.org server are stored in ttyrec format, which is commonly used for terminal records. Manually played games in NLE or MiniHack are saved in almost the same file format with the difference, that also the input actions in form of the used keys are stored. Unfortunately the terminal records cannot be used directly, so the first outcome of this thesis was a Python script, which translates the ttyrec files into arrays corresponding the "tty_chars" and "tty_colors" observations of the MiniHack environment. This script is available under the link in Section 1.2.

## 5 Implementation Details

The three neural networks for actor, critic and discriminator have basically the same architecture and differ only in the output layer and the adding of the actions in the case of the discriminator. Nevertheless they do not share their parameters. The input consists on the one hand of the "tty_chars" and "tty_colors" (optional) observations and on the other hand of their cropped versions "tty_chars_crop" and "tty_colors_crop", which are hero-centred arrays of size 9x9 and focus on the agent's immediate surroundings.

Vectorized environments were used for a better distribution of the samples and to reduce the number of forward passes, as the observations of all environments per step are forwarded as one batch. The usage of vectorized environments induces fixed-length trajectory segments, which allows to train the agent even for long-horizon games in a single episode.

Table 1: Hyperparameters for PPO and GAIL

| **Hyperparameter** | **PPO** | **GAIL** |
|---|---|---|
| Reward discount factor | 0.99 | 0.99 |
| Exponential weight discount | 0.94 | 1.0 |
| Clipping parameter | 0.2 | 0.2 |
| Actor learning rate | 0.0003 | 0.0001 |
| Critic learning rate | 0.001 | 0.001 |
| Discriminator learning rate | n/a | 0.00065 |
| Learning rate decay of actor & critic | 0.9995 | 0.9999 |
| Learning rate decay of discriminator | n/a | 0.9999 |
| Batch size | 2048 | 2048 |
| Mini-batch size agent | 128 | 256 |
| Mini-batch size discriminator | n/a | 256 |
| Number of sub-environments | 4 | 4 |
| Number of convolutional layers | 2 | 2 |
| Discriminator loss threshold | n/a | 0.0 |
| Update discriminator every $x$ epochs | n/a | 1 |

Table 2: Rewards used in the MiniHack environments

| **Parameter** | **Default** | **Room** | **Corridor**<br>**River**<br>**MazeWalk** |
|---|---|---|---|
| reward_win | 1 | 1 | 40 |
| reward_lose | 0 | 0 | -10 |
| penalty_step | -0.01 | -0.01 | -0.01 |
| penalty_time | 0 | -0.001 | -0.001 |

## 6 Hyperparameters

The search for the optimal hyperparameter configuration took a large part of the time, as the parameters of three neural nets (actor, critic and discriminator) had to be coordinated with one another. After the configuration (see Tab. 1) was chosen, only minor changes in single parameters were made during the evaluation runs, where necessary.

Besides the model hyperparameters also the parameters for each environment (mainly consisting of the rewards for the PPO agent and the horizon) had to be configured. The rewards are shown in Tab. 2. The horizons were changed frequently throughout the training, as they turned out to be a key factor for a successful GAIL training.

# 7 Evaluation

## 7.1 Simulation Setup

The evaluation of the agents started with each environment run only with one seed. When the training was successful, a new seed was added. It is to say, that with increasing complexity, more of the trainings failed, so sometimes only one run out of ten (or more) was successful at all. The initial idea was to train the agents on a larger set of variants on the environment, so they may be able to generalize the concept and apply it on unseen variants. This objective was abandoned, as the GAIL trainings were not as successful as hoped.

As measure of the performance served on the one hand the number of iterations needed until the training was finished, and on the other hand the time elapsed. If only the number of iterations were considered, it would have skewed the results, as one PPO iteration needs less time than a GAIL iteration, and, if the expert data exceeded the batch size, a GAIL run would have more than one iteration per epoch (which was not the case in this work). A training was considered as finished, when the mean environmental reward (which was not the reward the GAIL agent received, but was used for reasons of comparability) gained over the last 100 episodes reached a reward threshold depending on the reward for finding the exit an some tolerance for the time-progressing actions to get there.

## 7.2 Performance of GAIL in Comparison to PPO

In the relatively simple environment of the 5x5 room (trained up to 20 seeds), the GAIL agent is clearly outperformed by the PPO agent, both in steps and in time (see Fig. 1). The GAIL agent needs the first 500 training steps to train the discriminator up to a certain level, so the generator gets a feedback it can learn of. During this "burn-in" phase the PPO agent has already finished its learning. When the generator has reached a certain level of knowledge, the discriminator is regularly fooled, which results in the high loss, and that again leads to a worse training of the generator.

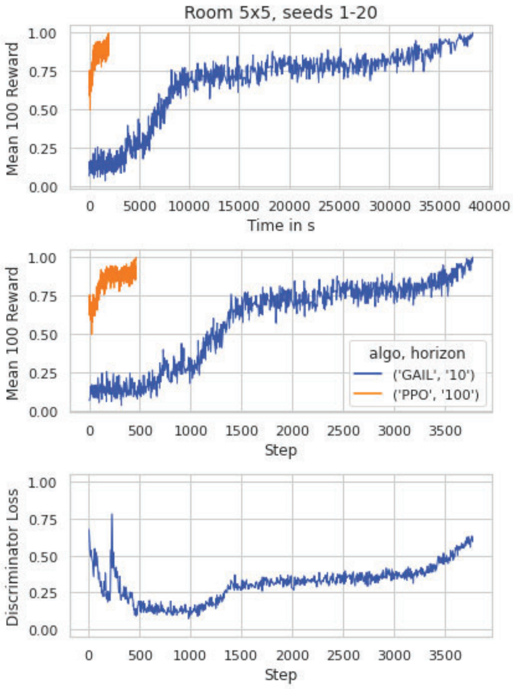


Figure 1: Room 5x5, seeds 1-20, performance per step and per time

In the larger version of the Room environment, the training was successful up to 10 seeds. Fig. 2 shows that the advantage of the PPO agent decreases. The discriminator loss shows the GAN-typical peaks and valleys when the generator outperforms the discriminator temporarily and vice versa.

The first setup of the MazeWalk environment with seed 1 was easy to solve for both the PPO and the GAIL agents, as only one action, namely moving north, was necessary. With adding a second variant of the environment, Fig. 3 shows, how different GAIL runs can be. The big difference between both the GAIL runs is probably coincidence and not necessarily caused by the small difference in the horizon. The pattern in the GAIL curves, where firstly they seem to be stuck, is explained by the short horizon, where the agents never have chance to reach the exit, an then quite suddenly get a clue about the task. A PPO agent would have problems to learn any-

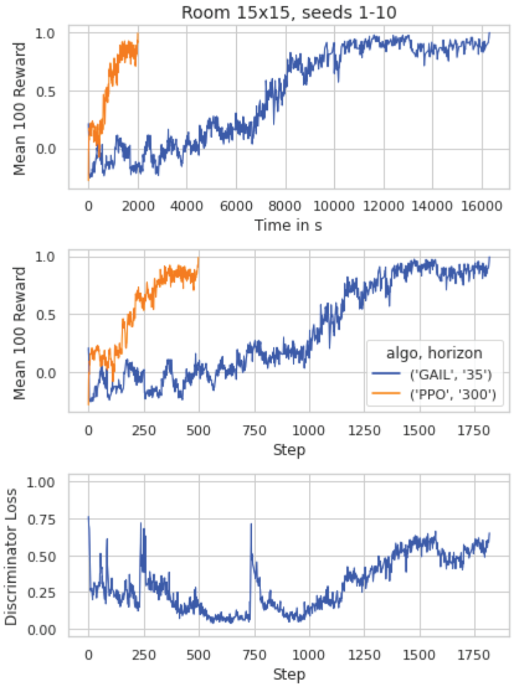


Figure 2: Room 15x15, seeds 1-10, performance per step and per time

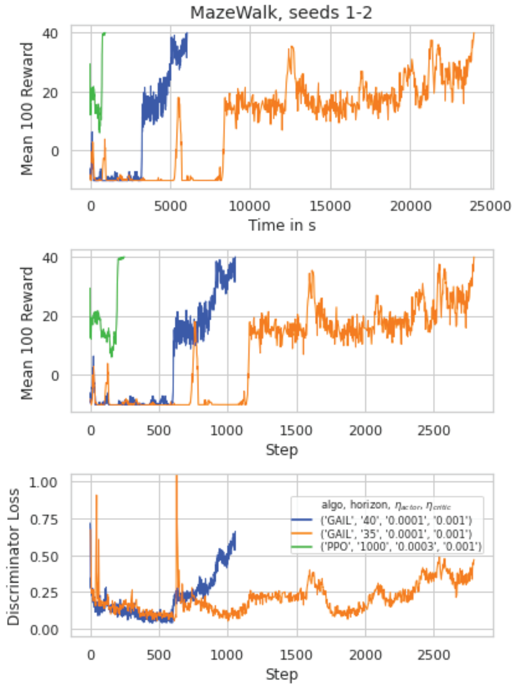


Figure 3: Mazewalk, seeds 1-2, performance per step and per time

thing within this narrow horizon, but the GAIL agent is rewarded for the similarity of its state-action pairs to the ones of the expert, not for reaching some point in the environment.

The River environment revealed the weaknesses of the simulation setup. Whereas the training with seed 1 was finished quickly because both agent variants managed to find a trivial solution by only moving always eastwards, the next step with adding another seed showed, why a regular test run with deterministic action selection was important. The PPO agent seemed to have solved the task, but when testing in deterministic mode, it became clear, that it had only reached the exit by chance frequently. The reward threshold had been set too low.

Conveniently both seeds 1 and 2 can be solved by simply moving eastwards, even when the exit lies further south than the entrance, because crossing a river field may move the hero to any of the neighbouring fields randomly. The PPO agent finds this solution, whereas the GAIL agent follows the expert's actions. With three variants, the GAIL trainings are not successful anymore. However, because of the constantly failing trainings, an attempt was made, if different horizons would lead to a better result. As Fig. 4 shows, they didn't. But despite the very similar reward curves, the discriminator loss curves diverge. With a very large horizon, as can be seen in the blue curve, the discriminator seems to be almost perfect as the loss drops to almost zero. But in this case, the discriminator only learns to distinguish between the states and not the state-action pairs, as the agent will stray around somewhere. This is not necessarily bad, as the reward curves show a similar training progress, but needs to be considered when comparing discriminator loss curves. On the other hand, the green curve shows an extremely short horizon of the absolute minimum, namely the exact length of the longest expert trajectory. The discriminator cannot distinguish between expert and agent data, and therefore the agent does not get enough feedback. It is the typical trade-off between exploration and exploitation.

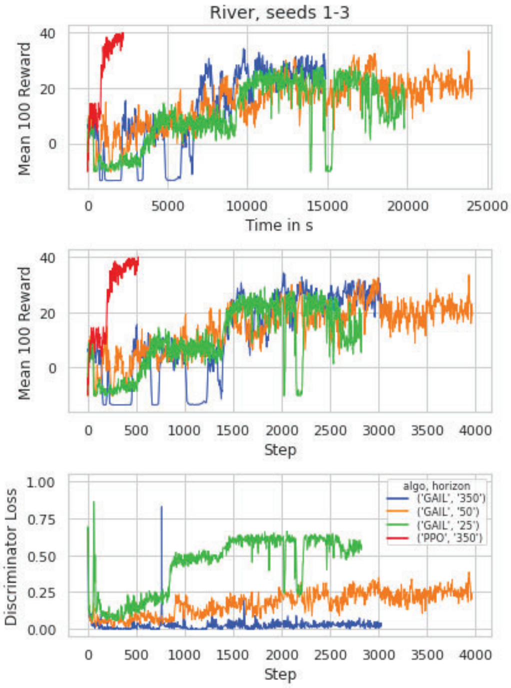


Figure 4: River, seeds 1-3, performance per step and per time

The training in the Corridor environment shows the performance of GAIL increasing with growing complexity of the environment, compared to PPO. The Corridor-R2 environment consists of two rooms connected with a corridor, where additional difficulties occur, like locked doors, which must be kicked multiple times to open, or hidden passages, which can only be found by multiple search actions. The first three seeds are relatively easy to solve, as entrance and exit are in the same room. Fig. 5 shows, that PPO solves this tasks quickly, whereas the GAIL agent needs almost ten times more iteration steps. The next three seeds of this environment contain more difficult variants with a locked door and a hidden passage, as mentioned before. Fig. 6 shows, that, although the GAIL training did not succeed within the given time frame, it performs far better in comparison to the PPO agent as with only the simpler variants of the environment.

The horizon, which is the maximum amount of steps before an environment is aborted, has a big

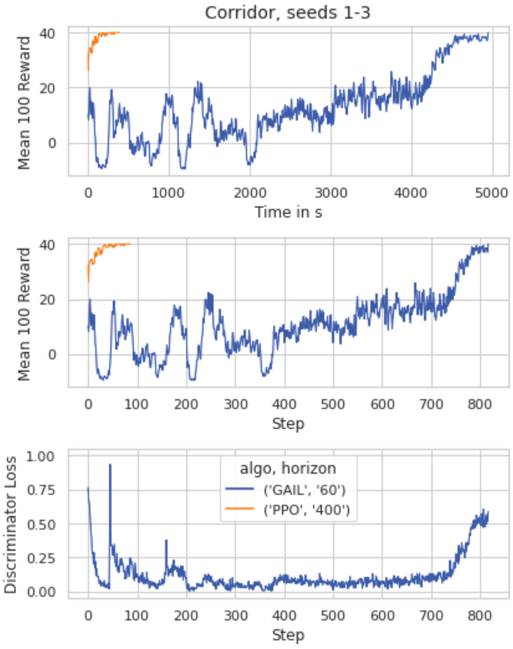


Figure 5: Corridor, seeds 1-3, performance per step and per time

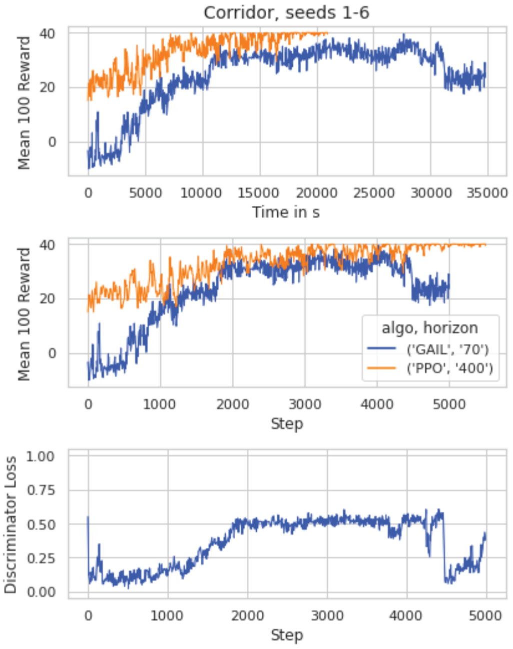


Figure 6: Corridor, seeds 1-6, performance per step and per time

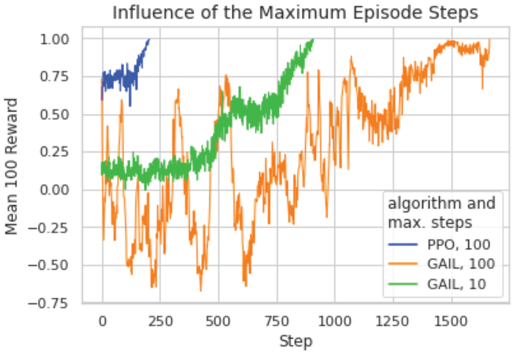


Figure 7: Influence of the horizon on the performance of the GAIL training in the `MiniHack-Room-Random-5x5-v0` environment, restricted to seven different seeds.

influence on the performance of the GAIL training (see Fig. 7). If the horizon is too large, the agent may get lost in regions, where no expert data is available and may not recover. If that happens, the discriminator distinguishes rather the states than the state-action pairs, thus becoming very strong, while the agent cannot learn, because it does hardly matter which action it takes. Suitable values for the maximum number of steps allowed are about two or three times the maximum length of the expert trajectories.

# 8 Summary and Outlook

Reinforcement and Imitation Learning methods are both very challenging. The balancing of two neural networks, be it actor and critic in RL, or generator and discriminator in GANs, is already hard, but adding another net as in GAIL makes it most demanding. Nevertheless, the training of both PPO and GAIL was frequently successful, and where it was not, it led to learnings for future work.

In easy environments PPO performed far better than GAIL, but the more complex the tasks became, the better GAIL worked out, in particular, where actions did not have an immediate effect, for example, when searching for a hidden passage needed longer in the Corridor task. Although the PPO agents often learned relatively quickly to find the exit, they often did not use the optimal path, or needed a very long time to optimize it, if the reward threshold was very strict. When the GAIL training worked, the agent almost always followed the expert's path.

For future work it will be useful to have a look at established methods for improving GAN performance, for example, try another cost function for the discriminator or strengthen the discriminator by making it deeper or giving it more training loops, either by a fixed step or depending on the current discriminator loss. The agent seems to learn best, if the loss of the discriminator is somewhere between 0.1 and 0.2. Future measures should keep a focus on how to maintain this value throughout a training run.

Another possibility to improve the performance may be the variation of the horizon during a training run. One may also try if more expert data per seed, either with slight variants in the solution or with added noise, may help.

Furthermore, the neural nets of actor and critic may be adapted, for example with more layers or more features.

Finally, there might be a way to include sparse environmental rewards in addition to the feedback of the discriminator, for example, for finding the exit, or related to the in-game score.

# References

[GPAM$^+$14] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C Courville, and Yoshua Bengio. Generative Adversarial Nets. In *NIPS*, 2014.

[HE16] Jonathan Ho and Stefano Ermon. Generative Adversarial Imitation Learning. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016.

[KNM+20] Heinrich Küttler, Nantas Nardelli, Alexander H. Miller, Roberta Raileanu, Marco Selvatici, Edward Grefenstette, and Tim Rocktäschel. The NetHack Learning Environment. In *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, 2020.

[NAO] Public NetHack server at alt.org (NAO). `https://alt.org/nethack/`, Accessed: 2022-01-08.

[SKK+21] Mikayel Samvelyan, Robert Kirk, Vitaly Kurin, Jack Parker-Holder, Minqi Jiang, Eric Hambro, Fabio Petroni, Heinrich Kuttler, Edward Grefenstette, and Tim Rocktäschel. MiniHack the Planet: A Sandbox for Open-Ended Reinforcement Learning Research. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*, 2021.

[SLA+15] John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust Region Policy Optimization. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 1889–1897, Lille, France, 07–09 Jul 2015. PMLR.

[SWD+17] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal Policy Optimization Algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

# Scientific Computing 2023

## Track: Gesundheit

*B. Pfeifer, S. Neururer, W. Hackl, M. Schweitzer, B. Tilg und P. Raudaschl:* **An Agent-based Framework for COVID-19 Simulation to Predict Outcome for the Federal State of Tyrol and Province South Tyrol** ........ 44

*Samereh Goodarzi:* **Brain Tumor Detection in MRI Images with Transfer Learning from Pre-Trained CNN Models** ........ 50

*Tristan Till, Sebastian Tschauner, Georg Singer, Holger Till und Klaus Lichtenegger:* **Machine learning in human healthcare: Optimizing computer-aided detection and diagnosis systems using an x-ray data set of pediatric wrist fractures** .. 60

# An Agent-based Framework for COVID-19 Simulation to Predict Outcome for the Federal State of Tyrol and Province South Tyrol

S Neururer[1,2], W Hackl[2], M Schweitzer[2], B Tilg[2], P Raudaschl[2], und B Pfeifer[1,2]

[1]Division for Digital Medicine and Telehealth, UMIT TIROL - Private University for Health Sciences and Health Technology, Hall (Tyrol), Austria
[2]Tyrolean Federal Institute for Integrated Care, Tirol Kliniken GmbH, Innsbruck, Austria

## Abstract

The current outbreak of COVID-19 has shown the deadly potential and the impact of infectious diseases on areas other than healthcare (e.g. economy or social life). Disease control measures, including social distancing, quarantines, curfews, or lockdowns, have to be adopted in a considerate manner. Infectious disease modelling supports decision makers with adequate information regarding the dynamics of the pandemic and assists in planning measures to prevent the healthcare system from collapsing. An agent-based simulation package named "surviral" for simulating infectious diseases is presented. This framework was used by the Tyrolean crisis management team, and provided decision-makers with very accurate models on federal state levels as well as on district and municipal level. The framework can be used for various infectious diseases and thus builds a basis for future monitoring.

## 1 Introduction

On 31 December 2019, the World Health Organization (WHO) was notified of cases of pneumonia with unknown causes in the Chinese city of Wuhan. Subsequently, on 7 January 2020, Chinese authorities identified the cause as a novel coronavirus, tentatively named 2019-nCoV [SB20, HS22]. On 11 March 2020, the WHO officially declared the outbreak a pandemic due to the rapid increase of cases outside China. By that time, more than 118,000 cases had been reported by WHO from 114 countries including 4,291 deaths. By mid-March 2020, the European region had become the epicenter of the pandemic, reporting more than 40 % of all confirmed cases worldwide. As of 28 April 2020, the European region accounted for 63 % of global mortality caused by the virus. In Austria, the first viral infections were registered on 25 February 2020. On 16 March 2020, a nationwide lockdown was ordered. Additionally, disease control measures, including social distancing, quarantines, curfews, or lockdowns were adopted in order to contain the spread of the disease. This was referred to as "flattening the curve", as there was a fear that the healthcare system would collapse due to the exponential increase in number of infections and in hospitalizations as well as intensive care needs. The Corona pandemic made us aware of the impact, a disease can have on our everyday lives. Millions of people became infected, and over 5 million people died from or with 2019-nCoV. Public life came to a halt. It has shown the deadly potential as well as the impact of infectious diseases on areas other than healthcare (e.g. economy or social life). Therefore, control and containment measures have to be adopted in a very considerate manner. To be able to contain the spread and to protect the health care system from collapsing, when no medicine and/or vaccination is available, and to plan and assess the effectiveness of (the above mentioned) measures, simulations and epidemiological modeling has become a major factor. The aim of this paper is to present the Surviral platform, an agent-based modeling and simulation framework, which enables the simulation of different communicable diseases by parameterizing the general model with individual disease parameters, geographical structures, and population densities. This framework provided the Tyrolean crisis management team with accurate models and forecasts of the disease dynamics on federal state, district and municipal levels in order to plan and adopt specific measures where needed.

## 2 Methods

Most contagious diseases can be modeled using mathematical approaches to analyze and understand the epidemiological behaviour or to predict the disease dynamics. Therefore, different approaches have been developed in the past. The classic SIR epidemic model, where class S denotes the number of susceptibles, class I denotes the number of infectives and class R denotes the number of recovered individuals. The sum of the given initial value problem is $S(t) + I(t) + R(t) = N$, with $N$ being the size of the population. However, the SIR model is not adequate to model birth and natural death, immigration and emigration, passive immunity and spatial arrangement adequately. To model infection diffusion through space, partial differential equations (PDE) are needed. With PDE models it is possible to simulate the spread of a disease over a population in space and time. However, the integration of geographical conditions, demographic realities, and keeping track of each individual remains impossible. For this purpose, agent-based models can be used. An agent-based model is a dynamical system, in which time and space is discrete. Each agent can be modeled to have a specific behaviour which enables accurate models [AAW21, SFI+21, LKKW21].

Cellular automata as well as agent-based tools for simulating infectious diseases can be used to discover the behavior of the disease or to work out contingency plans. Specialized models for different diseases have been presented in the past [KSM+21, MHR20, AAE+20, MMB20]. Our proposed framework, however, can simulate different infectious disease types and allows the use of different geographical maps with marked population densities and enables the modeler to change the behavior of the individuals during the spread. Furthermore, it allows changing parameters (medication, quarantine, birth rate, death rate, virus morbidity, etc.) during the simulation process to obtain realistic results. The framework was adapted to the COVID-19 related parameters and is used on a daily basis since the outbreak of the pandemic in March 2020.

The parameters of each individual are accessible at every time step of the simulation, which facilitates the exploration of data patterns using statistical approaches [PWH+10]. Figure 1 depicts the overall struture of the framwork. The framework itself is implementred in Java connected to a PostgreSQL (version 15) database system where all relevant source data as well as the simulation results are stored. Further, for visualization of results R (version 4.2.2) is used. The utilized packages are RPostgres, ggplot2, DBI, flextable, xlsx and lubridate.

Figure 2 depicts the main simulation routine where all state transitions are computed and stored.

## 3 Results

Using data sources provided by the federal state government, publicly available data sources, and information retrieved from hospitals, 10-days forecasts of the disease dynamics as well as occupancy of usual care and intensive care beds were calculated. They were placed at the disposal of decision makers. In fall 2021, a sharp increase in infections was predicted, both by models and virologists. As a result, the federal state government was facing the decision to impose a lockdown light or an entire lockdown. However, the models showed that a short-term lockdown was able to alleviate the situation for the hospitals, while a partial lockdown would have caused problems for the healthcare system. Results of the simulation or prediction can be seen in figure 3. Based on these results, it was concluded in advance that a short lockdown prior to Christmas would lead to decreasing infection numbers over the Christmas holidays.

In addition to the 10-days forecasts, long-term models and forecasts were calculated to predict the development of the infection waves, especially during the colder season. Also the impact of the waves for hospitals in terms of usual care and intensive care beds occupancy was predicted. Figure 4 shows that the incidence of infections had picked up over the summer months due to vacation returners as well as less attention to measures, leading to increased hospital occupancies with a time lag.

However, the models and forecasts were calculated not only for the federal state level, but also for the district and municipal levels. This allowed health policy makers to plan containment measures not only for the entire federal state, but to take them exactly where they were needed. This minimized the burden on the rest of the state's population.

The models were subsequently compared with the actual values. This comparison resulted a mean deviation for normal care beds of 1.5% and 1.2% for intensive care beds. This high accuray of the forecasts led to a high level of confidence in the system by health policy and decision makers.

## 4 Outlook & Discussion

The current outbreak of COVID-19 has shown the deadly potential and the impact of infectious diseases on areas other than healthcare (e.g. economy or social life). In order to better plan containment measures during the COVID-19 pandemic as well as estimate bed occupancy for normal and intensive care beds, the presented agent-based simulation framework "surviral" was developed and used. Surviral is a powerful tool for making accurate 10-day forecasts of pandemic dynamics. The forecasts, which were calculated not only at the state level but also at the district and municipal levels, were used by policy makers as a basis for decision making regarding containment measures such as lockdowns, curfews, masking requirements, etc. Simualtion at the district and municipal level also allowed measures to be more targeted and applied only to affected regions. Agent-based modeling is a relatively new modeling approach to meet the requirements resulting from the increasing complexity of our world. For parametrizing the presented agent-based model, different data sources were integrated in a database system and used. The "surviral" package was implemented in the programming language Java. R Studio was used to perform the analytics. The model forecasts the hospitalization rates (standard and intensive care) for the Federal State of Tyrol (Austria) and the Province South Tyrol (Italy) with an accuracy of about 1.5 percent average error. This ensured that standard care was maintained for as long as possible without restrictions. Furthermore, various measures were estimated and thereafter enforced. Among other things, communities were quarantined based on the calculations while, in accordance with the calculations, the curfews for the entire population were reduced. While the framwork showed a very high accuracy for the prediction of the COVID-19 pandemic in Tyrol (and South Tyrol), it is not limited to COVID-19 and can be used for various infectious diseases. Thus, it can be used as a basis for future monitoring and early warning systems (e.g. for the influenza or influenza-like diseases).

## References

[AAE+20] Hafeez Aderinsayo Adekola, Ibrahim Ayoade Adekunle, Haneefat Olabimpe Egberongbe, Sefiu Adekunle Onitilo, and Idris Nasir Abdullahi. Mathematical modeling for infectious viral disease: The covid-19 perspective. *Journal of public affairs*, 20:e2306, November 2020.

[AAW21] Amal Adel Alzu'bi, Sanaa Ibrahim Abu Alasal, and Valerie J. M. Watzlaf. A simulation study of coronavirus as an epidemic disease using agent-based modeling. *Perspectives in health information management*, 18:1g, 2021.

[HS22] Jingyuan Wang Jiawei Cheng Honghao Shi, Qijian Tian. Libepidemic: An open-source framework for modeling infectious disease with bigdata. In *Proceedings of the 31st ACM International Conference on Information and Knowledge Management*, CIKM '22, page 4980–4984, New York, NY, USA, 10 2022. Association for Computing Machinery.

[Ins23] Robert Koch Institut. Covid-19 (coronavirus sarscov-2), . last access: 06.02.2023.

[KSM$^+$21] Subhas Khajanchi, Kankan Sarkar, Jayanta Mondal, Kottakkaran Sooppy Nisar, and Sayed F. Abdelwahab. Mathematical modeling of the covid-19 pandemic with intervention strategies. *Results in Physics*, 25:104285–, 06 2021.

[LKKW21] Paul Layie, Vivient Corneille Kamla, Jean Claude Kamgang, and Yves Emvudu Wono. Agent-based modeling of malaria control through mosquito aquatic habitats management in a traditional subsahara grouping. *BMC Public Health*, 21(1), mar 2021.

[MHR20] Sarah B. Minucci, Rebecca L. Heise, and Angela M. Reynolds. Review of mathematical modeling of the inflammatory response in lung infections and injuries. *Frontiers in Applied Mathematics and Statistics*, 6, 8 2020.

[MMB20] Sayantan Mondal, Saumyak Mukherjee, and Biman Bagchi. Mathematical modeling and cellular automata simulation of infectious disease dynamics: Applications to the understanding of herd immunity. *The Journal of chemical physics*, 153:114119, September 2020.

[PWH$^+$10] B. Pfeifer, M. Wurz, F. Hanser, M. Seger, M. Netzer, M. Osl, R. Modre-Osprian, G. Schreier, and C. Baumgartner. An epidemiological modeling and data integration framework. *Methods of information in medicine*, 49:290–296, 2010.

[SB20] T. Zheng X. Pei S. Baloch, M.A. Baloch. The coronavirus disease 2019 (covid-19) pandemic. *J Exp Med. 250(4), 271-278.*, 2020.

[SFI$^+$21] Md Salman Shamil, Farhanaz Farheen, Nabil Ibtehaz, Irtesam Mahmud Khan, and M. Sohel Rahman. An agent-based modeling of covid-19: Validation, analysis, and recommendations. *Cognitive computation*, pages 1–12, February 2021.

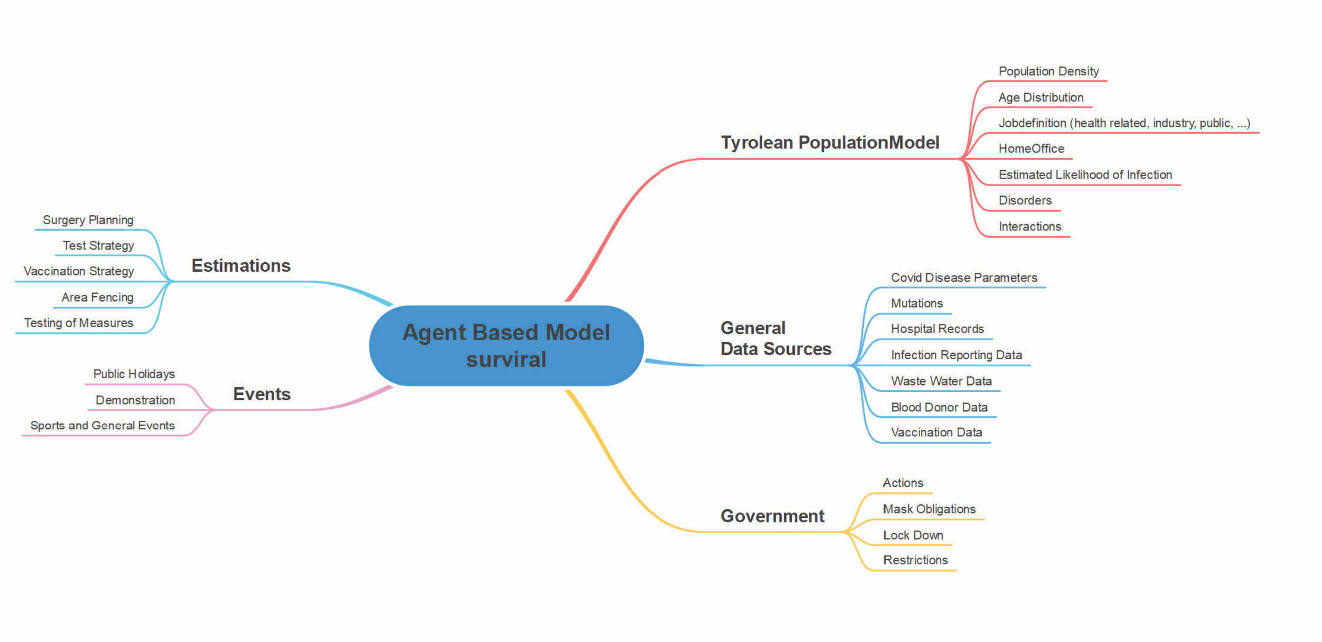
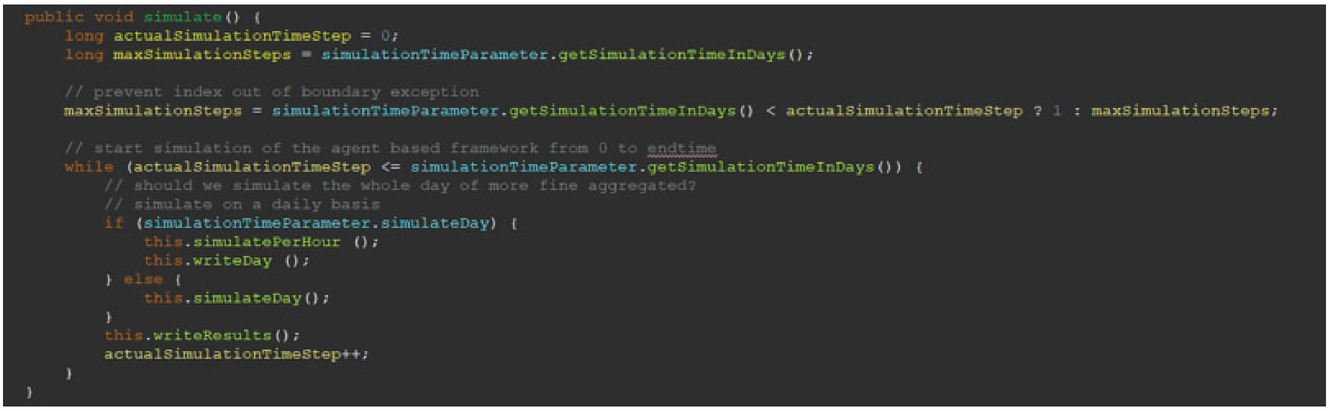


Figure 1: Overall structure of the framework including the data sources, the synthetic population model and the surviral simulation environment



```
public void simulate() {
    long actualSimulationTimeStep = 0;
    long maxSimulationSteps = simulationTimeParameter.getSimulationTimeInDays();

    // prevent index out of boundary exception
    maxSimulationSteps = simulationTimeParameter.getSimulationTimeInDays() < actualSimulationTimeStep ? 1 : maxSimulationSteps;

    // start simulation of the agent based framework from 0 to endtime
    while (actualSimulationTimeStep <= simulationTimeParameter.getSimulationTimeInDays()) {
        // should we simulate the whole day of more fine aggregated?
        // simulate on a daily basis
        if (simulationTimeParameter.simulateDay) {
            this.simulatePerHour ();
            this.writeDay ();
        } else {
            this.simulateDay();
        }
        this.writeResults();
        actualSimulationTimeStep++;
    }
}
```

Figure 2: Main simulation routine of the agent-based framework is presented; for each agent the simulation and state transaction function is computed and the results are written to the database. The simulation can be performed hourly or on a daily prediction basis to get a more aggregated view.

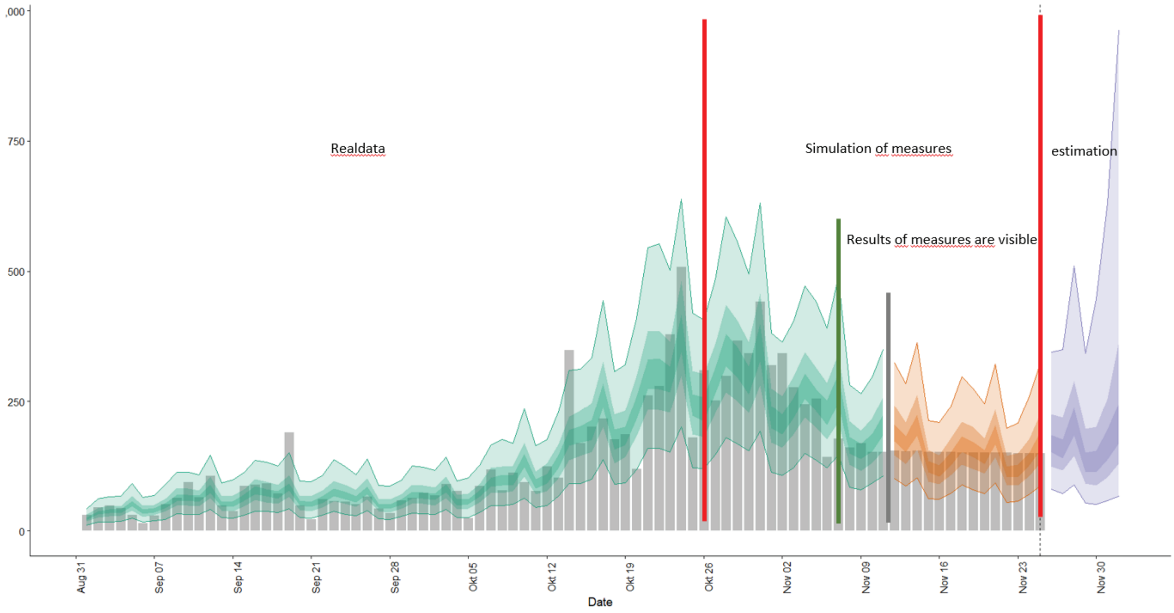


Figure 3: Simulation of measures (lockdown vs. lockdown light) for decision makers.

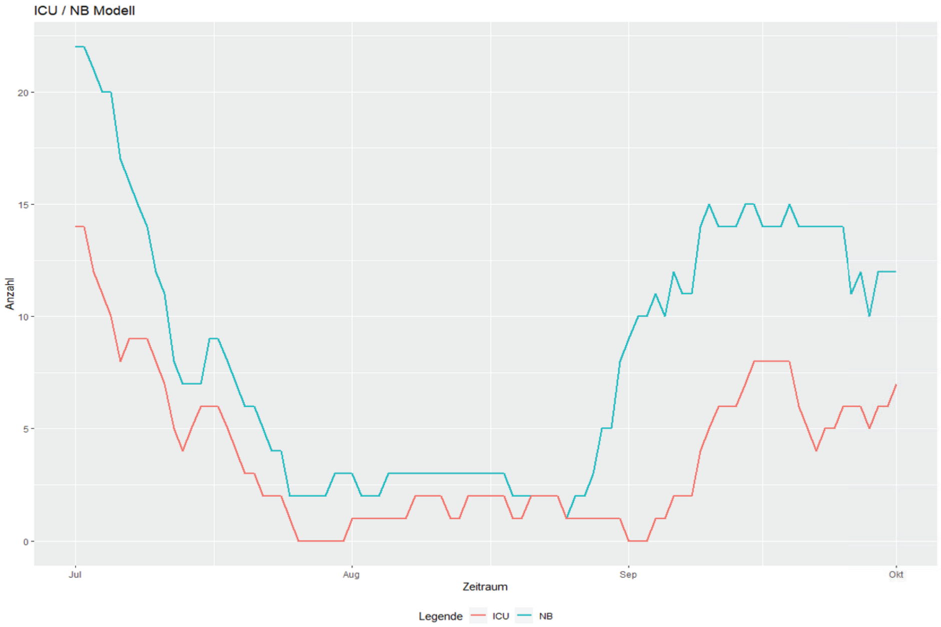


Figure 4: Long-term prediction to get a bigger picture of the infection dynamics and to be able to initiate actions as quickly and efficiently as possible

# Brain Tumor Detection in MRI Images with Transfer Learning from Pre-trained CNN Models

Samereh Goodarzi[1]

[1]FH CAMPUS WIEN - University of Applied Sciences, Favoritenstraße 226, 1100 Vienna, Austria

## Abstract

The aim of this work is to show how transfer learning can be applied in a convolutional neural network for tumor detection in brain MRI images with a very small image dataset. In the first level, a convolutional neural network is built from the scratch for a binary classification problem of brain tumor detection. Then, the model is enhanced using data augmentation methods, regularization, and hidden layers reduction. In the next step, three architectures of convolutional neural networks (VGG16, VGG19, and ResNet50) are implemented as based layers for rebuilding the network with a transfer learning approach. In the end, the models are trained on a small dataset from Kaggle containing 253 brain MRI images with and without tumors for 50 epochs with a batch size of 8, and the control points are stored for further analysis. The results demonstrate that even if methods for data augmentation are used to generate larger amounts of data, establishing a CNN model from the scratch, in the case of a small dataset available, is not advisable at all and will not produce acceptable or trustworthy results. Furthermore, it is shown that the VGG16-based model outperforms the VGG19 and ResNet50-based models for 50 epochs, according to a comprehensive examination of accuracy and confusion matrix findings for three transfer learning-based deep convolutional neural network models.

## 1 Introduction

A brain tumor is an abnormal cell growth inside the skull or brain. Brain tumors can develop as primary cancers when they originate there, or they can metastasize from other parts of the body and progress to the brain. Magnetic resonance imaging has proven to be a useful tool for clinical analysis and the diagnosis of brain tumors among all imaging techniques. The last few years have shown the importance of machine learning for using computer-aided diagnosis in medical applications. Convolutional neural networks (CNNs), which have recently experienced a boom, have started to perform better than other well-established models in a range of fields, including data processing, healthcare, and computer vision [JM21].

### 1.1 Research objectives

The main objective of this work is to build and train a convolutional neural network model from the scratch which can classify MRI scans of brain into two categories of with or without tumor. Implementing a transfer learning approach, different pretrained model architectures have been used to train our binary classification model and evaluate the changes in the model performance. Comparing various metrics, the performance of the model has been evaluated with or without transfer learning implementation for various pretrained architectures and the results have been analyzed.

# 2 Study Background: Automatic MRI image classification of brain tumors

Varied MRI picture categories have different contrast levels for various tissue types. Due to the intricacy of these MRI pictures, analysis is a challenging and time-consuming task. Manual analysis is prone to mistakes and can have serious consequences if one occurs because these analyses involve human life. Therefore, an automated method to accurately diagnose tumors is required. In recent years, a number of automatic MRI image classification methods have been presented. The most prevalent deep learning-based image analysis method is a convolutional neural network [WKW16]. Due to its architecture, which automatically learns picture information in a hierarchical fashion from low level to high level, a convolutional neural network dominates image processing tasks like classification and segmentation [RSK19]. Numerous automated approaches have recently been proposed and created by researchers in order to categorize brain cancers using MRI data [AS21]. A deep learning model is an ensemble of a feature extractor and a classifier put together. A deep learning model automatically takes care of extracting features, whereas in machine learning, we have to determine the features for the model, and then the model will give us the best classifier. If we are unsure of what kind of features to use, deep learning models are a good way to go but we need a large enough set of data to get started. A convolutional neural network classifier is made of several layers of neurons. For image classification, these can be dense or more frequently convolutional layers. For an image classification problem, dense layers will probably not be enough. We must try applying convolutional layers and take advantage of the many ways we can arrange them. But we can also use a “shortcut”. There are fully trained convolutional neural networks available for use. We only retrain the last layer we add. This technique is called Transfer Learning [RSK19]. Transfer learning appears to be highly helpful in the context of cancer diagnosis, where obtaining large-scale datasets has proven to be challenging or nearly impossible due to patient privacy concerns.

## 2.1 MRI image classification of brain tumors using CNN

The most widely used deep learning-based image analysis method is CNN. Multiple layers of convolution are applied by a convolutional neural network, improving the ability to extract picture features. The expected output is first tested to see if the model can predict it using the training data that is currently available. CNN is capable of using input photos to automatically identify features. Image features like pixel intensity, color, shape, and so forth are retrieved during training [PK21].

## 2.2 Image classification with transfer learning

Transfer learning is the step in deep learning where, rather than beginning from the scratch, the learning process uses patterns that have already been discovered while resolving a previous problem. Using pre-trained models that have already been trained on a sizable benchmark dataset allows for the implementation of transfer learning [RSK19]. When the amount of target training data is restricted, transfer learning becomes necessary. This can be the result of the data being uncommon, expensive to gather and identify, or inaccessible. The use of existing datasets that are linked to, but not precisely the same as, a target domain of interest makes transfer learning solutions an alluring strategy as big data repositories become more widespread. Transfer learning has been effectively used in a wide range of machine learning applications, including text sentiment classification, image classification, classification of human activity, classification of software defects, and classification of multilingual text [WKW16]. It can be employed for successful performance, particularly for medical pictures where substantial training data are not readily available. [RSK19].

# 3 Implementation structure

The convolutional neural network implemented in this research is structured based on multiple layers. The First layer is the convolutional layer which is used to detect features. The next layer will be the non-linearity layer to introduce non-linearity to the system. As the third part of the layers a pooling layer is applied which reduces the number of weights and prevents overfitting. Flattening layer as the next one prepares data for classical neural network. And finally, a fully-connected layer is used as a standard neural network for classification. Implementation outlines of this research are as follows:

- First we build a convolutional neutral network from the scratch, we choose the number of blocks and layers of each block and decide which hyperparameters to use and save the checkpoints of the results.

- In the next step we change the model and apply data augmentation and other methods to overcome the over-fitting problem and save the checkpoints again for the result analysis.

- The implementation is then followed by applying 3 different CNN pre-trained models as the base models for transfer learning in order to increase the prediction ability of the model.

- We gather all the results and analyze them to come to a conclusion in the end.

## 3.1 Environment set up and dataset

This project is built using the Google Collab platform, which is a great deep learning tool for building machine learning models. We first turn on the GPU for the runtime type of executing the code on Colab. Then we continue by importing the necessary libraries and dataset. The data set is Brain MRI Images for Brain Tumor Detection provided on Kaggle [dat] which is a very small dataset contaning 253 brain MRI images categorized into 2 classes of 125 brain images with tumor and 98 images of healthy brain. All the images are of 240 * 240 pixels. The dataset is directly imported from Kaggle through the python codes.

## 3.2 Train and test data preparation

To give the network clear and precise images, first the training data is prepared. The images are transformed into categorical data using sklearn LabelBinarizer [bia] and then into NumPy arrays. The one hot encoding generates binary labels for our two classes. After one-hot encoding, the final shape of the dataset is verified and the data is split into 70 percent training and 30 percent testing sets which then has been used for evaluation of the model performance.

## 3.3 Hyperparameters

Hyperparameters naming the loss function, optimizer and metrics should be defined for compiling the model. In order to train the model, the loss function is utilized in conjunction with the optimizer. The loss function BinaryCrossentropy() [cro], which matches our binary classification problem, the metrics Accuracy() [acc], and the Ädamöptimizer [ada] from Keras library have been used for compiling the model regarding related works in the literature. The weights were updated by taking into consideration the learning rate of 0.0001 as well. If the loss rate does not vary considerably over the course of the four epochs, this is reduced using the function ReduceLROnPlateau [red] from the Keras package, which is set to 0.0001. We used a minimal learning rate of 0.000001, which means that the function must not lower the learning rate during the training phase by more than 0.000001.

## 3.4 Model training parameters

The model has been trained for 50 epochs with the batch size of 8. The batch size can have a significant impact on the model's performance and the training time. In general the optimal batch size will be lower than 32. A small batch size ensures that each training iteration is very fast, and although a large batch size will give a more precise estimate of the gradients, in practice this does not matter much since the optimization landscape is quite complex and the direction of the true gradients do not point precisely in the direction of the optimum [GS19].

# 4 CNN model implementation from the scratch

For the implementation of the CNN model, the open-source Python library Keras was used. We had 3 blocks of convolutional layers with a nonlinearity layer following every convolutional layer using ReLU nonlinear function as it produces the best results in terms of the speed at which the neural network can be trained. The tensorflow MaxPool2D layer is applied as the pooling layer for each convolutional block. In the end of every block we used the keras Dropout layer which randomly sets input units to 0 with a frequency of rate at each step during training time, which helps to prevent overfitting. Inputs not set to 0 are scaled up by 1/(1 - rate) such that the sum over all inputs is unchanged. Finaly, we add the classifer block which consists of a flattening layer and 4 dense layers with units from 128 to 2(number of classes) in the end. The Softmax activation function is impelemented in the last dense layer which converts the scores to a normalized probability distribution to be displayed in results. According the the model summary, the total number of trainable parameters is 11, 373, 122 where the non-trainable parameters is zero as we are not using any pre-trained model in this step.

## 4.1 First model training outcomes

The first model training results is shown in a table in Tab. 1.

As it can be seen the model accuracy is very low (22%), and the loss is unacceptable. Also taking Fig. 1, into consideration which shows the behavior of the model accuracy and model loss during the epochs, it can be easily inferred that an over-fitting has occurred with the training process which leads to an increase in the validation loss while the accuracy loss decreases.A machine learning model's objective is to generalize patterns seen in training data such that it can accurately predict new data that has never been presented to the model before. When a model makes too many adjustments to the training data, it over-fits identifying patterns that don't exist and then performs poorly when predicting fresh data.

Table 1: First model training outcomes

| Parameters after 50 Epochs | Values |
|---|---|
| Total Training Time (GPU) | 82.95s |
| Time per Steps | 210ms |
| Training Loss | 0.0068 |
| Validation Loss | 1.0668 |
| Training Accuracy | 0.3609 |
| Validation Accuracy | 0.2292 |
| Total Model Loss | 1.0185 |
| Total Model Accuracy | 0.2237 |

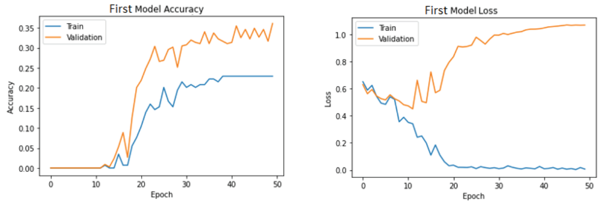


Figure 1: First model loss and accuracy behavior through 50 epochs

# 5 Data augmentation implementation to counteract overfitting

Convolutional neural networks have many learnable parameters, whereas the most advanced neural networks contain millions of trainable parameters and require a lot of training photos. Our CNN model was obviously at a high risk of

Table 2: Second model training results with augmented dataset

| Parameters after 50 Epochs | Values |
|---|---|
| Total Training Time (GPU) | 115.58s |
| Time per Steps | 95ms |
| Training Loss | 0.0311 |
| Validation Loss | 0.4153 |
| Training Accuracy | 0.1213 |
| Validation Accuracy | 0.1842 |
| Total Model Loss | 0.4153 |
| Total Model Accuracy | 0.1842 |

over-fitting given a very small dataset from the brain tumor photos. The fewer samples for training, the more models can fit our data. In order to avoid over-fitting, we used a number of techniques, including data augmentation, regularization using dropout and batch normalization, reducing the number of hidden layers in order to decrease the parameters and avoid complexity in our model. In our new model with data augmentation implementation, the total and trainable parameter counts have been drastically decreased to 2,952,686 and 2,952,680, respectively (almost 75% reduction).

### 5.1 Second model training outcomes

The data augmentation has been performed using ImageDataGenerator from Keras [aug] which generates batches of tensor image data with real-time data augmentation. The images have been rotated with an angle of 15 degree and a shift range of 0.01 has been applied to images width and height. The new model has been compiled and trained with the same parameters as the previous one taking also data augmentation into consideration. The results can be seen in a table in Tab. 2.

The behavior of the model accuracy and model loss during the epochs is shown in Fig. 2. As it can be seen,the over-fitting problem has been resolved using new model in combination with the data augmentation as the validation loss is decreasing according to a reduction in training loss values through epochs. However, the accuracy is still very low (less than 20%) and the total loss is yet considerable which means that the implemented optimizer and loss function are still not presenting well for finding the optimized value of weights in the new model even with less parameters and augmented dataset. This is the problem that we are going to solve it with transfer learning approach which is explained in the next sections.

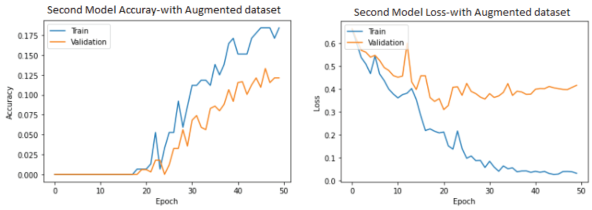


Figure 2: Second model loss and accuracy behavior through 50 epochs using data augmentation

## 6 Transfer learning implementation

In previous sections we have mentioned that the convolution layers, nonlinearity layers, pooling layers, flattening layer, and fully connected layers are the typical five key layers of a traditional CNN design. In the case of having a very small dataset and a very large amount of weights to be optimized, a practical approach is using a pre-trained model as the base layer of the model and then just adding the last block which contains the flattening and fully connected layers. In this approach we freeze (not train) the first convolutional blocks which have already been trained on a huge dataset(e.g. ImageNet, which contains 1.2 million images with 1000 categories) and then we use it as the feature extractor to initialize our model. In the so-called transfer learning method, with the help of pre-trained model weights, the number of trainable parameters will reduce considerably which leads to faster training process (in comparison with the same trained model from the scratch), higher performance, reliable results and simple operation due to reusable weights of the pre-trained model. Transfer learning is frequently used to solve challenges in predictive modeling that take input from images. Various transfer learning models have been mentioned in the literature such as: GoogleNet, Inception, DenseNet, ResNet, VGG , etc [GS19]. We have implemented 2 different models from The visual geometry group (VGG) network architecture namely VGG16 and VGG19 as well as one model from Residual network (ResNet) source which is ResNet50. These are three frequently used pre-trained models from related works which we are going through in the next sections and illustrate our implementation and results on each of the models.

### 6.1 VGG16 model implementation and results

VGG network's outstanding generalization capability and relatively simple construction make it popular in a wide range of fields. The top-5 test accuracy for the VGG16 model in ImageNet, a dataset of more than 14 million images divided into 1000 classes, is 92.7%. The model has 16 layers, which is why it is known as VGG16. These layers consist of thirteen convolutional layers and three fully connected layers. RGB images measuring 224 * 224 can be entered into the VGG16 model. Convolution layers with stride 1 employ 3 * 3 filters. Five maximum pooling operations are completed, applying a window size of 2 * 2 and a stride of 2. Three fully connected layers with 4096, 4096, and 1000 neurons each are present in the final section. The Softmax layer, the final layer, generates a probability value between 0 and 1 to identify the class to which the network outputs belong. All hidden layers employ the ReLU as their activation function. Local response normalization (LRN), which increases memory usage and computation time but does not enhance performance in the ImageNet dataset, is not a feature of the VGG16 network. In the VGG16 network, there are around 138 million trainable parameters [PG21]. In the first try of implementing transfer learning we built our model based on VGG16 and a classifier is just added to the based model as the last block of flattening and dense layers. All layers from the based model are frozen and they will not be trained anymore. The model has been compiled with the same hyperparameters as the previous models. The number of total, trainable and non-trainable parameters of 14747650, 32962 and 14714688 respectively are calculated in the model summary, which means that the number of trainable parameters has been decreased to a very great extend. In the end the model has been trained for 50 epochs with the batch size of 8 as before and taking data augmentation into consideration. The results of the training is gathered in Tab. 3. As expected, the accuracy has been increased remarkably from less than 20% for the model from the scratch to around 80% for the model based on VGG16.

Table 3: Model training results based on VGG16 with augmented dataset

| Parameters after 50 Epochs | Values |
|---|---|
| Total Training Time (GPU) | 199.022s |
| Time per Steps | 777ms |
| Training Loss | 0.5707 |
| Validation Loss | 0.5626 |
| Training Accuracy | 0.7456 |
| Validation Accuracy | 0.7763 |
| Total Model Loss | 0.5626 |
| Total Model Accuracy | 0.7763 |

### 6.2 VGG19 model implementation and results

Architecturally speaking, VGG19 and VGG16 are comparable. In VGG19, there are three extra convolutional layers. Three fully connected layers and sixteen convolutional layers total. Similar to VGG16, 3 * 3 filters are employed in stride 1 convolution layers here. Max pooling operations are performed using stride 2 and a window size of 2 * 2. Each of the three completely connected layers has 4096, 4096, and 1000 neurons. Softmax layer is the bottom layer. All hidden layers employ ReLU as their activation function. LRN is not a part of the VGG19 architecture, which includes around 143 million train-

Table 4: Model training results based on VGG19 with augmented dataset

| Parameters after 50 Epochs | Values |
|---|---|
| Total Training Time (GPU) | 221.46s |
| Time per Steps | 131ms |
| Training Loss | 0.5610 |
| Validation Loss | 0.5657 |
| Training Accuracy | 0.7574 |
| Validation Accuracy | 0.7632 |
| Total Model Loss | 0.5657 |
| Total Model Accuracy | 0.7632 |

Table 5: Model training results based on ResNet50 with augmented dataset

| Parameters after 50 Epochs | Values |
|---|---|
| Total Training Time (GPU) | 137.58s |
| Time per Steps | 81ms |
| Training Loss | 0.5799 |
| Validation Loss | 0.5574 |
| Training Accuracy | 0.7456 |
| Validation Accuracy | 0.6842 |
| Total Model Loss | 0.5574 |
| Total Model Accuracy | 0.6842 |

able parameters [PG21]. The second implementation of transfer learning has been done using VGG19 as the base model and the the classifier block is the same as for VGG16-based model. All layers from the based model are frozen to keep them non-trainable. Applying same hyperparameters and same learning rate for compiling process, the model summary gives the total, trainable and non-trainable numbers of parameters as 20057346, 32962 and 20024384 respectively. The non-trainable parameters are more than that for VGG16 due to greater number of connected layers and the trainable number of parameters are the same as VGG16 regarding the same filtering block in both models. In the last step the model has been trained again for 50 epochs with the batch size of 8 as previous training models with augmented dataset. The results of the training is shown in Tab. 4.

### 6.3 ResNet50 model implementation and results

ResNet has a different structure than the conventional sequential CNN design. With its various connections, ResNet tries to address CNN networks' degradation issue. The degradation issue arises as deep networks start to converge. The efficiency of the network hits saturation as the depth rises (as is to be expected), but then rapidly declines. To address the degradation issue, ResNet adds shortcut connections (directly connecting input of some (n + x)-th layer to some n-th layer) between layers.

Significant information from the previous layer can be conveyed to the subsequent layers using shortcut connections. Convolution layers are sequentially connected, but they also increase the output of the convolution block by adding the original input [HZRS16]. Five convolution steps make up ResNet50, a 50-layer network built using the ImageNet dataset as training data. Conv (1 x 1) is the sole convolution block of Conv1, which has only one convolution layer. Three, four, six, and three convolution blocks, respectively, are present in the remaining convolution layers (Conv2, Conv3, Conv4, and Conv5). There are three following layers in each convolution block: Conv (1 x 1), Conv (3 x 3), and Conv (1 x 1). By using the average pooling layer during downsampling, the feature map's size is altered. In addition to this, the network's last fully linked convolution layer is included for categorization purposes. The same approach as the previous implementation procedures has been used in this step for building the model based on ResNet50 and all layers from the based model are frozen to keep them non-trainable [PG21]. The number of total, trainable and non-trainable parameters are calculated as 23718978, 131266 and 23578712 respectively derived from the model summary. Finally the model has been trained with the same parameters as before for 50 epochs and the batch size of 8 with augmented dataset. The results of the training can be seen in Tab. 5.

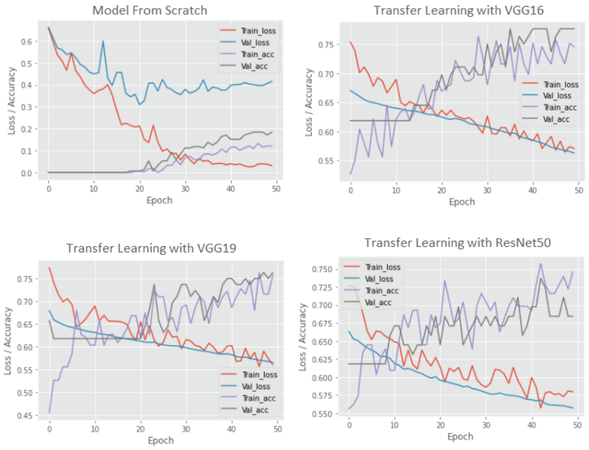


Figure 3: Comparing behavior of the models by observing accuracy and loss for 50 epochs

# 7 Summary and Outlook

In this study we tried to explain an application of transfer learning in the medical filed for finding abnormality in brain MRI images. We showed that in the case of small dataset provided for training the model, the accuracy on model prediction is too low for the results to be reliable even if data augmentation methods, regularization, and hidden layers reduction are applied. Implementing transfer learning approach gives remarkable better results on model prediction accuracy in this case. However, the performance of the model can improve using techniques for initiating much more reduction in validation and training loss along with the increase in accuracy value.

The outcomes of the codes has been presented in previous sections. In this section we go through an analysis over the findings and give a conclusion in the end. First we examined a model from the scratch which led to an over-fitting and a very low accuracy. In order to overcome the over-fitting issue, a new model with less complexity has been implemented considering data augmentation as well as using batch normalization and drop out techniques. In following steps, three CNN models with different architectures have been used as the base model. All three models were pre-triained models from the Keras library. The maximum number of 50 epochs was selected for training the models with the batch size number of 8. The hyperparameters were the same for all training steps for all models. The overall results are shown in the Fig. 3.

Table 6: Model training results based on ResNet50 with augmented dataset after 50 Epochs

| Parameters | Model Loss | Accuracy |
|---|---|---|
| Model from the scratch | 0.4153 | 0.1842 |
| VGG16 based | 0.5626 | 0.7763 |
| VGG19 based | 0.5657 | 0.7632 |
| ResNet50 based | 0.5574 | 0.6842 |

As it can be seen from the charts, the VGG16 shows a better fit for training and test validation and loss. However, the range of the average value for both metrics are the same for 3 models with the pre-tranined base layers as it can be found in Tab. 6. The loss value for the model from the scratch considering augmented dataset is slightly less than other 3 models, although the accuracy is too low to be worth for accepting the model. ResNet50 shows a less reliable prediction results due to the complexity of the model and low accuracy.

The confusion matrix for three models based on pre-trained CNN models have been illustrated in Fig. 4. The results of the confusion matrix shows that a total number of 76 samples were classified by every model. The model with VGG19 based layer has mis-classified 15 MRI images for no tumor and 3 images for having tumor. The mis-classification of the model based on VGG16 goes for 14 and 3 brain MRI images for no tumor and having tumor respectively. These numbers are relatively changed for ResNet50-based model to 19 and 5.

As a result, we have shown that in the case of having small number of datasets, building a CNN model from the scratch is not recommended at all and will not have an acceptable and reliable results even if the methods for data augmentation are applied for creating a bigger sets of data. An

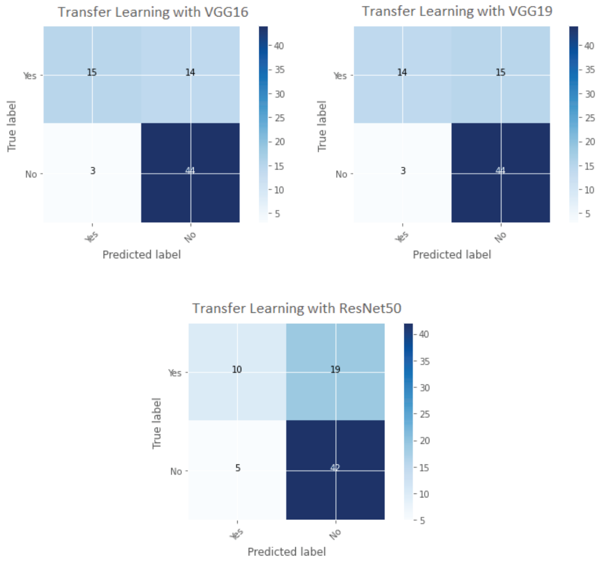


Figure 4: Confusion matrixes for CNN implementation based on pre-trained models

overall analysis for three transfer learning based deep convolutional neural network models on accuracy and confusion matrix results comes to the conclusion that the VGG16 based model shows a better performance than VGG19 and ResNet50 based models for 50 epochs. An improvement of the results however could be expected for more epochs of training.

## Acknowledgments

This article represents the results of my Bachelor's thesis at the FH-Campus Wien, University of Applied Sciences, Vienna. Therefore, I would like to take this opportunity to express my sincere thanks to my supervisor, Dr. Christian Steineder, for his scientific support and for illuminating my path to the completion of this study.

## References

[acc] Kera accuracy metrics. `https://keras.io/api/metrics/accuracy_metrics/`. Accessed: 2022-06-01.

[ada] Kera accuracy metrics. `https://keras.io/api/metrics/accuracy_metrics/`. Accessed: 2022-06-01.

[AS21] Shaveta Arora and Meghna Sharma. Deep learning for brain tumor classification from mri images. In *2021 Sixth International Conference on Image Information Processing (ICIIP)*, pages 409–412, 11 2021.

[aug] Keras image data augmentation. `https://www.tensorflow.org/api_docs/python/tf/keras/preprocessing/image/ImageDataGenerator`. Accessed: 2022-06-01.

[bia] sklearn.preprocessing.labelbinarizer. `https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.LabelBinarizer.html`. Accessed: 2022-06-01.

[cro] Binarycrossentropy. `https://www.tensorflow.org/api_docs/python/tf/keras/losses/BinaryCrossentropy`. Accessed: 2022-06-01.

[dat] Brain mri images for brain tumor detection. `https://www.kaggle.com/datasets/navoneel/brain-mri-images-for-brain-tumor-detecti` Accessed: 2022-06-01.

[GS19] A. Géron and an O'Reilly Media Company Safari. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, 2nd Edition*. O'Reilly Media, Incorporated, 2019.

[HZRS16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. pages 770–778, 06 2016.

[JM21] Sanjaya Jena and Debahuti Mishra. *A Convolutional Neural Network Framework for Brain Tumor Classification*, pages 187–195. 01 2021.

[PG21] Ozlem Polat and Cahfer Güngen. Classification of brain tumors from mr images using deep transfer learning. *The Journal of Supercomputing*, 77:1–17, 07 2021.

[PK21] Aaditya Pundir and Er Kumar. *Brain Tumor Classification in MRI Images Using Transfer Learning*, pages 307–319. 01 2021.

[red] tf.keras.callbacks.reducelronplateau. `https://www.tensorflow.org/api_docs/python/tf/keras/callbacks/ReduceLROnPlateau`. Accessed: 2022-06-01.

[RSK19] Meena R. and R.Shantha Selva Kumari. Classification of mr brain images for detection of tumor with transfer learning from pre-trained cnn models. In *2019 International Conference on Wireless Communications Signal Processing and Networking (WiSPNET)*, pages 508–511, 03 2019.

[WKW16] Karl Weiss, Taghi Khoshgoftaar, and DingDing Wang. A survey of transfer learning. *Journal of Big Data*, 3, 05 2016.

# Machine learning in human healthcare: Optimizing computer-aided detection and diagnosis systems using an x-ray data set of pediatric wrist fractures

Tristan Till[1], Sebastian Tschauner[2], Georg Singer[3], Holger Till[3], and Klaus Lichtenegger[1]

[1]FH JOANNEUM - University of Applied Sciences, Graz, Austria.
[2]Department of Radiology, Division of Pediatric Radiology, Medical University of Graz, Graz, Austria.
[3]Department of Pediatric and Adolescent Surgery, Medical University of Graz, Graz, Austria.

## Abstract

Machine learning opens great possibilities for automating diagnostics in pediatric radiology. However, basic implementations of object detection procedures most likely do not deliver optimal results. With each new object detector, model complexity increases and model customization becomes less transparent, making optimization increasingly difficult for medical professionals. Therefore, this paper will provide an overview of adjustments that can be performed on an existing x-ray dataset with predetermined constraints regarding time and resources.

Firstly, a baseline model was established. Secondly, the data preprocessing, image sizes during training and testing, YOLOv7 configurations and hyperparameters were modified and analyzed regarding their impact on the mean average precision (mAP) of predictions.

Hereby, omitting image preprocessing procedures (+17.65% mAP.5, +18.10% mAP.5:.95) and increasing image sizes during training (+25.51% mAP.5, +39.78% mAP.5:.95) were associated with the highest increases in average precision. Moreover, each category analyzed could achieve improvements in model performance when compared to the baseline. Analyzing these models with Grad-CAM showed differences in saliency maps generated by better and worse models. In general, higher-scoring models seemed to investigate regions surrounding the ground truth more heavily, producing noisier saliency maps than models with lower mean average precision.

In conclusion, making adjustments to a radiological dataset and the corresponding YOLOv7 procedure can significantly impact model performance. This paper provides an overview, on how modular, systematic changes to a YOLOv7 object detection model can affect detection mAP.5 and mAP.5:.95.

## 1 Introduction

Machine learning (ML) has found various kinds of application in medical healthcare, supporting professionals in task automation, information retrieval and decision support. Especially in radiology applications have been developed, adapted and implemented into daily business, in hopes of reaping the benefits of the resulting applications and systems [WS12]. Most notably, computer-aided detection (CADe) and computer-aided diagnosis (CADx) systems assist medical professionals in the interpretation of patient data and scans [SGH$^{+}$22].

CAD systems have a long history, dating back to the late 1950s, when biomedical researchers explored the capabilities of expert systems in medicine, computer programs taking patient data as input and producing a diagnostic output. With technology advancing, early approaches have been optimized and expanded, using artificial intelligence and specialized computer algorithms to increase predictive performance [YT19]. Most commonly, objects in medical images are located and highlighted using bounding boxes or image segmentation [WS12]. Due to the high degree of complexity of the problem at hand, deep learning algorithms are not only preferable to traditional neural networks [Lai15], but also show signs of equal or superior performance in certain fields when compared to medical professionals [JHZ+21]. However, increasing model complexity, data quality reliance, explainability and ethical concerns limit practical implementation from fully flourishing. Every system is only beneficial to medical practitioners, if it is simple, safe and trustworthy [GC17]. Achieving full explainability and trustworthiness, why a machine learning model classified a data point in a certain way, remains difficult. Particularly when neural networks or deep neural networks are concerned, the inner-workings are often too complex for humans to understand [GC17]. Therefore, Explainable Artificial Intelligence methods are utilized to present a machine learning model in a comprehensible fashion. The results of such models can be of visual, textual, or example-based nature [vdVKGV22], [SSL20]. Gradient-weighted class activation mapping (Grad-CAM) represents a model-agnostic adaptation of traditional Class Activation Mapping (CAM) [ZKL+16] not limited to global average pooling. Guided Grad-CAM acts as a hybrid between Grad-CAM and guided backpropagation, combining both methodologies through element-wise multiplication. The resulting method produces visualizations of higher resolution that are class discriminative [SCD+17] [vdVKGV22].

The experiments for this paper analyze a You-Only-Look-Once object detector of the seventh generation (YOLOv7) for detecting pediatric wrist fractures. The dataset and realistic baseline parameters in terms of scope and resources are inherited by a paper published by Nagy. The former contains 20,327 images labelled for nine classes, only two of which, "text" and "fractures", occur in more than 11% of images [NJH+22]. For future reference, values concerning "all classes" target all nine classes of the dataset, "fractures" reference the class fractures, while "imbalanced classes" analyze behaviour for non-text and non-fracture classes. The ablation study will be analyzed and a baseline model for this paper will be established, before presenting research results, all of which are are based on the corresponding bachelor's thesis [Til23].

## 2 Experiments and Results

### 2.1 Ablation Study

The paper "A pediatric wrist trauma X-ray dataset (GRAZPEDWRI-DX) for machine learning" [NJH+22] acts as a reference point in terms of scope and available resources in a realistic application environment. However, the YOLOv5m configuration was substituted by the newest YOLO [RDGF16] variation at time of planning the experiments, YOLOv7 [WBL22]. Dataset and base configuration parameters remained unchanged whenever possible and were only altered for investigative purposes. The former contains labels for a total of nine classes. By default, images were edited using contrast and sharpness enhancing measures. Tab. 1 provides an overview of the resulting baseline YOLOv7 model.

Table 1: Baseline model

| Parameter | Baseline Value |
|---|---|
| Architecture | YOLOv7 |
| Configuration | None |
| Train-Val-Test Split | 15327 - 4000 - 1000 |
| Image Size | 640x640 |
| No. of Epochs | 50 |
| Batch Size | 8 |
| Base Hyperparameters | hyp.scratch.custom.yaml |
| IoU | 0.65 |

### 2.2 Results

During the experiments, six different variations of parameters were tested. These range from altering model architectures without providing a configuration file, including model configurations, varying image sizes during testing and training, omitting preprocessing procedures (Method Editing in Tab. 2) , and adapting the initial learning rate (Method Hyp - lr0 in Tab. 2), as well as momentum (Method Hyp - m inTab. 2) in settings. All variations were trained for the same amount of epochs, batch size and intersection-over-union (IoU) settings. The results are found in Tab. 2. Predictive power is evaluated by analyzing the mean average precision (mAP), valuing the IoU of ground truth and predicted bounding box after a 50% threshold is passed (mAP.5), or weighted depending on the degree of conformity in 5% intervals, ranging from 50% to 100% (mAP.5:.95) [PNDS20, WC22].

# 3 Discussion

## 3.1 Archtictures and Configurations

Using configuration was generally preferable to using no configuration, excluding the YOLOv7-Tiny architecture. Every architecture equally scaled to YOLOv7 or upwards, performed similarly well, with only minor differences to be detected. In the given setting, using the YOLOv7 or YOLOv7-X configurations produced the best results on average. The YOLOv7-Tiny configuration performed the worst across all metrics besides inference time. YOLOv7-W6, YOLOv7-E6, and YOLOv7-D6 configurations have shown to be more performant on the MS COCO dataset, as their architectures were larger, including more layers, gradients and parameters. However, neither of those models performed better than YOLOv7 or YOLOv7-X on average.

As described above, each model was trained for 50 epochs on a batch-size of 8. These values were lower than in comparable studies, like [IRY+22], [WBL22], [KAF+21]. Moreover, the default value in the code of [WBL22] on GitHub (https://github.com/WongKinYiu/yolov7), configured a default number of epochs of 300 with a batch size of 16. Therefore, the results shown in Tab. 2 suggest that those models with larger architectures have not converged before the end of training. In consequence, when paired with a larger number of epochs or batch-size, these configurations may outperform the YOLOv7 or YOLOv7-X configuration. Alternatively, as shown below, modifying certain hyperparame-

Table 2: Resulting mAP for YOLOv7 variations, Models marked with '*' are equivalent to the baseline

| Method | Variation | mAP.5 | mAP.5:.95 |
|---|---|---|---|
|  | Default* | 0,494 | 0,274 |
| No CFG | YOLOv7-Tiny | 0,544 | 0,326 |
|  | YOLOv7* | 0,494 | 0,274 |
|  | YOLOv7-X | 0,455 | 0,259 |
|  | YOLOv7-W6 | 0,473 | 0,262 |
|  | YOLOv7-E6 | 0,405 | 0,214 |
|  | YOLOv7-D6 | 0,445 | 0,252 |
| CFG | YOLOv7-Tiny | 0,499 | 0,291 |
|  | YOLOv7* | 0,494 | 0,274 |
|  | YOLOv7 | 0,56 | 0,337 |
|  | YOLOv7-X | 0,559 | 0,34 |
|  | YOLOv7-W6 | 0,521 | 0,314 |
|  | YOLOv7-E6 | 0,544 | 0,319 |
|  | YOLOv7-D6 | 0,542 | 0,323 |
| Image | 320 — 320 | 0,543 | 0,324 |
|  | 320 — 640 | 0,444 | 0,242 |
|  | 480 — 480 | 0,504 | 0,28 |
|  | 480 — 640 | 0,445 | 0,23 |
|  | 640 — 480 | 0,498 | 0,275 |
|  | 640 — 640* | 0,494 | 0,274 |
|  | 640 — 800 | 0,459 | 0,249 |
|  | 800 — 640 | 0,62 | 0,383 |
|  | 800 — 800 | 0,579 | 0,364 |
| Editing | Edited* | 0,494 | 0,274 |
|  | No Edit | 0,64 | 0,385 |
| Hyp - lr0 | 0.1 | 0,353 | 0,176 |
|  | 0.01* | 0,494 | 0,274 |
|  | 0.001 | 0,554 | 0,348 |
|  | 0.0001 | 0,497 | 0,307 |
| Hyp - m | 0.9 | 0,444 | 0,242 |
|  | 0.937* | 0,494 | 0,274 |
|  | 0.99 | 0,528 | 0,321 |
|  | 0.999 | 0,35 | 0,182 |

ters, like learning rate or momentum, may influence the convergence speed. Thus, this matter may require further investigation before a conclusion can be drawn.

Using only pre-trained weights without a configuration file led to worse results on average. Except for YOLOv7-Tiny, all models experienced losses of around 9.21% to 32.92% regarding the mean average precision. On the other hand, YOLOv7-Tiny scored better when disregarding the configuration. The results showed a similar trend as those above for using configuration files, in a sense that smaller model architectures delivered better results than larger ones. Assuming the hypothesis about too short training times to be true, configuration files could increase the convergence speed. This could explain the shift of the performance peak from medium-sized models, like YOLOv7 or YOLOv7-X, to the smallest architecture provided.

However, the influence of configuration files did not affect all models positively. YOLOv7-Tiny experienced performance increases of 9.02% and 12.03% for mAP.5 and mAP.5:.95, respectively. The corresponding configuration could purposely inhibit model performance for keeping the model as lean as possible. The direct influence on using configuration files regarding convergence speed and performance potential requires further investigation before being conclusive.

### 3.2 Modifying the Dataset

Results for using different image sizes during testing and training were subject to variation more than alterations of the configuration. The model trained on images sized 800x800 and tested on images of 640x640 scored second best on mAP.5 and mAP.5:.95 for all classes. Testing on larger images than those used during training caused models to perform similarly bad to omitting model configuration. Training and testing the model on unedited data showed the largest increases in mAP.5 and mAP.5:.95 for all classes, while simultaneously lowering the inference time necessary for detection.

Increasing the image size for training to 800x800 delivered the best results concerning average precision. Smaller image sizes for training yielded increased performance of object detection averaged across all classes. However, fractures were detected equally or less precise. Using smaller image sizes may have reduced general noise in images during training. This would have facilitated focusing on more recognizable features and achieving higher average precision rates on such than comparable models. More subtle features, e.g. fractures, benefited from images of higher resolution. Additionally, the data suggested that image size for testing and inference time were correlated, causing inference time to rise, when images of higher resolution were used. Training and testing on different resolutions may be a viable solution for time critical applications. The small sample size tested suggested towards YOLOv7 object detection models to be downwards compatible regarding training and testing image sizes. Detecting images of lower resolution than during training produced similar results as using the training size, benefiting from faster inference without compromising quality. On the other hand, upwards compatibility, testing on images with higher resolution than during training, did not seem to be likely.

The preprocessing procedures seemed to inhibit full feature visibility. Average precision for all classes was increased when training and testing with unedited data. However, it remained unclear, to what degree contrast or sharpening contribute individually to the decrease in performance. Applying these procedures separately and analyzing the resulting models may provide further knowledge.

### 3.3 Altering Hyperparameters

Modifying the hyperparameters initial learning rate (lr0) and momentum (m) had significant negative effects on object detection performance for all classes, as seen for lr0=0.1 or m=0.999. On the other hand, adjusting these to benefit the model, minorly influenced all classes, but may yield large performance increases for individual ones. Setting the learning rate to 0.001 produced the highest scoring model regarding mAP.5 for fractures.

An increased initial learning rate led the model to quickly draw conclusions about certain classes. As seen with predicting fractures, if these labels occurred frequently, errors were less prevalent and balanced out over time. However, full model convergence may not have been achieved in a time- or resource-constrained environment. Lower learning rates allowed the object detection model to compensate the imbalanced classes. Lowering this figure to 0.0001 increased model performance for all classes, roughly matching it for only fractures as well, when compared to the baseline. Different training parameters, like epochs or batch size, would most likely require different learning rates for quickest model conversion. Nonetheless, an initial learning rate of 0.001 accommodated the circumstances best.

A momentum of 0.99 yielded the best scores for recall, mAP.5, and mAP.5:.95 for all classes, as well as fractures viewed separately, for the current model configuration. However, the overall increase in percentage was significantly larger for less common classes than for fractures. Values of 0.9 or 0.999, on the other hand, performed worse regarding the same three values. Both experienced larger drop offs in average precision for less common classes. Slightly increasing momentum showed greater model performance overall. The momentum parameter showed signs of being more sensitive towards lesser, and more stable towards more frequent features.

However, simply selecting the best scoring variation for each method analyzed will most likely not result in the optimal model. As e.g. [IRY+22] shows, various components interact with and influence each other. A simple shift in the base configuration may have caused the model to show a different behavior as the one analyzed above. Fully understanding the relationship between the methods mentioned previously goes beyond the scope of this paper and requires further research on this topic.

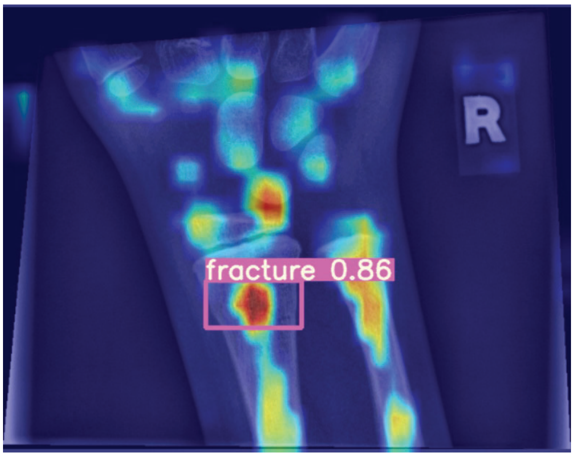


Figure 1: Baseline saliency map

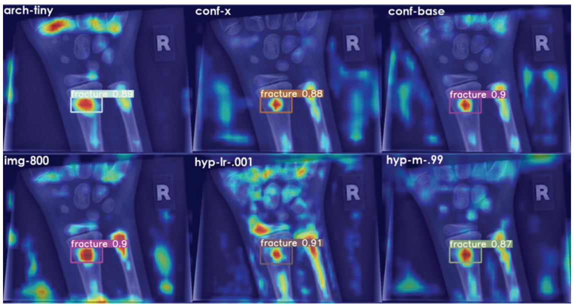


Figure 2: Better models saliency maps

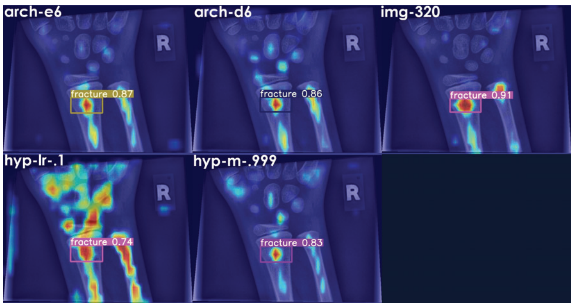


Figure 3: Worse models saliency maps

### 3.4 Grad-CAM Analysis

Object detection models that predicted object locations with higher precision, accuracy and mean average precision, are described as better models. Grad-CAM was used to analyze model performance, beyond the resulting precision values. Therefore, saliency maps of different object detection models, grouped by performing better or worse than the baseline (Fig. 1), were generated for the same image.

In general, the models, which scored better in previous experiments (Fig. 2), produced noisier saliency maps than their worse counterparts. High attention was directed towards the area inside the resulting bounding box. However, some models produced regions of high focus outside the perimeters of the bounding box, without being classified. The worse models (Fig. 3) showed signs of being less noisy than the better models, except for hyp-lr-.1. Attention was primarily directed towards the actual fracture, rather than the surrounding regions. This seemed counterintuitive, as these models predict their bounding boxes with less confidence, when compared to the others.

The saliency maps may have hinted towards an imbalance in accuracy and precision for the object detection models tested. The worse models produced comparably precise saliency maps, directing their focus towards few regions with great intensity. Better models took more features into consideration, some even being analyzed indepth. However, the example image was comparably easy to predict, as e.g. a model with a test score of 0.35 mAP.5 by hyp-m-.999, generated a correct bounding box. For true positives, as in said image, worse models produced appealing results, as model prediction was accurate and precise. As soon as the prediction outcome was uncertain and not immediately verifiable, these became increasingly problematic.

Object detection models should primarily be precise and secondarily accurate. Using object detection models in practice that focus on a small amount of image regions, of which none contain a fracture, despite one being present, made it prone to detecting false negative results. These may have caused the comparably low scores in mean average precision while testing. Moreover, producing widespread saliency maps allowed medical professionals to reassure model validity. They generated key reference points, in which models directed more attention to certain regions than others. These could have assisted radiologists in checking detection outcomes more effectively and efficiently.

## 4 Conclusion

Despite all parameters analyzed producing at least one variation with better results than the baseline, adjusting the dataset to fit model requirements improved the mean average precision during testing most significantly. Moreover, sharpness and contrast enhancements prohibit models from learning features as effectively and efficiently than unedited images. In addition, increasing the image size during testing caused model mAP.5 and mAP.5:0.95 to increase by 25.51% and 39.78%, despite being tested on images of smaller scale.

Using larger architectures did not automatically cause the model to perform better. Especially in tight environments, where batch-size and number of epochs were kept constant, these models seemed to fail to reach conversion in the available time. If resource-constraints cannot be improved, adjusting hyperparameters like learning rate or momentum may accelerate the process. Furthermore, customized hyperparameters for the baseline increased mAP by up to 27.01%. When analyzed with Grad-CAM, higher-scoring models produced noisier saliency maps than lower-scoring ones. Therefore, better models seemed to take more image regions into consideration, while worse models filter out large parts of images. In consequence, models with lower mAP generated saliency maps with significantly less widespread parameters, which, if detection failed, could cause them to omit relevant regions prematurely. Applying these procedures separately and analyzing the resulting models may provide further knowledge about the efficacy of machine learning in human healthcare, especially pediatrics.

## References

[GC17] Chloe Gui and Victoria Chan. Machine learning in medicine. *University of Western Ontario Medical Journal*, 86(2):76–78, 2017.

[IRY+22] Iza Sazanita Isa, Mohamed Syazwan Asyraf Rosli, Umi Kalsom Yusof, Mohd Ikmal Fitri Maruzuki, and Siti Noraini Sulaiman. Optimizing the hyperparameter tuning of yolov5 for underwater detection. *IEEE Access*, 2022.

[JHZ+21] Dakai Jin, Adam P Harrison, Ling Zhang, Ke Yan, Yirui Wang, Jinzheng Cai, Shun Miao, and Le Lu. Artificial intelligence in radiology. In *Artificial Intelligence in Medicine*, pages 265–289. Elsevier, 2021.

[KAF+21] Ryosuke Kuwana, Yoshiko Ariji, Motoki Fukuda, Yoshitaka Kise, Michihito Nozawa, Chiaki Kuwada, Chisako Muramatsu, Akitoshi Katsumata, Hiroshi Fujita, and Eiichiro Ariji. Performance of deep learning object detection technology in the detection and diagnosis of maxillary sinus lesions on panoramic radiographs. *Dentomaxillofacial Radiology*, 50(1):20200171, 2021.

[Lai15] Matthew Lai. Deep learning for medical image segmentation. *arXiv preprint arXiv:1505.02000*, 2015.

[NJH+22] Eszter Nagy, Michael Janisch, Franko Hržić, Erich Sorantin, and Sebastian Tschauner. A pediatric wrist trauma x-ray dataset (grazpedwri-dx) for machine learning. *Scientific Data*, 9(1):1–10, 2022.

[PNDS20] Rafael Padilla, Sergio L Netto, and Eduardo AB Da Silva. A survey on performance metrics for object-detection algorithms. In *2020 international conference on systems, signals and image processing (IWSSIP)*, pages 237–242. IEEE, 2020.

[RDGF16] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[SCD+17] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017.

[SGH+22] Wilson Silva, Tiago Gonçalves, Kirsi Härmä, Erich Schröder, Verena Carola Obmann, María Cecilia Barroso, Alexander Poellinger, Mauricio Reyes, and Jaime S Cardoso. Computer-aided diagnosis through medical image retrieval in radiology. 2022.

[SSL20] Amitojdeep Singh, Sourya Sengupta, and Vasudevan Lakshminarayanan. Explainable deep learning models in medical image analysis. *Journal of Imaging*, 6(6):52, 2020.

[Til23] Tristan Till. Machine learning in human healthcare: Computer-aided detection and diagnosis systems in radiology on the basis of a pediatric wrist trauma x-ray dataset. Bachelor's thesis, University of Applied Sciences Joanneum Graz, 2023.

[vdVKGV22] Bas HM van der Velden, Hugo J Kuijf, Kenneth GA Gilhuijs, and Max A Viergever. Explainable artificial intelligence (xai) in deep learning-based medical image analysis. *Medical Image Analysis*, page 102470, 2022.

[WBL22] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. *arXiv preprint arXiv:2207.02696*, 2022.

[WC22] Luke Wood and Francois Chollet. Efficient graph-friendly coco metric computation for train-time model evaluation. *arXiv preprint arXiv:2207.12120*, 2022.

[WS12] Shijun Wang and Ronald M Summers. Machine learning and radiology. *Medical image analysis*, 16(5):933–951, 2012.

[YT19] Juri Yanase and Evangelos Triantaphyllou. A systematic survey of computer-aided diagnosis in medicine: Past and present developments. *Expert Systems with Applications*, 138:112821, 2019.

[ZKL+16] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2921–2929, 2016.

1
1
0
0
0
0
1
1
0

# Scientific Computing 2023

## Track: Luftfahrt

*Philip Kirschnek:* **Design, Construction and Fluid Flow Calculation of a Centrifugal Compressor** ........ 72

*Elias Müller und Fabian Pircher:* **Development of a calculation algorithm for the power design of a vertical icing wind tunnel (IWT)** ........ 82

*Mario Rammelmüller:* **Implementation and comparison of various algorithms for attitude and heading reference systems of drones in MATLAB®** ........ 88

*Julian Gräf:* **Numerical investigation of the effects of a winglet configuration on the lift and drag coefficients of the SD7062 wing profile**. ........ 100

*Sarah Luckeneder:* **Numerische Simulation von Tragflächenvereisungen** ...... 110

*Maximilian Angermayr:* **Three-dimensional simulation of in-flight icing on an aerofoil using Ansys FENSAP-ICE** ........ 114

# Design, Construction and Fluid Flow Calculation of a Centrifugal Compressor

Philip Kirschnek

FH JOANNEUM – University of Applied Sciences, Institute of Aviation, Alte Poststraße 149, 8020 Graz, Austria

## Abstract

This paper deals with the dimensioning, design and fluid flow calculation of a radial compressor of aero engines. Specifically, in the first part, a centrifugal compressor is designed using thermodynamic equations and via thermal turbo machinery formulas with Matlab® [Mat21]. In order to facilitate the operation for the user, a GUI is resorted to, which plots the velocity triangles, key figures of the centrifugal compressor and other parameters, depending on about 18 input values.

The dimensions calculated in Matlab are then transferred directly to an .xlsx file, which is also linked to CATIA® [Das12]. With CATIA, an impeller model is designed which is automatically updated and follows the calculated geometric dimensions. The CATIA model is mainly used for visualization purposes, so that the User of the Matlab GUI gets to see what the via Matlab calculated impeller could possibly look like. In the last step, the flow calculation of a centrifugal compressor is performed using Ansys® [ANS22]. This flow calculation is intended to show whether the analytical methodology of compressor design comes close to numerics and how well these two methods can be compared.

Since the CATIA model is not adopted by Ansys, due to simplification purposes, a new impeller geometry is created with the BladeGen-Tool. TurboGrid is used for meshing the flow channel and CFX for inserting boundary conditions, as well as for the evaluation of the results.

## 1 Introduction

Centrifugal compressors, such as the one shown in Figure 1, are used in many areas. Whether in automotive engineering as so-called turbochargers/centrifugal compressors with a volute casing, or in gas engine technology and thermal turbo machinery as the final compressor stage after an axial compressor, as well as in many other industrial areas.

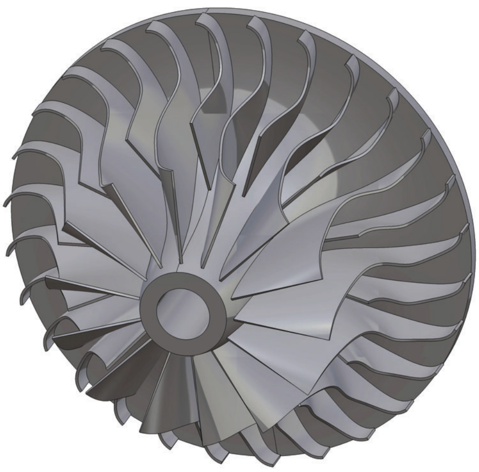


Figure 1: Impeller of the centrifugal compressor – designed with CATIA V5-6R2012

In this paper, however, mainly aviation is relevant, thus the use of centrifugal compressors in aircraft engines is given special importance. In particular, a compact calculation tool for centrifugal compressor geometry is to be designed, which can also be used for teaching purposes. Knowing how a centrifugal compressor operates plays a role in many technical areas and is indispensable in science.

The operation of the centrifugal compressor is explained in this chapter, as well as the dependence of the blade curvature on the flow velocities and angles.

## 1.1 Operating Principle of the Centrifugal Compressor

The entire compressor consists of an impeller, as shown in Figure 1, a casing and a diffuser.

The rotor, i.e. the impeller, is bladed. As displayed in Figure 2, the fluid flows into the impeller coming from the left, ideally swirl-free.

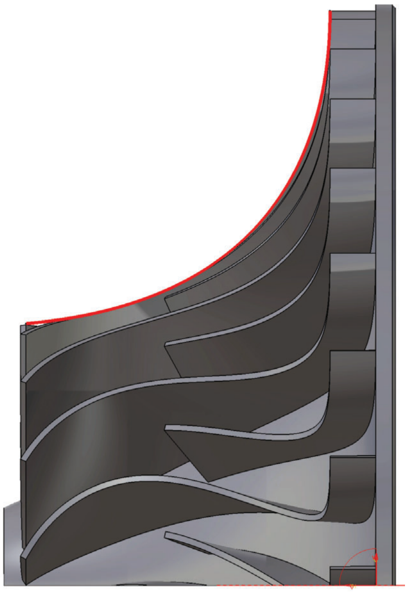


Figure 2: Side view of the impeller

In Figure 2, the curve in red at the top of the blades indicates the geometry of the compressor, the housing, which defines the outside. The fluid experiences acceleration in the radial direction and also an increase in pressure. The high stage pressure ratio, i.e. the compression ratio of a compressor stage, is due to the centrifugal force experienced by the fluid [Bor12, p. 13]. In order to keep the inflow cross-section as open as possible for the flowing medium, every second blade is used in a shortened form and starts further back in the blade channel. The short blade elements are usually called *splitter blades.* The centrifugal compressor performance is improved by a stronger blade curvature of the front blade elements. These blades are called *inducer blades.* The exact blade geometry is discussed in chapter 1.2.

Once the fluid has passed through the impeller, the *diffuser* follows. In the diffuser, the flow is to be decelerated. The high kinetic energy is converted into pressure energy, the compression pressure and thus the pressure ratio increase further. The efficiency of the centrifugal compressor depends heavily on the diffuser.

## 1.2 The Influence of the Flow Triangles

The flow triangles, both of the compressor inlet and outlet, are instrumental in determining the blade geometry.

A velocity triangle is generally used to represent the flow velocities and their direction to show the operating principle of the compressor, as well as to determine the blade geometry needed to meet the requirements. The blade curvature thus depends on the flow angles. These will be explained in more detail below.

A distinction is made between a backward curvature, radially ending blade and a forward curved blade geometry. The exact case distinctions are shown in Figure 3, where $\beta$ is defined as the blade angle.

Typically, in turbocharger and aero engine technology, an impeller has a forward curvature at the inlet and a backward curvature or even a radial end at the outlet.

A typical flow triangle for the compressor outlet can be seen in Figure 4. Such a velocity triangle is common for a backward curvature. Shown in the graph are the *relative velocity* $w$, *circumferential velocity* $u$, *meridional velocity* $c_m$, *absolute velocity* $c$, and the *absolute velocity in circumfer-*

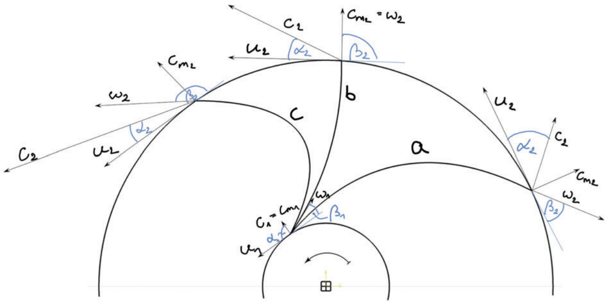


Figure 3: Impeller blade curvature for different velocity triangles. a: backward curved blade $\beta < 90°$; b: radial ending blade $\beta = 90°$; c: forward curved blade $\beta > 90°$ [Eck53, p. 281]

*ential direction* $c_u$. The index 2 in Figure 4 refers to the compressor outlet.

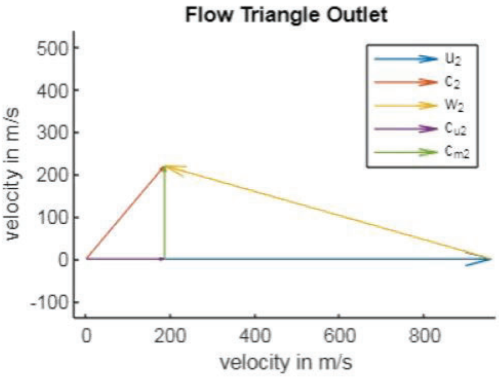


Figure 4: Flow triangle at the impeller outlet – Index 2 and its parameters:
circumferential velocity u; relative velocity w; meridional velocity $c_m$; absolute velocity c; absolute velocity in circumferential direction $c_u$

It should be noted that the calculated blade angle does not correspond to the actual flow angle. This is due to the forces acting in the rotating system, centrifugal forces on the one hand, Coriolis forces on the other.

Normally, the fluid flow should be parallel to the blade at all times, but due to the Coriolis force, the particles experience an acceleration opposite to the direction of rotation and are deflected from their path. Thus, the individual air particles are pushed to the left side of the trailing vane, for reference see Figure 5. In Figure 5, $\omega$ indicates the direction of rotation. Negative pressure behind the leading blade and positive pressure in front of the trailing blade results inside the flow channel. Equalizing flows and vortices cause an increase in flow velocity on the negative pressure side, and a deceleration on the positive pressure side. Frictional losses increase and finally flow separation occurs. The resulting detachment area can no longer be used by the flow, and a constriction of the channel occurs. In Figure 5, the design channel width is marked as $h$, the constricted channel width as $h*$. The velocity profile within the flow channel is also evident.

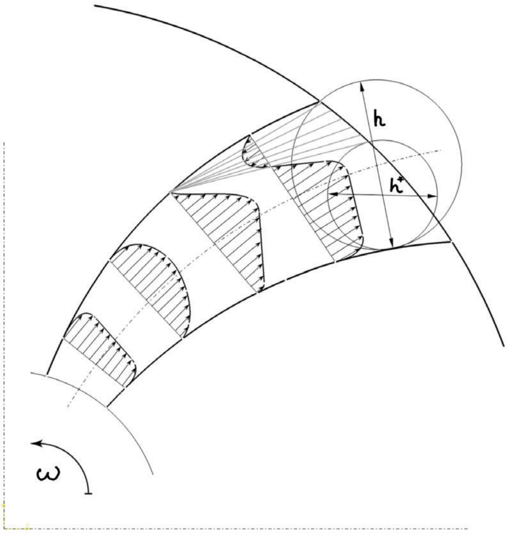


Figure 5: Actual velocity distribution in the flow channel with $\omega$ being the angular velocity [Eck53, p. 297]

The further narrowing of the flow channel results in a flow angle that no longer corresponds to the blade angle. As a result, the compressor cannot convert the theoretically possible power. This power loss is described by the *underperformance factor* $\mu$ [Pal18, p. 61]. The underperformance factor indicates the power loss due to separation and, according to [Eck53, p. 296], is determined as shown in Equation (1):

$$\mu = \frac{1}{1 + \frac{\pi}{2} \cdot \frac{\sin(\beta_2)}{z \cdot \left(1 - \frac{D_1}{D_2}\right)}} \tag{1}$$

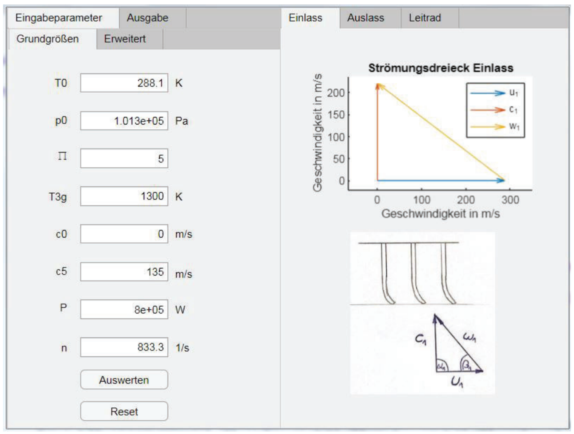
The parameter $z$ indicates the number of blades, the angle $\beta_2$ is the blade angle at the outlet, the ratio $\frac{D_1}{D_2}$ corresponds to the *inlet diameter* divided by the *outlet diameter*.
Of course, to describe the efficiency of compressors, a lot more factors need to be taken into consideration. However, the already thoroughly mentioned underperformance factor is one of the most popular in the compressor sector. The points discussed should give a brief overview about impellers at first glance.

# 2 Methods

The Methods chapter is divided into three parts. First, it describes how the calculation of the centrifugal compressor is performed in Matlab R2021a. Then, the parameterized design of the impeller on CATIA V5-6R2012 is discussed. The flow calculation of the designed system is implemented using Ansys Workbench 2022R2.

## 2.1 Matlab

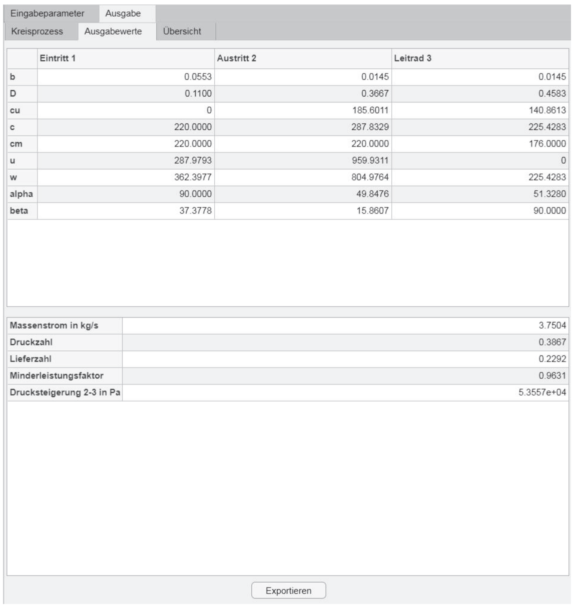
Matlab is ideal as software for calculating mathematical problems, in this case for compressor estimation, as well as for visualizing the calculated data.
Using this program, a complete conceptual calculation of the compressor impeller was performed. This includes the flow angles and velocities, any dimensions, i.e. the geometries of the inlet, outlet and guide vane/diffuser, as well as key figures that characterize the centrifugal compressor based on its efficiency. The exact calculation procedure will not be discussed in detail here, as it has been carried out and explained in detail in the work by [Kir23].
With the help of a GUI (=*Graphical User Interface*) the calculation shall be facilitated for the user. The user is able to vary 18 input parameters, according to which the compressor is subsequently designed with the already briefly described calculation rule. The input options include, for example, the *number of blades* $z$, the desired *shaft power* $P$, *ambient pressure* $p_0$ and *temperature* $T_0$, the *airspeed* $c_0$, various efficiencies and quite a few other parameters.
The startup window of the GUI is visualized in Figure 6.
**Notice:** The GUI is not translated into english.



T0 288.1 K
p0 1.013e+05 Pa
Π 5
T3g 1300 K
c0 0 m/s
c5 135 m/s
P 8e+05 W
n 833.3 1/s

Figure 6: Graphical User Interface for the impeller design

On the right side, in Figure 6, the flow triangles of inlet, outlet and diffuser are shown. In the upper diagram, the calculated vectors, and in the diagram below, an interpretation guide of the velocity triangles.
The left side of the GUI in Figure 6 is dedicated to the input and output of numerical data. As already described, the input tab contains the user's target data, distinguishing between the *basic variables* and the *extended* tab. The most important characteristic values are to be defined in the basic variables window; further data, rather less relevant for the standard user, in the extension window.
The output window itself is divided into three elements. In the first window, a cycle process of a typical Turboprop engine is displayed, including the calculated values in tabular form. The cycle process will not be discussed in detail here. Of greater importance is the second output window, see Figure 7. The lower table shown in the Figure lists the most important key figures for characterizing the centrifugal compressor. These include the *mass flow rate* and the *underperformance factor*. The latter has already been explained. Other key figures are also given,



| | Eintritt 1 | Austritt 2 | Leitrad 3 |
|---|---|---|---|
| b | 0.0553 | 0.0145 | 0.0145 |
| D | 0.1100 | 0.3667 | 0.4583 |
| cu | 0 | 185.6011 | 140.8613 |
| c | 220.0000 | 287.8329 | 225.4283 |
| cm | 220.0000 | 220.0000 | 176.0000 |
| u | 287.9793 | 959.9311 | 0 |
| w | 362.3977 | 804.9764 | 225.4283 |
| alpha | 90.0000 | 49.8476 | 51.3280 |
| beta | 37.3778 | 15.8607 | 90.0000 |

| | |
|---|---|
| Massenstrom in kg/s | 3.7504 |
| Druckzahl | 0.3867 |
| Lieferzahl | 0.2292 |
| Minderleistungsfaktor | 0.9631 |
| Drucksteigerung 2-3 in Pa | 5.3557e+04 |

Figure 7: Output values of the impeller calculation

and a detailed explanation can again be found in [Kir23].

The upper table in Figure 7 shows the velocities listed in chapter 1.2, the dimensions, i.e. diameter and blade height, as well as the relevant flow angles. The unit system used is the *SI-System*, only the angles deviate from this convention and are given in *degrees*.

The *Export*-Button saves all calculated data and graphs. Tables are saved as Excel files, graphs as .jpg files. Thus, it is possible to make a number of parameter changes to compare afterward.

After evaluating the data, the most important parameters are transferred to CATIA for the impeller design.

## 2.2 CATIA

The impeller was designed using CATIA V5-6R2012. Since the compressor impeller has a relatively complex geometry, the model was initially created in *Generative Shape Design*, CATIA's surface design tool. Subsequently, the geometry was converted into a solid in *Part Design*. The created surface model of the impeller can be seen in Figure 8.

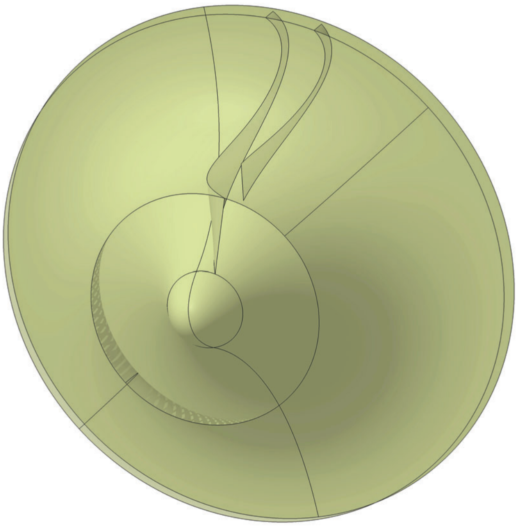


Figure 8: Impeller shape model

The inner surface represents the impeller hub, the outer surface the tip, thus the outer boundary of the blades. The blades themselves can be seen between the impeller hub and the slightly-transparent housing, i.e. the outer surface. It is noticeable that the right blade is made shorter, as a so-called *splitter blade*. If this blade was not shortened, the inflow cross-section would be considerably reduced. Thus, it is shortened in order to keep the flow channel open as much as possible.

With the help of circular patterns, the next step was to transform the impeller into a solid, as shown in Figure 1. As already mentioned, this model is automatically updated after changing the input parameters, thus after the Matlab calculation.

The CATIA model is mainly used for visualization purposes, so that the user of the Matlab GUI gets an overview about the calculated geometries and angles and will not be used within Ansys due to simplification purposes.

## 2.3 Ansys

The simulation software Ansys enables the fluid flow and strength calculation of the centrifugal compressor. If only a CFD-(= Computational Fluid Dynamics) calculation of the impeller is desired, the use of the programs *BladeGen*, *TurboGrid* and *CFX* is sufficient. If one is further interested in a strength calculation, the additional use of the applications *Mechanical-Model* and *static-mechanical analysis* is necessary. The overall structure and linkage of the programs in the Ansys *Workbench* user environment is shown in Figure 9.

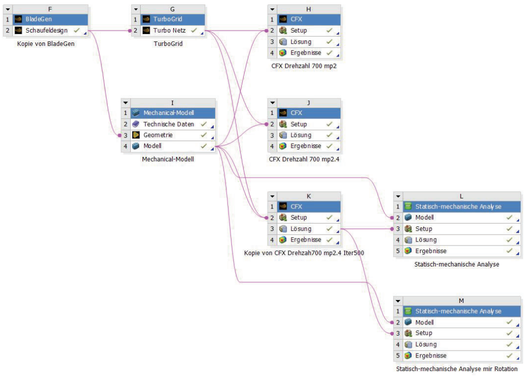


Figure 9: Ansys Workbench structure (Images used courtesy of ANSYS, Inc.)

The *BladeGen* environment allows the design of the centrifugal compressor. The model is not adopted by CATIA, as the use of the BladeGen tool allows the performance of the centrifugal compressor to be better determined.
The two meshing programs *TurboGrid* and *Mechanical-Model* are used respectively for the meshing of the flow channel and the blade itself. *CFX* solves the flow calculation under the specification of initial conditions, while the *static-mechanical analysis* performs the strength calculation of the compressor impeller.

**BladeGen** In the BladeGen [ANS22] environment, the blade geometry in particular is defined. Decisive influencing parameters on the initial model are, as so often, the blade angles at the inlet and outlet, as well as the blade thickness at the hub and at the tip of the blade. The basic design window can be found in Figure 10. Here, the blade angle is shown in the bottom left of the diagram, while the blade thickness is shown to the right. The two diagrams above show the blade cross-section and the blades in plan view on the right.
This compressor model represents the basis for a successful flow calculation. The smallest inaccuracies can have severe consequences, leading either to wrong results, or to error messages in the solver.

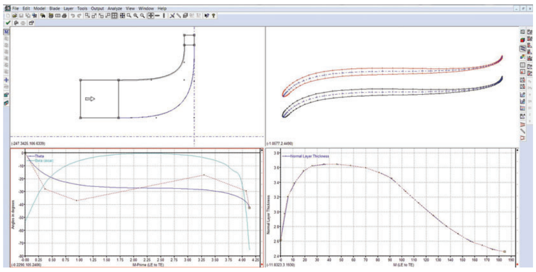


Figure 10: Ansys BladeGen User-Interface (Images used courtesy of ANSYS, Inc.)

**TurboGrid** The program TurboGrid [ANS22] is responsible for the meshing of the flow channel. The blade itself is not meshed. The rotational symmetry of the impeller in relation to the axis of rotation can be used elegantly, and only a single flow channel has to be meshed. The whole mesh, see Figure 11, is the result of arranging together the individual elements.

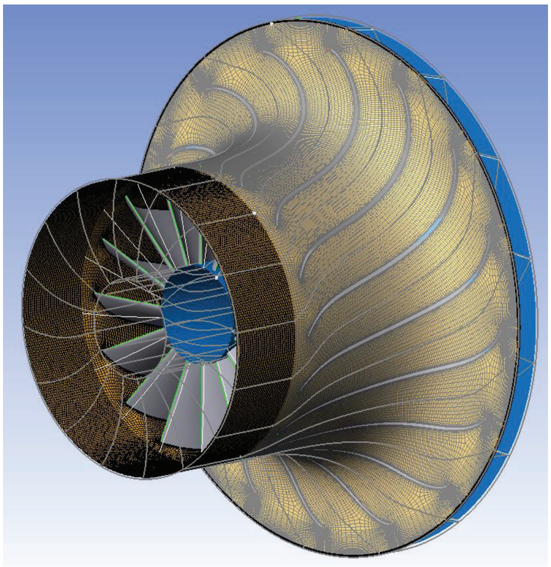


Figure 11: Meshed flow channel (Images used courtesy of ANSYS, Inc.)

The element size and number is crucial for the accuracy of the results. During this flow calculation, a maximum element number of only 500,000 could be realized, due to the limitation of the student version. The results are nevertheless meaningful and illustrative.

**CFX** The CFX [ANS22] environment is divided into *Setup*, *Solution* and *Results*. In the setup, the boundary conditions are set. In this case, a rotational speed is to be specified at which the compressor rotates. Likewise, the mass flow rate at the outlet, as well as the relative pressure to the ambient air at the inlet, are specified.

Before starting the *Solver*, it is recommended to define so-called *Monitor Points*. Monitor points allow the output of certain physical data at a mesh region during the solving process. This has the great advantage that it can already be decided during the solution process, whether the output data is realistic or not. For example, when plotting the averaged flow velocity over the outlet cross-section, an enormous value or even a very low value would indicate an error.

Figure 12 shows the plot of the outlet pressure monitor point. The x-axis shows the number of iterations, the y-axis the pressure.

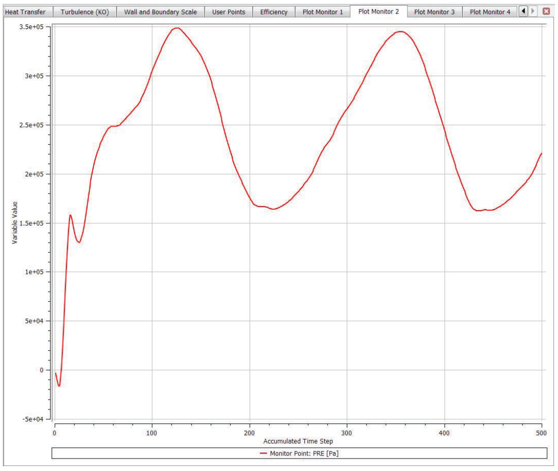


Figure 12: CFX-Solver – pressure monitor
Shows the static pressure at the impeller outlet over the past iterations of the calculation (Images used courtesy of ANSYS, Inc.)

An oscillation of the pressure curve is noticeable, the solution does not converge in this case, the reason being either a too high mass flow rate or a too high rotational speed of the impeller. Usually, the pressure curve should approach a specific value for the pressure.

This pressure curve can nevertheless be used for the evaluation of the results.

## 3 Results

In the *CFX-Post* area, the results of the flow simulation are evaluated. A predefined report allows the evaluation of all relevant data of the compressor. The pressure/entropy/velocity distributions are displayed over the relevant sections, as well as the Mach number and a few other parameters/-factors. Figure 13 shows the velocity distribution along the impeller, the *M-axis* is set as the axis of rotation. The very low velocity in the outlet region (right) of the leading blade is striking. The low flow velocity can be attributed to the channel vortex and the aerodynamic processes in the flow channel described in chapter 1.2.

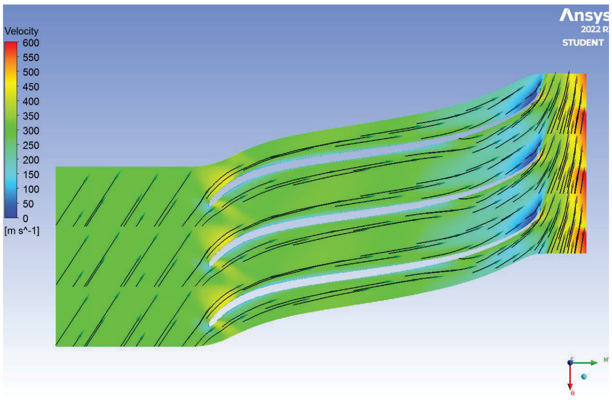


Figure 13: Velocity distribution over the impeller depth at 50 % blade height (Images used courtesy of ANSYS, Inc.)

The streamlines of the velocity in the channel system also indicate, how the flow behaves in the flow channel and can be seen in Figure 14.

Streamlines represent curves that are tangential to the velocity field. Thus, the orientation of the

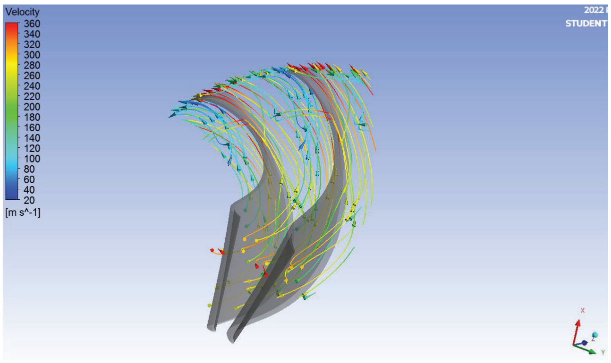


Figure 14: Velocity streamlines in the flow channel (Images used courtesy of ANSYS, Inc.)

streamlines at each individual point corresponds to that of the velocity vector at that point. They are used to illustrate the flow of a fluid.

In the real word, streamlines are often visualized in a wind tunnel, with the help of a fog. This method is very well known in the field.

Last but not least, the pressure curve along the compressor depth determines whether the compressor ultimately fulfills its function of compressing the air. The velocity distribution already gives a more detailed insight into the flow behavior and the functioning of the impeller. For complete characterization, the pressure distribution over the impeller depth is shown in Figure 15.

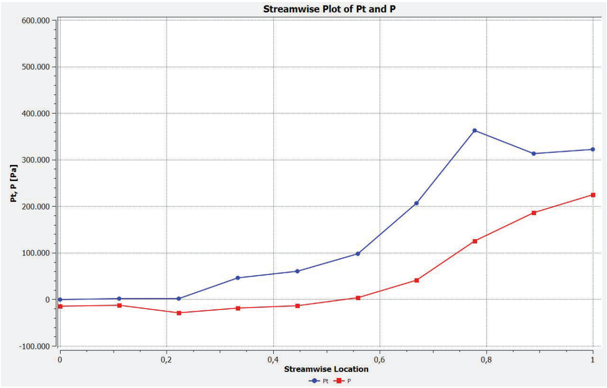


Figure 15: Pressure distribution at 50 % blade-height (Images used courtesy of ANSYS, Inc.)

In Figure 15, the red lower curve shows the course of the static pressure, the upper blue curve the total pressure. The x-axis shows the impeller depth, the y-axis the pressure relative to the ambient pressure. The diagram depicts that at an ambient pressure of 1 bar, the *total pressure ratio* $\Pi_g$ is about 4.2. A subsequent diffuser could increase the static pressure considerably.

Ansys is capable of writing the performance data to a table. The compressor calculated using Matlab was designed as well as possible in Ansys, the evaluation of the performance data, see Table 1, shows a striking similarity of the analytical and numerical calculation methods. The lower required *drive power* $P$ in kW, which was determined by Ansys, is due to the reduced mass flow rate $\dot{m}$, the total pressure ratio additionally due to the reduced rotational speed $n_K$.

Table 1: Performance data comparison: Ansys - Matlab of the parameters:
rotational speed $\mathrm{n_K}$; mass-flow-rate $\dot{\mathrm{m}}$; volume-flow-rate $\dot{\mathrm{V}}$; shaft power P; volumetric efficiency $\varphi$; total pressure ratio $\Pi_\mathrm{g}$; total temperature ratio $\mathrm{T_{2g}}/\mathrm{T_{1g}}$

| | Ansys | Matlab | |
|---|---|---|---|
| $n_K$ | 700.00 | 833.33 | $^1/_\mathrm{s}$ |
| $\dot{m}$ | 3.20 | 3.75 | $^\mathrm{kg}/_\mathrm{s}$ |
| $\dot{V}$ | 2.73 | 3.40 | $^{\mathrm{m}^3}/_\mathrm{s}$ |
| $P$ | 655.65 | 800.00 | kW |
| $\varphi$ | 0.22 | 0.23 | |
| $\Pi_g$ | 4.19 | 5.00 | |
| $\frac{T2g}{T1g}$ | 1.68 | 1.65 | |

The results thus show that analytical calculations are quite relevant for conceptual design, since the deviation from the data of a flow simulation is comparatively small.

In Table 1, $\frac{T_{2g}}{T_{1g}}$ is the *total temperature ratio* of the outlet divided by the inlet, whereas $\varphi$ stands for the *volumetric efficiency.*

The volumetric efficiency represents the ratio of the real volume flow to the theoretically maximum possible [Eck53, p. 300].

## Summary and Outlook

This paper focused on the design, construction and flow calculation of a centrifugal compressor. The Matlab design tool and the automatically updating CATIA model are fully functional.

The determined flow results of the Matlab program are also to be evaluated as quite positive and deviate only slightly from the Ansys flow calculation. Thus, the results are not to be regarded as unrealistic. A deviation of the results is natural, the analytical methods nevertheless come close to the numerical calculation. The deviation amounts to only a few percent, thus showing a high accuracy.

This work is an excerpt of the bachelor thesis by [Kir23] and represents only a fraction of the whole project; in case of greater interest, please refer to this work.

## References

[ANS22] ANSYS, INC. Student (Version 2022 R2), 2022.

[Bor12] Oliver Borm. *Instationäre numerische Untersuchung der aerodynamischen Rotor-Stator-Interaktion in einem Radialverdichter*. Dissertation, Technische Universität München, München, 2012.

[Das12] Dassault Systemes. CATIA (Version V5-6R2012), 2012.

[Eck53] Bruno Eckert. *Axial- und Radialkompressoren*. Springer, Berlin, Heidelberg, 1953.

[Kir23] Philip Kirschnek. *Auslegung, Konstruktion und Strömungsberechnung eines Radialverdichters*. Bachelorarbeit, FH JOANNEUM Luftfahrt/Aviation, Graz, 2023.

[Mat21] MathWorks. MATLAB (Version R2021a), 2021.

[Pal18] Josef Palla. Gasstrahltriebwerke 1. Skriptum, HTBLA Eisenstadt - Flugtechnik, 2018.

# Development of a calculation algorithm for the power design of a vertical icing wind tunnel (IWT)

Müller Elias[1] and Pircher Fabian[1]

[1]FH JOANNEUM – University of Applied Sciences, Institut Luftfahrt/Aviation, Alte Poststraße 147, 8020 Graz, Austria

## Abstract

This project deals with the development of a calculation algorithm for the power-related design of the components of an icing wind tunnel for researching icing phenomena on wing profiles at the FH JOANNEUM in Graz. By considering pressure loss calculation, general fluid dynamics, heat flow analysis and general thermodynamics, a calculation algorithm is developed, which calculates the required power quantities of the wind tunnel elements such as the air conditioning unit and the fan depending on input variables such as the general channel cross-section, the test chamber cross-section and the test flow velocity of the wind tunnel medium. The results are calculated as a function of input parameters selected by the user and are displayed by using 3D plots, which makes it easier for the user in the project decision phase to commit to a concept or to consider the effects caused by changes during the progress of the project. This calculation is programd using the software MATLAB 2021b.

## 1 Introduction

Wind tunnels have been used since the beginning of the 20th century to draw conclusions about the behaviour of assemblies, constructions, vehicles and aircraft in reality from model tests. In aviation, individual assemblies of the aircraft or even entire aircraft models are usually tested in wind tunnels for their aerodynamic properties and for occurring environmental conditions such as icing. In order to be able to simulate all flight conditions, several test runs in different wind tunnels may be necessary. The Institute of Aviation at the FH JOANNEUM - University of Applied Sciences in Graz currently operates a horizontal icing wind tunnel to generate and research icing phenomena on wing models. However, this channel no longer corresponds to the current state of the art and is therefore to be redesigned as part of a project. This work deals with the development of a calculation algorithm for the performance design of a closed wind tunnel concept.

### 1.1 Types of wind tunnel

In general, two wind tunnel models are distinguished on the basis of flow recirculation. In the open wind tunnel, the flow medium is taken from the environment, accelerated by a fan, smoothed by flow calming and released back into the environment after the test chamber. According to [BRP99, p. 27], this type of wind tunnel requires more energy in operation, generates more noise emissions and the flow medium properties are strongly dependent on the existing environmental conditions. In the closed wind tunnel, where the flow medium is returned to the fan after the test chamber, thus creating a circuit, the flow conditions are dependend on the environmental conditions, less noise is produced and the control of the medium temperature is easier to implement, which is essential for the operation of an icing wind tunnel. In Fig. 1 we show the con-

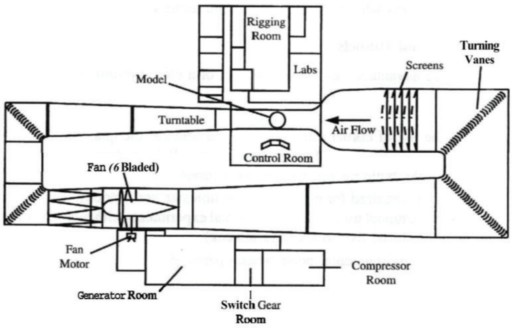


Figure 1: Top view of a closed wind tunnel [BRP99, p. 26]

struction of a closed wind tunnel.

A further subdivision of wind tunnels is made by the type of flow through the test chamber. A horizontal wind tunnel model is one in which the flow medium flows horizontally through the test chamber. In icing wind tunnels, where water droplets have to be injected into the flow medium to form ice, this means that the path of the water droplets is influenced by gravity. In order to make the supercooled water droplets hit the test model to create icing, an exact combination of nozzle position and flow velocity is required. A vertical wind tunnel, on the other hand, has a test area with vertical flow, which eliminates the effects of gravity on the droplet trajectory.

## 1.2 Final wind tunnel concept

In order to be able to apply and test the calculation algorithm developed for the design, a concept is required on which the calculations are to be based. This was worked out in the concept phase of this project. In addition, an attempt was made to eliminate all negative effects of the current wind tunnel at FH JOANNEUM. The design phase was done in six steps. In each step, a new design was constructed with a different arrangement of the individual components with respect to the previous model. Subsequently, the disadvantages of each concept were analysed and evaluated. The evaluation resulted in the final design shown in fig (2). The heat exchanger is located on the right side of the duct due to the required large distance with respect to the injection nozzle system. This has the advantages of a more homogeneous temperature distribution in the duct cross-section and fewer problems regarding component icing due to the injected water droplets. The fan is located in the upper duct cross-section. The narrowing of the channel cross-section at the bottom ensures as much space as possible between the bridge transition installed at FH JOANNEUM and the icing channel. In addition, in contrast to the current icing channel, there is a heating element in the lower horizontal segment after the measuring chamber. If this is switched on instead of the heat exchanger, the icing channel can be quickly deiced. The injection is located directly after the calming zone. This positioning is again chosen because of the other icing of the stilling elements. As a general rule, elements that are important for operation should not be mounted in the immediate vicinity of the measuring chamber and the injection system, as this would lead to an increased risk of icing and thus to a restriction in operation or disturbances of the flow field.

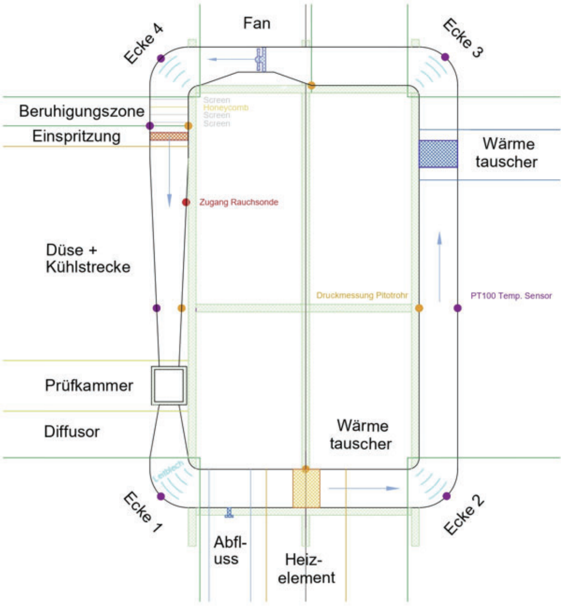


Figure 2: Wind tunnel concept as the basis for the power calculations.

Fig. 2 shows us on the left side (from top to bottom): corner 4, the calming section, the nozzle-system, a rejuvenation, the test chamber, a diffusor and corner 1. On the right side: corner 2 and 3 and the heat exchanger. On the bottom is a drainage and the heating element and on the top you can find the fan.

## 2 Methods of calculation

In order to fully design a concept the power quantities of the wind tunnel components are required, among other things. The three main components of an icing wind tunnel are the fan to accelerate the flow, the air conditioning unit to cool the flow medium and thus generate icing conditions and a heating coil to defrost the wind tunnel between two test runs in order to achieve proper flow conditions.

### 2.1 Fan power

The required fan power is calculated by taking into account the volume flow and the pressure loss occurring in the path of the tunnel.

$$P_{Fan} = \dot{V} \cdot \Delta p \tag{1}$$

The volumetric flow rate depends on the test chamber cross-sectional area and on the flow velocity, if the flow conditions are below $Ma = 0.3$ and thus the simplified incompressible continuity equation may be assumed. The total pressure loss $\Delta p$ is calculated using the equations (2-4, [BK21, p. 227]) the sum of all individual and pipe friction losses generated by the tunnel elements in the flow and by the tunnel walls.

$$\Delta p = \sum \Delta p_{indiv} + \sum \Delta p_{pipe} \tag{2}$$

The individual pressure loss contributions $p_{indiv}$ are calculated by multiplying the individual, component-dependent pressure loss coefficient by the density of the flow medium and the square of the flow velocity.

$$\Delta p_{indiv} = \frac{1}{2} \cdot \rho \cdot v^2 \cdot \zeta \tag{3}$$

The pipe friction pressure losses depend on the cross-sectional geometry (hydraulic diameter $D_h$), the pipe friction coefficient $\lambda$, the length of the tunnel section $L$ and also on the density $\rho$ and the square of the prevailing flow velocity $v$.

$$\Delta p_{pipe} = \lambda \cdot \frac{L}{D_h} \cdot \rho \cdot \frac{v^2}{2} \tag{4}$$

### 2.2 Air conditioning unit power

The calculation of the required air conditioning unit power is a conservative estimation by adding the required individual single powers for the actual cooling process and for the compensation of the stationary heat flow loss as well as the power introduced into the system by the fan during the operation of the icing wind tunnel. The cooling process itself is assumed to be an adiabatic process, so the required individual power is calculated with

$$P_C = \frac{Q_C}{t_C} \tag{5}$$

where $t_C$ is the desired cooling time and

$$Q_C = m \cdot c_p \cdot \Delta T \tag{6}$$

the amount of heat to be removed, which is calculated by the mass $m$, the heat capacity $c_p$ and the temperature difference $\Delta T$ to be achieved. The heat flow loss to be compensated by the air conditioner is calculated with the equations used by [BS19, p. 35-37] for the heat transfer at a multi-layer wall (7-8).

$$\dot{Q}_{stat} = kA(\theta_1 - \theta_2) \tag{7}$$

where $\theta_1$ and $\theta_2$ are the inner and the outer temperatures, $A$ is the radiation surface and where for the heat transfer coefficient $k$

$$\frac{1}{kA} = \frac{1}{\alpha_1 A_1} + \sum_i \frac{\delta_i}{\lambda_{mi} A_{mi}} + \frac{1}{\alpha_2 A_2} \tag{8}$$

is applied. The variable labels can also be taken from Fig. 3.

The consideration of the power input generated by the fan is done by introducing a standard efficiency $\eta$, with which the power to be

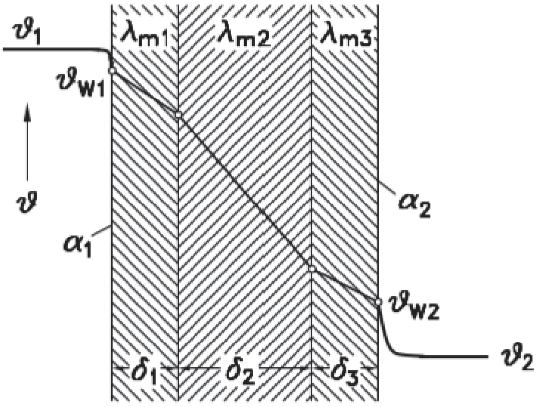


Figure 3: Heat transfer through a multi-layer wind tunnel wall [BS19, p. 37]

compensated by the air conditioner $P_{AC}$ can be calculated. The variables used in equation 9 can be taken from previous equations.

$$P_{AC} = P_C + \dot{Q}_{stat} + P_{Fan}(1 - \eta) \cdot (-1) \qquad (9)$$

The estimation of the required total output of the air conditioning unit is finally made by adding these individual powers. Since under real conditions the heat loss during the cooling process would be unsteady, as it depends on the inner and outer temperature, which would change during the cooling process, the result of the air conditioning unit power is only a conservative estimate.

### 2.3 Heating register power

Also, the calculated heating register power using the equations (5-6) is an estimation of the required power amount due to the assumption of adiabatic process operation. However, since the heat losses occur in the heating phase and help to heat the channel, this calculated value is sufficient for the design of the heating coil.

## 3 Software implementation

The development of the calculation algorithm includes the digitalisation of all required formulas and equations including all relevant conditional dependencies (If-/Else conditions) that lead to a change in the program flow. As shown in Fig. 4, the program is fed by a parameter initialisation file with input data that must be specified by the user beforehand. These include, for example, the geometric dimensions of the wind tunnel cross-section and the test chamber or the lengths of the individual wind tunnel sections. Furthermore, material parameters such as the thermal conductivity of the wall layers or the temperature and type of flow medium must be entered. All other parameters such as Reynolds numbers, kinematic viscosities, the density of the flow medium, etc. are calculated by the program itself at the required positions. Individual parameters can be entered additionally by the user if he does not wish to have the data calculated or if values for these parameters have already been fixed.

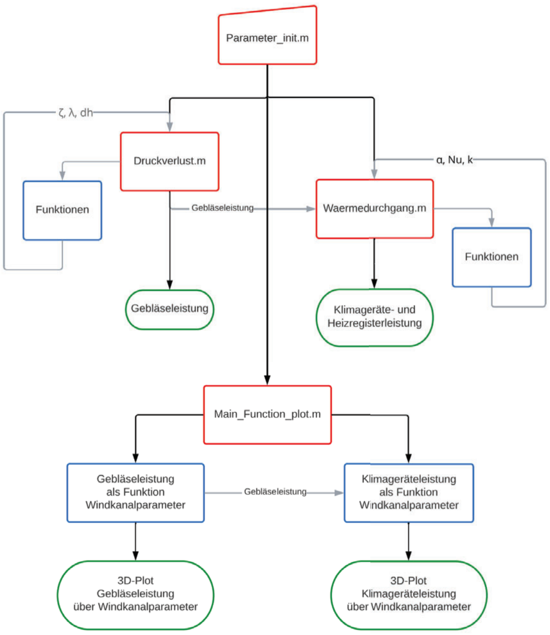

Parameter_init.m
ζ, λ, dh
Druckverlust.m
Funktionen
Gebläseleistung
Waermedurchgang.m
α, Nu, k
Funktionen
Gebläseleistung
Klimageräte- und Heizregisterleistung
Main_Function_plot.m
Gebläseleistung als Funktion Windkanalparameter
Gebläseleistung
Klimageräteleistung als Funktion Windkanalparameter
3D-Plot Gebläseleistung über Windkanalparameter
3D-Plot Klimageräteleistung über Windkanalparameter

Figure 4: Flowchart of the calculation process (software: LUCIDCHART); The names in the boxes correspond to the program name of the MATLAB files; Colorcode: red - main program, blue - function, green - result, grey - variables handed over;

The main programs then independently calculate the output variables of the power for the fan, for the air conditioner and for the heating coil. Since the output of the air conditioner depends on the fan power, the fan power is calculated first. In the main programs, the algorithm runs through various functions which, for example, serve to determine the correct Nusselt number for the current flow case for the calculation of the heat transfer coefficient $\alpha$. All functions are called independently and pass the parameters to the main program at the required positions.
By running another main program, it is possible for the user to display the results of the powers as a function of certain input parameter configurations (test chamber height and test chamber width or test cross-section and test flow velocity). This makes it easier for the project team to recognise and analyse the effects on the power quantities of any parameter changes during the decision-making phases. To program the calculation code MATLAB 2021b was used.

## 4 Results

After the developed calculation program has been executed, the user is provided with the required power values in MATLAB Command Window. These values only apply to the concept for a closed vertical icing wind tunnel developed in the concept phase and to the initial material and input data. After the execution of the additional program, the performance parameters are also displayed above the input configurations. These results are also only valid for the elaborated concept in sec. 1.2. The jumps in power contained in the 3D plots for certain parameter configurations (see fig 5) can be explained by the use of individual equations and functions in the limit of their validity, such as in the selection of the correct pipe friction coefficient $\lambda$ as a function of the hydraulic diameter $D_h$, or in the calculation of the individual pressure loss coefficients $\zeta$. In those formulas and functions, the correlation of the parameter configurations (test chamber height and test chamber width or test cross-section and test flow velocity) plays an important role in relation to the processing of the calculation program and thus to the formula actually used for the calculation. Fig. 5 shows a potential relationship for the flow medium air at −20 °C between the fan power and the mentioned input parameter configurations of the test section cross-section and the maximum occurring flow velocity.

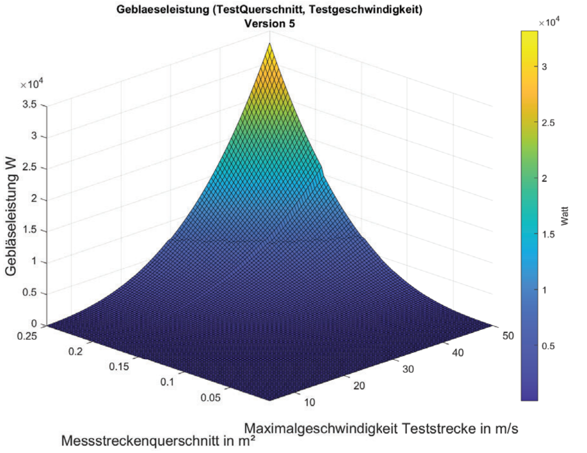


Figure 5: Potential output of the fan power over the input parameters: the configuration test section cross-section and the flow velocity for the flow medium air at −20 °C (software: MATLAB 2021b)

## 5 Summary and Outlook

The aim of this project was to develop a calculation algorithm for the design of a new vertical icing wind tunnel concept for the FH JOANNEUM in order to investigate and research icing phenomena on aircraft wing models. The problems occurring on the current model regarding the influence of gravitational effects on the droplet trajectory were eliminated by choosing a vertical test chamber. Furthermore, the variability of the flow velocity of the wind tunnel medium was thus also established, which allows the user to carry out tests with different flow velocities. By digitalising and integrating the formulas described in section (2) into a calculation algorithm, the required power quantities of the wind tunnel components can be calculated, varied and specified at the push of a button. The project management is given a quick overview of the required component capacities and thus also of the possible costs arising from the realisation of a selected wind tunnel model through the parameterised input by means of an input file. The use of 3D plots allows the project team to compare several wind tunnel configurations and therefore to decide on a wind tunnel configuration even very late in the project schedule. The selection process was made clearer and easier by the calculation program and the output of results via 3D plots. In addition, the program offers scope and quick results for short-term changes to the wind tunnel concept during the entire course of the project. Since the results are based on a large number of conservative information assumptions, the performance values obtained also correspond to conservative values and not to those of the real case. In order to improve the algorithm in the future and thus obtain accurate results, the conservative assumptions mentioned could be replaced by real occurring data. This requires a revision of the calculation algorithm to improve the accuracy of the performance values.

## Acknowledgments

We would like to thank our two project supervisors, Wolfgang Hassler PhD and Andreas Tramposch PhD from the FH JOANNEUM, who gave us the opportunity to work on this project and who supported us with their technical expertise in the fields of fluid mechanics and thermodynamics and thus contributed significantly to the success of this project.

## References

[BK21] Sabine Bschorer and Konrad Költzsch. *Technische Strömungslehre: Mit 262 Aufgaben und 31 Beispielen*. Springer Vieweg, Wiesbaden, Germany, 12th edition, 2021.

[BRP99] Jewel B. Barlow, William H. Rae, and Alan Pope. *Low-speed wind tunnel testing*. Wiley-Interscience, USA, 3. edition, 1999.

[BS19] Hans Dieter Baehr and Karl Stephan. *Wärme- und Stoffübertragung*. Springer Vieweg, Berlin, Germany, 10. edition, 2019.

# Implementation and comparison of various algorithms for attitude and heading reference systems of drones in MATLAB®

Mario Rammelmüller[1]

[1]FH JOANNEUM – University of Applied Sciences, Institute Aviation, Alte Poststraße 149, 8020 Graz, Austria

February 28th, 2023

## Abstract

Due to the increasing popularity of drones in commercial, military, and recreational areas, various companies and research institutions are intensely engaged in research and development of new technologies in the field of small unmanned aerial systems. For example, work is being done on parachute rescue systems for complex industrial drones, which are designed to protect the aerial system from a hard impact on the ground in case of a crash. The parachute is activated as soon as a certain criterion, such as exceeding a predetermined angle of inclination of the drone, is met.

The angle of inclination of a drone is calculated based on the roll and pitch angle. As it is not possible to directly measure the attitude angles of an aerial system, they are calculated in attitude and heading reference systems. These systems consist of various sensors and a data processing center, in which different algorithms estimate the actual attitude angles of the aircraft. An accurate calculation of the angles is essential to protect the drone in uncontrollable flight situations and to prevent erroneous activation of the rescue system.

In order to ensure the versatility in their operating environments and uncomplicated implementation of the parachute rescue systems of the project partner of this article, three simple algorithms are studied and compared. The three algorithms, namely the complementary filter, the Madgwick filter and the Kalman filter, are implemented in MATLAB® and evaluated in a variety of tests. These filters are limited to processing data from accelerometers and gyroscopes.

The results show significantly more accurate calculations from the implemented filters compared to the currently used algorithm of the project partner in tests over short time periods. In the long-term test with more complex flight maneuvers, only the Madgwick filter can provide usable results and an improvement to the algorithm of the project partner.

## 1 Introduction

The determination of the attitude angles of a drone or any general aircraft plays a vital role in aviation technology. Accurate measurement of these angles is essential for instrument flight and autonomous flying using an autopilot system. Additionally, an autopilot system can assist the pilot in controlling the aircraft, such as compensating for external disturbances like wind gusts. However, the challenge in determining the attitude is that the Euler angles, which represent the aircraft's orientation relative to the reference coordinate system, cannot be directly measured. Instead, mathematical algorithms, also known as filters, are used to calculate these angles from input data from various sensors. Such a system of sensors and data processing center is called an Attitude and Heading Reference System (AHRS), which can contain a variety of sensors such as accelerometers, gyroscopes, magnetometers, barometers, or receivers for a Global Navigation Satellite System (GNSS) and is usually equipped with high-performance processing chips.

In the case of a small Unmanned Aerial System (sUAS), such as a drone, smaller and lighter AHRSs are preferably installed due to their weight. These often consist of a less powerful microcontroller, accelerometers, and gyroscopes. The combination of these two sets of sensors is also called an Inertial Measurement Unit (IMU). The advancement of these sensors into low-cost microchips, known as Microelectromechanical Systems (MEMSs) or MEMS sensors, allows modern IMUs to weigh only a few grams, enabling their use on small drones. However, they have the disadvantage of providing highly noisy data that needs to be optimized through filters. The accelerometers measure the linear accelerations in the axis direction in the body-fixed coordinate system, while the gyroscopes measure the angular rates around the three axis directions in the body-axis system.

Limiting the system to an IMU offers several advantages if precise determination of the yaw angle, height, and exact position is not required. Firstly, it allows the use of simpler filters that require less processing power. Secondly, it removes sources of interference, allowing the sUAS to be used in environments with magnetic interference and limited GNSS reception and still obtain valid attitude angles. Additionally, the incorporation of a magnetometer would require calibration every time the system is turned on.

## 1.1 Goal of this article

This article provides a concise conclusion of the author's first bachelor's thesis [Ram23]. Therefore, the information given in this article is directly adopted from the thesis which is only published to a limited extent. For more in-depth knowledge about the topic, feel free to contact the author. As this work is part of a research project by FH JOANNEUM and the company Drone Rescue Systems (DRS), it will be limited to the processing of gyroscope and acceleration sensor data with simple filters for some of the reasons mentioned earlier. The company DRS develops parachute systems for drones that operate autonomously, meaning they work independently of the autopilot data. These parachute systems are triggered at a predetermined tilt angle of the sUAS and protect the drone from a hard impact on the ground in case of loss of control by the pilot. The goal of this article is to study various filters regarding their performance to find an improvement over the currently used algorithm. Specifically, the complementary filter, the Madgwick filter, and the Kalman filter are evaluated.

## 1.2 Structure of this article

The next chapter (1.3) is a literature review of the sources that provide a basis for this article. This is followed by the description of the methods used in this work in chapter 2. It contains an explanation of the mathematical foundations (2.1) required for the subsequent description of the implemented filters (2.2). In chapter 3, the qualitative and quantitative results of the algorithms are presented. Chapter 4 discusses the used methods and results and finally chapter 5 provides a summary of this article.

## 1.3 Literature review

Several scientific articles have already been published with the aim of examining the performance of various filters in different test setups. The Master's thesis of Joshua B. Milam provides a basis for this article, in which already known filters are compared with newly developed algorithms [Mil18]. His research not only focuses on processing data from one IMU, but also examines the benefits of a second installed IMU.

The fusion of the two IMUs is intended to result in significantly more accurate results. It turns out that the filters he developed, which use both IMUs to calculate the angle of inclination, provide significantly more accurate results. The already known filters, specifically the complementary filter, Madgwick filter, Mahony filter, and the Kalman filter, behave very similarly in terms of their accuracy, with slightly more accurate results from the Kalman filter.

Madgwick, 2010, describes his newly developed Madgwick filter and compares it with the Kalman filter, which is integrated in the test sensor [Mad10]. Madgwick, 2010, argues that this filter is less computationally expensive than the Kalman filter and is effective even at low frequencies [Mad10, p. 1]. These aspects are intended to open up new possibilities for real-time attitude estimations of systems with limited computing power and even surpass the Kalman filter in accuracy.

Since the Kalman filter is one of the most important filters ever developed, there are also a variety of publications on this topic [ZM09, p. xix]. Poulouse et al., 2019, describes an algorithm for estimating the position in buildings [PEH19]. A simple Kalman filter is used to calculate the roll and pitch angle from data of acceleration sensors and gyroscopes of a smartphone. Li and Wang, 2013, presents an adaptive Kalman filter for processing data from MEMS sensors, which is intended to increase performance and solve several problems of the Kalman filter [LW13].

# 2 Methods

The mathematical equations of the algorithms from some of the previously mentioned sources were programmed in MATLAB® for this study. The filters are evaluated based on measured data from the project partner DRS. To perform a representative comparison, various realistic flight tests were conducted. When comparing the filters, the raw data from the external MEMS sensors mounted on the drone are read and processed by the filters. The calculated attitude angles are then compared with the data from the drone's internal autopilot. It is assumed that the drone's AHRS works well enough to use its calculated angles as a reference for the comparisons.

The algorithms' performance is qualitatively evaluated using various plots. Furthermore, a quantitative analysis is conducted as well. In this analysis, the root mean square error (RMSE) of the individual Euler angles, according to equation (1), is calculated. The variable $\hat{y}$ represents the filter's estimate of the current iteration, and $y$ represents the reference value. This deviation is a typical quantitative measure of the AHRS's performance [Mad10, p. 19].

$$\text{RMSE} = \sqrt{\sum_{i=1}^{n} \frac{(\hat{y}_i - y_i)^2}{n}} \tag{1}$$

As a symmetric drone does not need to differentiate between pitch and roll angles, and the project partner DRS has defined a single critical angle as the trigger criterion for the parachute system, an additional measure is determined to compare the filters. This measure is the tilt angle of the drone $\zeta$, which is described in equation (2). The index $f$ represents the filter used for calculating the Euler angles.

$$\zeta_f = \arccos(\cos(\phi_f) \cdot \cos(\theta_f)) \tag{2}$$

## 2.1 Conventions and mathematical basis

To enable uniform processing of the input data, it is necessary to establish a convention for the used coordinate system. In this work, the Euler angles are described as the orientation of the body-fixed coordinate system relative to the co-moving North-East-Down (NED) coordinate system. The measured data from the external IMU on the drone are processed in the body-fixed coordinate system, which may lead to differences in the algebraic signs of the equations compared to other publications. The distance between the

IMU and the center of mass of the drone is taken into account by subtracting the resulting centrifugal accelerations. The installation angle of the IMU is also corrected to ensure that the measurements are consistent with the body-fixed coordinate system. The body-fixed system of the drone and the IMU is illustrated in figure 1.

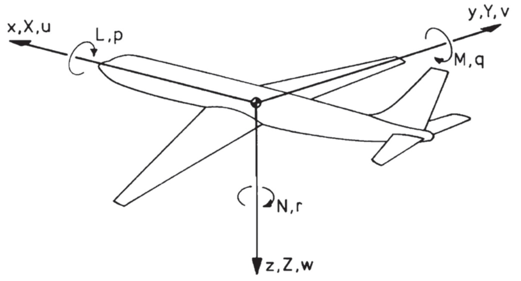


Figure 1: Depiction of the body-fixed coordinate system using an aircraft [BAL13, p. 9]

The mathematical equations of the implemented filters in MATLAB®, are based on quaternions. Representing the attitude angles as a quaternion is another way to describe the body-fixed coordinate system relative to the reference system. A quaternion $q$ is a hyper-complex number of rank 4 which is well suited for use as a rotation operator [Kui99, p. 104]. Representing the attitude angles as a quaternion offers mathematical advantages, such as a higher numerical performance [BM12, S. 254]. It also avoids the singularity of the Euler angle representation at the pitch angle $\theta$ of 90°. At this pitch angle, it is no longer possible to differentiate between the roll and yaw angle. Since the transformations in the source code are implemented using built-in functions of MATLAB®, the explanation of the basic mathematical equations is omitted. For more in-depth information regarding quaternions and their algebraic properties, [Kui99] is recommended.

## 2.2 Implemented filters

The studied filters described in the following chapters are all implemented as three-dimensional algorithms. However, due to neglecting the magnetic field data, the heading $\psi$ is not accurately calculated. It is not possible to determine the initial value of the heading from the available data, resulting in an offset in the calculated yaw angles. This can be ignored for the purposes of this study, as only the pitch angle $\theta$ and roll angle $\phi$ are required for the calculation of the tilt angle $\zeta$.

### 2.2.1 Complementary filter

The first implemented filter is the complementary filter. As the name suggests, this filter complements two components to the value of 1. Specifically, these are the component resulting from the integration of the angular rates and the component calculated from geometric relationships of the acceleration vectors. The integration of the angular rates calculates attitude angles that are very accurate over short time periods, but experience a steadily increasing offset from the reference value due to the integration of measurement errors. Calculations using the acceleration data provide inaccurate Euler angles due to the heavily noisy data, but they are not affected by long-term drift. The complementary filter utilizes the properties of both estimates by applying a low-pass filter to the calculations based on the accelerations and a high-pass filter to the integrated angular rates [Mil18, p. 15].

After the initialisation of the unit quaternion as a starting point for the integration, follows the calculation loop over the data points of the measurements. In the first step, the quaternion rate $\dot{q}$ is calculated from the angular rates and the quaternion of the previous iteration according to equation (3). This is based on [Kui99, p.264] and [Mad10, p.6]. The quaternion rate contains the angular velocities of the Euler angles and is therefore marked with the index $\omega$.

$$\dot{q}_{\omega,i} = \frac{1}{2} \begin{bmatrix} 0 & -p_i & -q_i & -r_i \\ p_i & 0 & r_i & -q_i \\ q_i & -r_i & 0 & p_i \\ r_i & q_i & -p_i & 0 \end{bmatrix} \begin{bmatrix} q_{1,i-1} \\ q_{2,i-1} \\ q_{3,i-1} \\ q_{4,i-1} \end{bmatrix} \tag{3}$$

Subsequently, the integration of the quaternion rate is performed in equation (4). Firstly, the quaternion rate is multiplied by the time step $\Delta t$ and then added to the previous quaternion. This subsequently represents the component of angular velocities in the complementary filter, again marked with the index $\omega$.

$$q_{\omega,i} = q_{i-1} + \dot{q}_{\omega,i} \cdot \Delta t_i \tag{4}$$

Furthermore, the estimation of the Euler angles is determined by the geometric relationships of the acceleration vectors, as shown in equation (5). These calculations assume that the Earth's gravity is the only acceleration acting on the acceleration sensors [Mil18, p.15]. This is not the case in an accelerated flight which leads to additional deviations in the calculations using this method. Another source of disturbance for this estimation of Euler angles is the heavily noisy data from the MEMS acceleration sensors. The Euler angles are transformed into a quaternion $q_{a,i}$ with the index $a$ for acceleration, after the calculation. Since the yaw angle cannot be calculated from the accelerations, it is set to zero.

$$\begin{bmatrix} \psi_{a,i} \\ \theta_{a,i} \\ \phi_{a,i} \end{bmatrix} = \begin{bmatrix} 0 \\ \text{atan2}(-a_{x,i}, a_{z,i}) \\ \text{atan2}(a_{y,i}, a_{z,i}) \end{bmatrix} \tag{5}$$

Subsequently, the sensor fusion with the complementary filter is performed as shown in equation (6). The equation is based on [Mil18, p.15], but is adapted to the use of quaternions. The calculation parameter of the complementary filter $k$ specifies the factor by which the component from the angular rates $q_\omega$ enters the equation. The quaternion from the acceleration data $q_a$ is considered with the complementary factor. Finally, the calculated quaternion is normalized and transformed back into Euler angles.

$$q_i = k \cdot q_{\omega,i} + (1-k) \cdot q_{a,i} \tag{6}$$

### 2.2.2 Madgwick filter

The second algorithm implemented in this work is the Madgwick filter, which also relies on quaternions. The equations in this chapter are taken from [Mad10]. For further background information and derivations of the equations, it is recommended to study the original publication. Similar to the complementary filter, an initial condition in the form of a unit quaternion is defined for the Madgwick filter. Then the calculation loop follows over all data points, with the current iteration indicated in the form of the index $i$.

The first step is to calculate the quaternion rate $\dot{q}_{\omega,i}$, as seen in equation (3). The Madgwick filter uses a gradient descent algorithm to calculate the current quaternion [Mad10, p. 7]. For this reason, the objective function of the Madgwick filter $f$, which is to be minimized, is defined over the quaternion of the previous iteration and the current acceleration data. For the sake of clarity, the index for the previous iteration $i-1$ is neglected in equation (7) and (8) for the quaternion components.

$$f_i = \begin{bmatrix} 2(q_2q_4 - q_1q_3) - a_{x,i} \\ 2(q_1q_2 + q_3q_4) - a_{y,i} \\ 2(\frac{1}{2} - q_2^2 - q_3^2) - a_{z,i} \end{bmatrix} \tag{7}$$

Additionally, the Jacobian $J$ of the objective function is calculated in equation (8).

$$J_i = \begin{bmatrix} -2q_3 & 2q_4 & -2q_1 & 2q_2 \\ 2q_2 & 2q_1 & 2q_4 & 2q_3 \\ 0 & -4q_2 & -4q_3 & 0 \end{bmatrix} \tag{8}$$

Using the previously determined objective function and its Jacobian, the objective function gradient can be calculated as shown in equation (9).

$$\nabla f_i = J_i^T f_i \tag{9}$$

The calculation of the quaternion of the current iteration is ultimately performed using equation (10). Within the parentheses, the estimated derivative of the quaternion is calculated and integrated over the time interval $\Delta t$. The calculation parameter of the Madgwick filter $\beta$ is utilized to tune the filter, indicating the extent to which the acceleration data is weighted in the Madgwick filter. Prior to transformation into Euler angles, the quaternion is normalized.

$$q_i = q_{i-1} + \left( \dot{q}_{\omega,i} - \beta \frac{\nabla f_i}{||\nabla f_i||} \right) \Delta t_i \tag{10}$$

### 2.2.3 Kalman filter

The third and final filter examined in this work for its performance is the Kalman filter. In this article, the linear Kalman filter based on [PEH19] is used. The Kalman filter also operates in the quaternion space, which is why the unit quaternion defined by equation (11) is initialized for the quaternion of the Kalman filter ($x$). However, this initial condition does not simultaneously represent the quaternion of the first iteration, hence the index 0.

$$x_0 = \begin{bmatrix}1 & 0 & 0 & 0\end{bmatrix}^T \tag{11}$$

In addition, some matrices are initialized before the calculation loop. These include the error covariance matrix of $x$ ($P$) according to equation (12), the observation matrix ($H$) according to equation (13), the process noise covariance matrix ($Q$) according to equation (14), and the measurement noise covariance matrix ($R$) given in equation (15). In the matrices Q and R, one can recognize the calculation parameters of the Kalman filter ($\alpha$). The matrix $Q$ essentially describes how strongly the values of the acceleration sensors can be trusted. The opposite is provided by the matrix $R$, which indicates the weighting of the angular rates in the Kalman filter.

$$P_0 = \begin{bmatrix}1 & 0 & 0 & 0\\0 & 1 & 0 & 0\\0 & 0 & 1 & 0\\0 & 0 & 0 & 1\end{bmatrix} \tag{12}$$

$$H = \begin{bmatrix}1 & 0 & 0 & 0\\0 & 1 & 0 & 0\\0 & 0 & 1 & 0\\0 & 0 & 0 & 1\end{bmatrix} \tag{13}$$

$$Q = \alpha_1 \begin{bmatrix}1 & 0 & 0 & 0\\0 & 1 & 0 & 0\\0 & 0 & 1 & 0\\0 & 0 & 0 & 1\end{bmatrix} \tag{14}$$

$$R = \alpha_2 \begin{bmatrix}1 & 0 & 0 & 0\\0 & 1 & 0 & 0\\0 & 0 & 1 & 0\\0 & 0 & 0 & 1\end{bmatrix} \tag{15}$$

The system model matrix ($A$) is calculated in the calculation loop according to equation (16).

$$A_i = \begin{bmatrix}1 & 0 & 0 & 0\\0 & 1 & 0 & 0\\0 & 0 & 1 & 0\\0 & 0 & 0 & 1\end{bmatrix} + \frac{1}{2}\Delta t_i \begin{bmatrix}0 & -p_i & -q_i & -r_i\\p_i & 0 & r_i & -q_i\\q_i & -r_i & 0 & p_i\\r_i & q_i & -p_i & 0\end{bmatrix} \tag{16}$$

Subsequently, the previously explained calculation of the Euler angles based on the normalized acceleration data is performed. For completeness, this is shown again in equation (17). These Euler angles are then transformed into the quaternion from the acceleration data ($z$) in the form of a column vector.

$$\begin{bmatrix}\psi_{a,i}\\\theta_{a,i}\\\phi_{a,i}\end{bmatrix} = \begin{bmatrix}0\\\text{atan2}(-a_{x,i}, a_{z,i})\\\text{atan2}(a_{y,i}, a_{z,i})\end{bmatrix} \tag{17}$$

The next step is the prediction of $x$ according to equation (18) and $P$ according to equation (19). These are calculated *a priori*, that is, with their values from the previous iteration. They are marked with a superscript "-". The calculation of $x_i^-$ represents the integration of the angular rates.

$$x_i^- = A_i x_{i-1} \tag{18}$$

$$P_i^- = A_i P_{i-1} A_i^T + Q \tag{19}$$

The next step is the estimation. This is based *a posteriori* on the new knowledge gained from the acceleration data in quaternion $z$ and quaternion $x^-$, as well as the error matrix $P^-$ from the prediction. Equation (20) calculates the Kalman gain matrix ($K$). The quaternion $x$ and matrix $P$ are determined in equations (21) and (22), respectively. Finally, the calculated quaternion $x$ is normalized and transformed back into Euler angles.

$$K_i = P_i^- H^T (H P_i^- H^T + R)^{-1} \tag{20}$$

$$x_i = x_i^- + K_i(z_i - H x_i^-) \tag{21}$$

$$P_i = P_i^- - K_i H P_i^- \tag{22}$$

# 3 Results

The results, which are presented in this article, represent only a small selection of the experiments conducted. The aim is to provide a representative overview of the performance of the filters in typical tasks of an industrial drone. The flight tests were carried out using a DJI-M300 drone. It is important to note that the calculation parameters of the filters were optimized for the RMSE of the respective reference tilt angle in every experiment. As described in section 2.2, all algorithms are implemented as three-dimensional filters. However, since the calculation of the yaw angle has no influence on the tilt angle of a drone, these results are not discussed. In all plots comparing the filters, blue represents the complementary filter, red represents the Madgwick filter, green represents the Kalman filter, and black represents the reference.

For reasons of computational efficiency, only a representative segment of each flight is taken from the datasets. The used filters are compared with the reference angles from the drone's autopilot. The autopilot calculates the Euler angles using an extended Kalman filter, which may have some deviations compared to the actual attitude of the drone. However, in these comparisons, it is assumed that the data from the autopilot corresponds to the actual attitude angles of the drone. Additionally, the implemented filters in these experiments are compared with the currently used filter of Drone Rescue Systems. Since the latter does not include an installation correction factor for the IMU, that is rotated about the vertical axis, the explicit evaluation of the pitch and roll angles is not useful. Therefore, the data from the project partner's filter is only included in the tilt angle plots, where it is marked in purple.

## 3.1 Flight test 1: Line without hover

In the first flight attempt by the project partner DRS, the drone flew along a straight line without hovering before reversing. Figure 2 shows the pitch angle and figure 3 shows the roll angle of the drone in flight test 1. The drone pitches down to fly forward from the initial position, then pitches up strongly without hovering to initiate the return flight. Since the return flight is longer, the drone needs to fly forward again to return to the starting point and stops with a brief pitch up. The same applies to the roll movements in the second part of the flight. In between, a 90-degree turn around the vertical axis is performed to stay on the same straight line, however this is not visible in these plots. A qualitative comparison of the algorithms based on the plots is only useful to a limited extent since the filter calculations are very close to each other and deviate approximately evenly from the reference provided by the drone's autopilot.

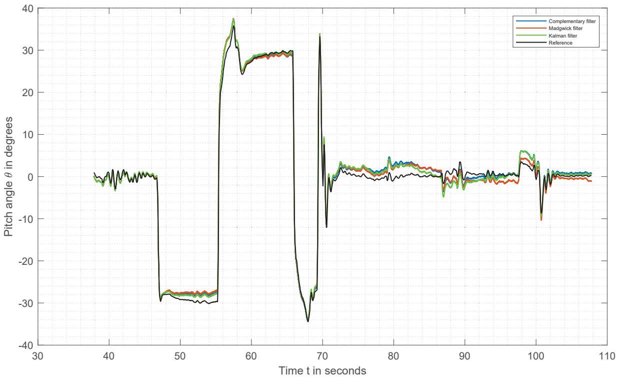


Figure 2: Pitch angle $\theta$ over time in flight test 1

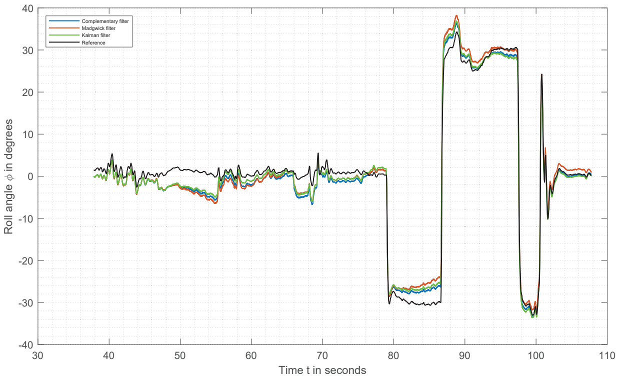


Figure 3: Roll angle $\phi$ over time in flight test 1

Only a few conclusions can be drawn about differences in the performance of the implemented algorithms in the depiction of the tilt angle, shown in figure 4. Generally, the results of the implemented filters are nearly identical. However, it is noticeable that the algorithms exhibit greater deviations from the reference in the second flight phase using a rolling motion. Additionally, the calculated tilt angle of the Drone Rescue Systems filter is presented in this plot. It deviates significantly more from the reference than the implemented filters. During fast changes in the tilt angle at the beginnings and ends of forward and backward flights, the DRS filter's calculations approach the reference. However, in phases with constant angles of inclination, the accuracy of the calculations decreases significantly.

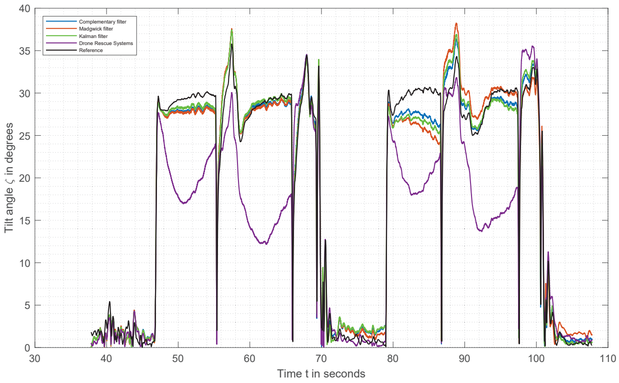


Figure 4: Tilt angle $\zeta$ over time in flight test 1

These findings are also confirmed by the calculation of the root mean square errors (RMSEs), which are listed together with the optimized calculation parameters of this flight experiment in table 1. The complementary filter calculates the tilt angle $\zeta$ with the lowest deviation. However, the Kalman filter and Madgwick filter are only slightly behind on accuracy. The supposition that the pitch angle $\theta$ is calculated more accurately than the roll angle $\phi$ is confirmed, with the algorithms yielding very similar RMSEs for the individual angles. Upon examining the optimized calculation parameters, it is noticeable that the acceleration data is only minimally incorporated into the calculations for the complementary filter and Madgwick filter. As expected from the qualitative analysis, the calculated RMSE of the tilt angle for the DRS filter is significantly higher than the values of the other filters.

Table 1: Quantitative comparison of the filters based on the RMSE of the angles in degrees with optimised calculation parameters in flight test 1

| **Flight test 1** | RMSE $\zeta$ | RMSE $\phi$ | RMSE $\theta$ | Calculation parameters |
|---|---|---|---|---|
| **Complementary** | 1.64 | 2.50 | 1.63 | 0.9999466 |
| **Magwick** | 2.12 | 2.85 | 1.58 | 0.0068125 |
| **Kalman** | 1.87 | 2.45 | 1.61 | [8.4643e+07 1.1877e+16] |
| **DRS** | 7.79 | / | / | / |

### 3.2 Flight test 2: Endurance flight

In the second flight experiment, the filters are compared over a longer time period, with a series of more complex maneuvers, specifically the continuous flight of a slow eight. In this maneuver, unavoidable errors in the calculation of the attitude angles caused by additional accelerations, occur due to the constant change of direction of the drone. Due to the very large dataset, the following plots are difficult to interpret and are therefore only discussed superficially.

Figure 5 shows the pitch angle over time. It can be seen, that the complementary filter reaches its limit. It oscillates with a high frequency with substantial deviations from the reference. This indicates that the weighting of the acceleration data is significantly higher than in the previous flight experiment. This can be explained by the longer duration of the experiment and the resulting increasing drift due to the integration of measurement inaccuracies. As a result, the course of the reference cannot be adequately calculated. The Kalman filter also shows large deviations, but it can display the rough trajectory of the reference. The Madgwick filter can reproduce the reference most accurately.

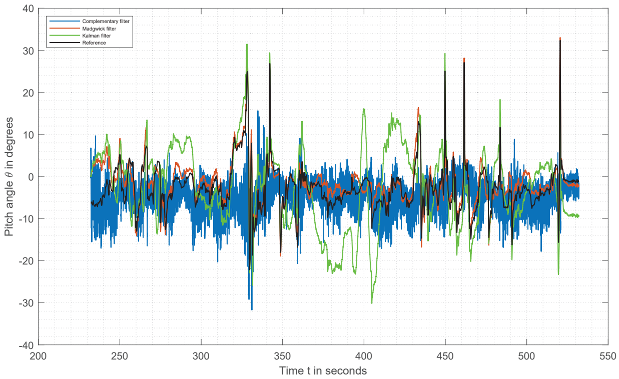


Figure 5: Pitch angle $\theta$ over time in flight test 2

The depiction of the roll angle in figure 6 continues this trend. The results of the complementary filter are unusable due to high-frequency oscillations without a true representation of the reference. The Kalman filter exhibits large deviations, but its results approach the reference in wide parts of the flight test. The Madgwick filter can represent the reference with smaller errors.

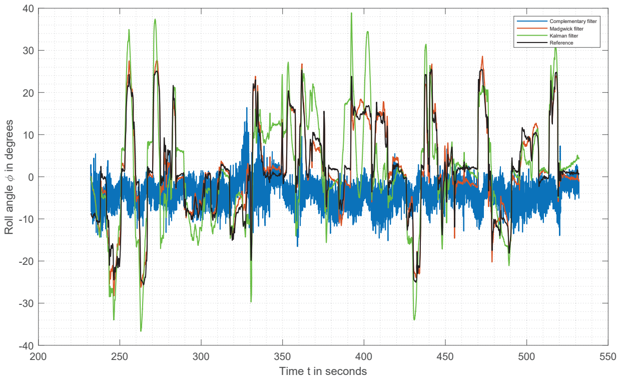


Figure 6: Roll angle $\phi$ over time in flight test 2

In figure 7, these effects are further amplified in the depiction of the tilt angle. Due to the unusable results from the complementary filter, its calculations are not displayed in this plot for purposes of clarity. The Kalman filter produces significant deviations and can only partially track the reference trajectory. However, the comparison between the Madgwick filter and the algorithm currently used by the project partner seems interesting. In this flight experiment, these two filters provide similar results with only minor deviations from the reference.

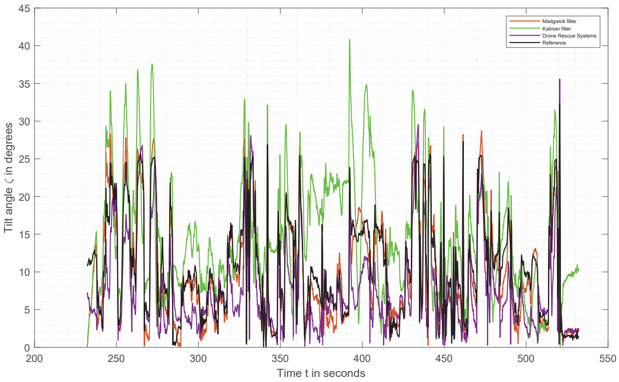


Figure 7: Tilt angle $\zeta$ over time in flight test 2

To more precisely analyze the results of the tilt angle, the absolute deviation of the algorithms from the reference is presented in figure 8. Due to the reasons mentioned earlier, the results of the complementary filter are neglected. The results of the Kalman filter exhibit continuously high deviations. The attitude angle is overestimated most of the time, as evidenced by the positive deviation. In general, the DRS algorithm deviates more strongly than the Madgwick filter, but the results are very similar when calculating the maximum errors. The largest absolute deviation above the reference of the DRS algorithm is 11.24 degrees, while that of the Madgwick filter is 8.84 degrees. The largest calculated deviation below the reference for the DRS algorithm is $-10.94$ degrees, which is almost identical to the Madgwick filter at $-10.90$ degrees. In terms of the risk of erroneous triggering or non-triggering of the parachute, the two filters behave very similarly, with slightly more accurate results from the Madgwick filter.

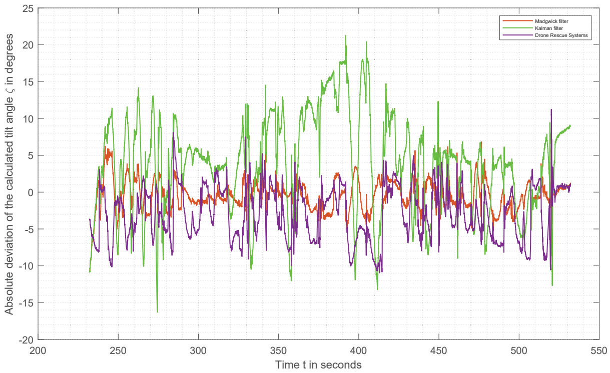


Figure 8: Absolute deviation of the calculated tilt angle $\zeta$ over time in flight test 2

These findings are highlighted in table 2. The RMSEs of the complementary filter and the Kalman filter for the tilt angle are very similar, but the results of the complementary filter are negligible because its calculations oscillate with large deviations around the reference at a high frequency. This is caused by the high weighting of the heavily noisy data from the acceleration sensors, as can be detected when looking at the optimised calculation parameter. In comparison to the other implemented filters, the Madgwick filter can calculate all considered angles with the lowest RMSEs. As expected from the qualitative analysis, the Madgwick filter also provides a significantly lower RMSE for the tilt angle $\zeta$ compared to the DRS algorithm.

Table 2: Quantitative comparison of the filters based on the RMSE of the angles in degrees with optimised calculation parameters in flight test 2

| **Flight test 2** | RMSE $\zeta$ | RMSE $\phi$ | RMSE $\theta$ | Calculation parameters |
|---|---|---|---|---|
| **Complementary** | 7.90 | 12.76 | 6.45 | 0.9222656 |
| **Magwick** | 2.06 | 2.28 | 2.60 | 0.0116875 |
| **Kalman** | 7.54 | 7.15 | 8.10 | [1.3026e+08,6.7378e+15] |
| **DRS** | 4.59 | / | / | / |

# 4 Discussion

This article shows that there is definite potential for improvement in the currently used algorithm of Drone Rescue Systems. In the long-term flight test in section 3.2, the Madgwick filter performs significantly better, and thus, based on the conducted experiments, is favored as an alternative to the currently used algorithm. The greatest unknown factor is that the filter's calculation parameters are optimized individually for the input data of each test. To draw a meaningful conclusion on which filter is best suited for drone applications and whether any of the algorithms represent an improvement over the DRS algorithm, further long-term experiments with various complex maneuvers would need to be conducted, with the calculation parameters remaining unchanged between trials.

Another possibility to improve the results would be to use a similar filter as in [Mil18], which calculates the angles based on two IMUs. The DRS parachute system already has two installed IMUs, making this feasible without any significant changes to the parachute system. However, it is unclear whether the processing of data from a second IMU located very close to the first IMU in the parachute system would provide as significant advantages as in [Mil18]. Additionally, a significant improvement could lie in integrating the already installed magnetometers, although this would require sensor calibration at system startup. In further experiments, the potential benefits would need to be weighed against the disadvantages.

Furthermore, it should be noted that the reference is represented by the autopilot's calculations. Even if a seemingly more powerful extended Kalman filter is used, it can still calculate erroneous attitude angles. Hence, it would be advantageous to compare the filters in further experiments with a test setup that can determine the drone's actual attitude.

# 5 Summary

To calculate the Euler angles of a drone, data from various sensors must be processed by an attitude reference algorithm. Due to the requirement to keep the parachute rescue systems of the project partner Drone Rescue Systems as user-friendly and versatile as possible in their operational environment, only acceleration data from acceleration sensors and angular rates from gyroscopes are taken into account for the calculations. Specifically, in this article, the complementary filter, the Madgwick filter, and the Kalman filter are qualitatively and quantitatively compared. Using data from various flight tests, it is also examined whether any of the tested filters represents an improvement over the currently used algorithm.

The filters are implemented in MATLAB® and compared in various scenarios. In the results of flight test 1, it is apparent that, with the proper choice of calculation parameters, the filters can represent a significant improvement over the currently used algorithm. In the second flight test, which considers a longer time span with more complex flight maneuvers, the complementary filter reaches its limits. Even the Kalman filter can only partially represent the reference with large deviations. The Madgwick filter works most accurately in this experiment, representing a significant improvement over the currently used algorithm of the project partner. In order to be able to make a definitive statement about which filter is best suited for use in the attitude reference system of the parachute system, additional long-term experiments with a series of complex maneuvers must be carried out, and the calculation parameters of the filters must not be changed between tests.

# References

[BAL13] Rudolf Brockhaus, Wolfgang Alles, and Robert Luckner. *Flugregelung.* Springer, 3 edition, 2013.

[BM12] Randal W Beard and Timothy W McLain. *Small unmanned aircraft: Theory and practice.* Princeton university press, 2012.

[Kui99] Jack B Kuipers. *Quaternions and rotation sequences: a primer with applications to orbits, aerospace, and virtual reality.* Princeton university press, 1999.

[LW13] Wei Li and Jinling Wang. Effective adaptive Kalman filter for MEMS-IMU/magnetometers integrated attitude and heading reference systems. *The Journal of Navigation*, 66(1):99–113, 2013.

[Mad10] Sebastian O. H. Madgwick. An efficient orientation filter for inertial and inertial/magnetic sensor arrays. *Report x-io and University of Bristol (UK)*, 25:113–118, 2010.

[Mil18] Joshua Bruce Milam. Design and Evaluation of Novel Attitude Estimation System Using MEMS Sensors for Indoor UAS. Master's thesis, West Virginia University, 2018.

[PEH19] Alwin Poulose, Odongo Steven Eyobu, and Dong Seog Han. An indoor position-estimation algorithm using smartphone IMU sensor data. *IEEE Access*, 7:11165–11177, 2019.

[Ram23] Mario Rammelmüller. Implementierung und Vergleich verschiedener Lagereferenzalgorithmen für Drohnenanwendungen in MATLAB®. Bachelorarbeit, FH JOANNEUM, 2023.

[ZM09] Paul Zarchan and Howard Musoff. *Fundamentals of Kalman Filtering: A Practical Approach*, volume 232 of *Progress in Astronautics and Aeronautics.* American Institute of Aeronautics and Astronautics, Virginia, 3 edition, 2009.

# Numerical investigation of the effects of a winglet configuration on the lift and drag coefficients of the SD7062 wing profile

Julian Gräf, BSc[1]

[1]FH JOANNEUM - University of Applied Sciences, Institute Aviation, Alte Poststraße 149, 8020 Graz, Austria

February 15th, 2023

## Abstract

Due to the fact that the overall efficiency of an aircraft can be improved by means of aerodynamics, it is of the utmost importance to determine not only the lift but also the drag coefficient of an airfoil, when it comes to the design of an aircraft wing. For sure, the effort is to maximize the lift on the one hand, and to minimize the drag on the other hand. In aerodynamics, a basic distinction is made between the zero-lift drag and the lift-induced drag. The zero-lift drag consists of the skin friction and the form drag, which exist due to the geometry of the aircraft. The opposite is the lift-induced drag, which develops during flight because of the lift generation. Basically, lift is created by the pressure difference between the upper side and the lower side of the wing. Due to this pressure difference, the pressure equalizes at a certain point, which causes induced drag by twisting the airflow and producing vortices along the wing trailing edge. In order to minimize the induced drag and to reduce the fuel consumption consequently, early developments since the Wright brothers lead to modern winglet designs. Winglets are vertical or angled aerodynamic components, which are mounted on the wing tips. The purpose of this paper is to illustrate the effects of a winglet configuration compared to a configuration without winglets by means of numerical flow simulations. For that, the lift and the drag coefficients of the SD7062 wing profile are calculated for a wide variety of angle of attacks and compared with experimental data to validate the numerical results. Finally, statements about the effects of winglets on the aerodynamic coefficients can be made and the chosen method for investigating the winglet configuration can be discussed critically.

## 1 Introduction

According to [1], the induced drag accounts for approximately 30 % of the total drag in conventional aircraft. The Prandtl lifting line theory explains that the generation of induced drag is mainly related to the tip vortices caused by three-dimensional flow at the wing tip and not necessarily to all vortices along the trailing edge, as represented in Figure 1. [2]

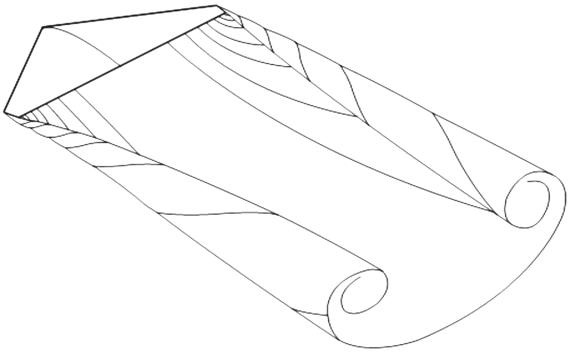


Figure 1: Schematic illustration of vortices along the trailing edge [3]

According to [4] and [5], next to curved trailing edges and fins mounted on the wing tips, Whitcomb first systematically studied the design theory of winglets. Experiments showed that winglets are a very effective way to reduce tip vortices around the wing tip region. For that purpose, different winglet designs were developed over the years, as the blended winglet configuration shown in Figure 2.

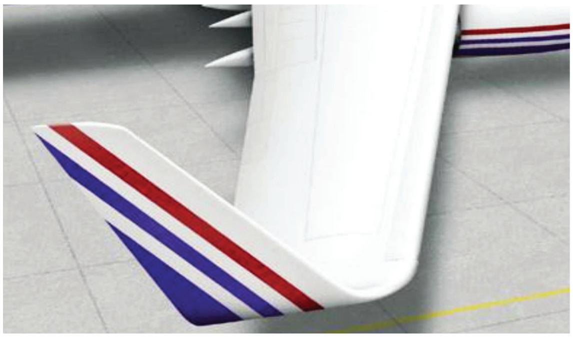


Figure 2: Blended winglet configuration [6]

In addition to blended winglets, a large variety of other wing tip devices for reducing the induced drag are used today, as the raked wing configuration or the endplate winglet for example. [7] Nevertheless, the investigations described in this paper regarding the effect of winglets on the aerodynamic coefficients were carried out with the low Reynolds number and the high lift-to-drag ratio profile SD7062. This airfoil combined with blended winglets forms the wing of a real model airplane with a total wingspan of 2.3 m, illustrated in Figure 3. [8]

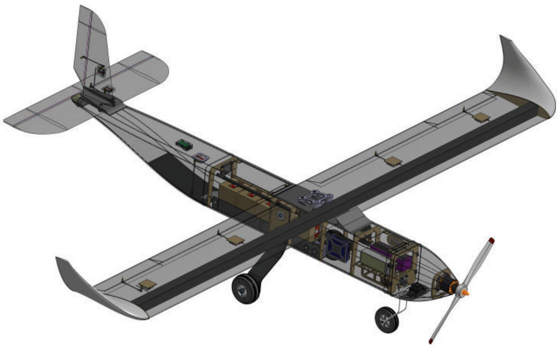


Figure 3: Isometric view of the model airplane with blended winglets [8]

## 2 Methods

In this section, the analysed wing including its design parameters is specified. Besides this, the mesh refinement study is presented, whose purpose is to optimize the mesh in order to find the final mesh for the ensuing numerical flow simulations. Moreover, the settings of the CFD simulations are shown with reference to the boundary conditions and the used output expressions. Finally, the approach to determine the lift and the drag coefficients is described, which is based on mathematical and aerodynamic equations.

### 2.1 Geometry preparation

The wing of the model airplane is adapted from the Selig/Donovan SD7062 airfoil, which is often used for low flight speeds. Based on a two-dimensional point cloud downloaded from the airfoil plotter „airfoiltools“ and visualized in Figure 4, the three-dimensional wing geometry including the winglets were designed in CATIA V5-6R2012 [9] by extruding the profile lines.

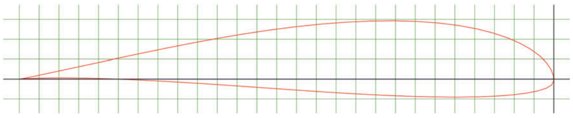


Figure 4: Point cloud of the SD7062 airfoil [10]

Furthermore, the wing area $S = 0.54$ m$^2$ and the aspect ratio AR $= 7.41$ of the configuration without winglets were determined with the chord length $c = 0.27$ m and the wingspan $b = 2$ m by the Equations 1 and 2. [11]

$$S = b \cdot c \tag{1}$$

$$\mathrm{AR} = \frac{b^2}{S} \tag{2}$$

The values for the taper ratio $\lambda = 1$ and the sweep angle $\Lambda = 0$ are noticeable, as the wing is rectangular. Nevertheless, the calculations of the wing area and the aspect ratio for the wing with winglets must be performed equivalent. This leads to a wing area of $S_\mathrm{W} = 0.62$ m$^2$ and an aspect ratio of $\mathrm{AR}_\mathrm{W} = 8.52$. [11]

## 2.2 Mesh refinement study

The mesh refinement study is a suitable tool for weighing the meshing requirements. It is a fact that the simulation time increases with an increase in elements, but also the accuracy of the results increases in that case. Therefore, two different mesh types were analysed to find a so-called final mesh for the CFD simulations.

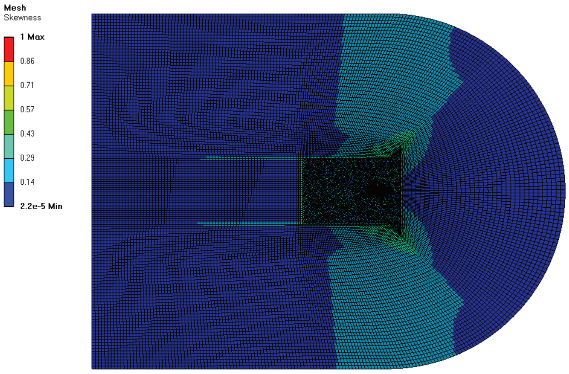


Figure 5: Final computational mesh with a semicircular inlet and 5.8 million elements [12]

Firstly, a rectangular mesh with approximately three million elements was created. Secondly, a C-grid structured mesh with a semicircular inlet to the fluid domain with 5.8 million elements was generated. Figure 5 displays the skewness of the final mesh. Next to a low aspect ratio, the mesh skewness is an important parameter to argue about the mesh quality. The skewness determines how close to ideal a cell is, by measuring the angles of the actual cells and comparing it to the angles of the ideal elements.

$$\max\left[\frac{\theta_{\text{max}}-\theta_{\text{e}}}{180-\theta_{\text{e}}}, \frac{\theta_{\text{e}}-\theta_{\text{min}}}{\theta_{\text{e}}}\right] \tag{3}$$

According to [13], the determination of the skewness for triangular and quadrilateral elements is based on the normalized angle deviation method. There, skewness is defined as the maximum of the ratio of angular deviations from the ideal elements by using the largest angle in the cell $\theta_{\text{max}}$, the smallest angle in the cell $\theta_{\text{min}}$ and the angle for an equi-angular cell $\theta_{\text{e}}$. For an equilateral triangle $\theta_{\text{e}} = 60°$ and for a square $\theta_{\text{e}} = 90°$ apply.

Figure 5 indicates that the generated mesh has high quality, as 57 % of the cells have an excellent skewness value below 0.25 and 99 % have a skewness below 0.5. Next to the good skewness values and the fine resolution of the flow area, the inflation layers shown in Figure 6 help to dissolve the boundary layer close to the wing. For that reason, both wing configurations were meshed in the same way, including the equivalent number of meshing elements and inflation sizings.

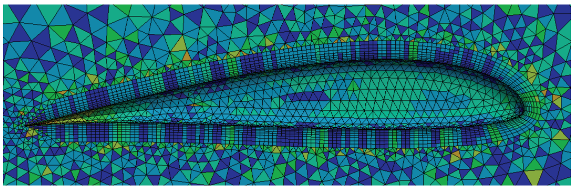


Figure 6: Representation of the inflations layers around the SD7062 airfoil [12]

## 2.3 Numerical simulation settings

The CFD simulations were setup identical for both configurations in ANSYS CFX-Pre [14]. The implemented fluid is air at a temperature of 25° C. As it was treated as a continuous flow, a stationary isothermal flow was assumed. Based on [15], the Shear-Stress-Transport model was used as a turbulence model with the Gamma-Theta model for transitional turbulence. Furthermore, monitoring points were created to follow the convergence within 500 iterations.
For the boundary conditions, the flow at the inlet was defined as subsonic with a free stream velocity $v_{\infty}$, which was entered in Cartesian coordinates. There, the u-velocity was called $v_{\text{x}}$ and the v-velocity $v_{\text{y}}$. The value for the w-velocity was set to 0 m/s, as there is no flow in the z-direction. The wing surface was specified as a smooth no slip wall, which means that the velocity of the flow at the wing itself tends to be zero. The flow at the outlet was defined with a constant static pressure. The outer wall of the mesh region was specified as a free slip wall, as the flow field is infinitely large and the velocity should not be zero at these boundaries.

Figure 7 represents the applied boundary conditions in ANSYS CFX-Pre [14]. Next to the inlet (red), the outlet (blue), the no slip wall at the wing (green) and the free slip wall (grey), a symmetry plane at the point of the wing root was created. As the wing is symmetrical and consequently also the flow, it is sufficient to calculate the flow variables only half of the wing.

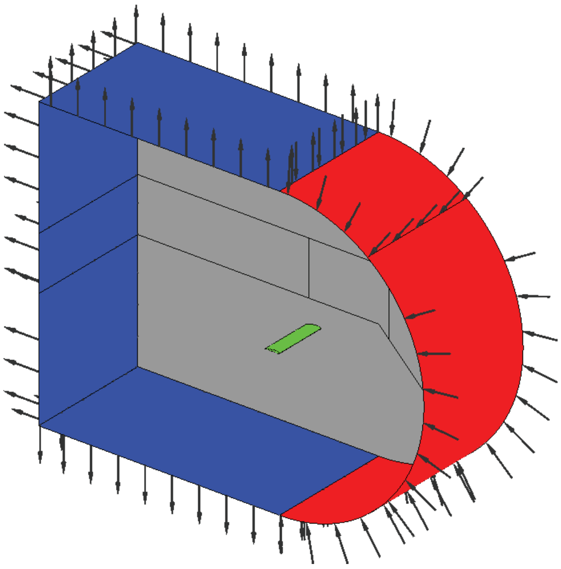


Figure 7: Presentation of the boundary conditions in ANSYS CFX-Pre [14]

Finally, three input parameters were defined, namely the outlet pressure $p_{\text{out}} = 0$ Pa, the inlet velocity $v_\infty = 19.44$ m/s and the angle of attack $\alpha = 0°$. Out of the velocity and the angle of attack, the u- and v-velocity components of the inlet flow are calculated automatically.

$$v_{\text{x}} = v_\infty \cdot \cos(\alpha) \tag{4}$$

$$v_{\text{y}} = v_\infty \cdot \sin(\alpha) \tag{5}$$

By implementing these relationships, it is possible to use the same mesh for each simulation with different angles of attack, as there is no rotation of the wing geometry, but rather a flow rotation by the value of the entered angle of attack. Therefore, 21 design points for both configurations with a constant inlet velocity and a constant outlet pressure, but with an angle of attack range of $-5° \leq \alpha \leq +15°$ were investigated.

## 2.4 Results processing

Besides the input parameters, various output parameters were created to summarize the results in an automatically generated table. There, the y-plus values as well as the normal force and the tangential force acting on the wing were exported. Due to the fact that the simulations are isothermal, the fluid density remains constant with $\rho = 1.185$ kg/m$^3$. Furthermore, also the inlet velocity $v_\infty = 19.44$ m/s, the chord length $c = 0.27$ m and the dynamic viscosity $\mu = 1.83 \cdot 10^{-5}$ kg/ms are constant for all simulations. Out of these values, the Reynolds number Re $\approx 340000$ can be determined. [16]

$$\text{Re} = \frac{\rho \cdot v_\infty \cdot c}{\mu} \tag{6}$$

In contrast, it has to be mention that the output variables are not constant at all. At the top, the normal forces $F_{\text{N}}$ and the tangential forces $F_{\text{T}}$ differ, as for each simulation another angle of attack in the range of $-5° \leq \alpha \leq +15°$ was used. Furthermore, also the y-plus values at the airfoil wall vary for each simulation, although they were all in a range of about $y^+ \approx 20$. The y-plus value is a dimensionless wall distance calculated by the absolute distance from the wall $y$, the friction velocity $u_{\text{T}}$ and the kinematic viscosity $\nu$.

$$y^+ = \frac{y \cdot u_{\text{T}}}{\nu} \tag{7}$$

According to [17], this parameter is prominent when judging the applicability of wall functions, which help to predict wall bounded turbulent flows. In the viscous wall region with $y^+ < 50$, there is a direct effect of the viscosity on the shear stress. Conversely, in the outer layer with $y^+ > 50$, the effect of viscosity is negligible.

In agreement with [18], the longitudinal movement of an aircraft is described by three variables: the lift force $F_{\text{L}}$, the drag force $F_{\text{D}}$ and the moment $M$.

$$F_{\text{L}} = \frac{c_{\text{L}} \cdot \rho \cdot v_\infty^2 \cdot S}{2} \tag{8}$$

$$F_{\text{D}} = \frac{c_{\text{D}} \cdot \rho \cdot v_\infty^2 \cdot S}{2} \tag{9}$$

$$M = \frac{c_{\text{M}} \cdot \rho \cdot v_\infty^2 \cdot c \cdot S}{2} \tag{10}$$

It is a fact that the lift and the drag force are defined as the perpendicular vector (lift) and the parallel vector (drag) in dependence of the free-stream velocity $v_\infty$. These forces act on the aerodynamic center ac. For that reason, the forces calculated with ANSYS [19] do not represent the lift and the drag forces of the wing, as the numerical software calculates the normal and tangential forces. Hence, it is obvious to export the normal and the tangential forces, and import it to MATLAB® [20], where the determination of the lift and drag force can be implemented.
Figure 8 displays the forces acting on an aircraft wing with a given angle of attack.

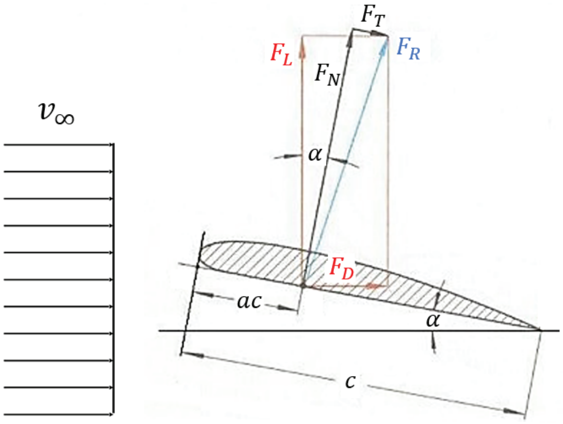


Figure 8: Visualisation of the lift and drag force calculation with a given angle of attack [21]

The resulting aerodynamic forces consist of the corresponding vector components of the normal force $F_\mathrm{N}$ and the tangential force $F_\mathrm{T}$ acting on the wing. Thus, the lift force $F_\mathrm{L}$ and the drag force $F_\mathrm{D}$ follow. [21]

$$F_\mathrm{L} = F_\mathrm{N} \cdot \cos(\alpha) - F_\mathrm{T} \cdot \sin(\alpha) \tag{11}$$

$$F_\mathrm{D} = F_\mathrm{N} \cdot \sin(\alpha) + F_\mathrm{T} \cdot \cos(\alpha) \tag{12}$$

The calculation of the corresponding aerodynamic coefficients is done using Equation 8 for the lift coefficient $c_\mathrm{L}$ and Equation 9 for the drag coefficient $c_\mathrm{D}$. [21]

$$c_\mathrm{L} = \frac{2 \cdot F_\mathrm{L}}{\rho \cdot v_\infty^2 \cdot S} \qquad c_\mathrm{D} = \frac{2 \cdot F_\mathrm{D}}{\rho \cdot v_\infty^2 \cdot S} \tag{13}$$

Furthermore, the zero-lift drag $c_\mathrm{D0}$ and the lift-induced drag $c_\mathrm{Di}$ can be calculated, as their sum is equivalent to the total drag coefficient $c_\mathrm{D}$.

$$c_\mathrm{D} = c_\mathrm{D0} + c_\mathrm{Di} \tag{14}$$

As the lift-induced drag $c_\mathrm{Di}$ is proportional to the square of the lift coefficient $c_\mathrm{L}$

$$c_\mathrm{Di} = \frac{c_\mathrm{L}^2}{\pi \cdot e \cdot \mathrm{AR}} \tag{15}$$

the zero-lift drag $c_\mathrm{D0}$ follows out of the total drag coefficient $c_\mathrm{D}$ according to Equation 14 with a known Ostwald efficiency factor $e$. [22]

$$e = \left[0.47 + \left(\frac{1}{\sqrt{\mathrm{AR}}}\right)\right] \cdot \cos(\Lambda)^{0.1} \tag{16}$$

In order to process the results coming from the CFD simulations, a routine was implemented in MATLAB® to apply all mentioned equations. Firstly, all 21 normal forces as well as all 21 tangential forces were exported from ANSYS CFD Post [23] and imported into MATLAB®. Furthermore, all constant values were added to the routine. On this basis, the wing areas, the aspect ratios, the Reynolds number and the Ostwald efficiency factors were determined. After that, the routine starts with a data separation and with an interpolation of the normal and tangential forces. For that purpose, the angle of attack range was reduced from 1°-steps to 0.01°-steps, in order to get a better distribution and a finer resolution of the plotted diagrams. Thereafter, the calculations of the lift and drag forces starts by following the Equations 11 and 12, out of which the lift-to-drag ratio can be calculated accordingly.

$$\frac{F_\mathrm{L}}{F_\mathrm{D}} = \frac{c_\mathrm{L}}{c_\mathrm{D}} \tag{17}$$

Moreover, the aerodynamic coefficients are determined based on Equation 13, which can be used for the advanced mathematical calculations. So, the lift-induced drag coefficients and the zero-lift drag coefficients follow from the Equations 14, 15 and 16. Last but not least, the aerodynamic coefficients predicted by X-foil [24] were implemented and compared to the numerically calculated values. After that, a lot of data is plotted to visualise the results.

# 3 Results

This section intends to present the results of the CFD simulations as well as the plots which were generated within the MATLAB® routine.

## 3.1 CFD simulations

In order to ensure comparability between the two configurations, all figures are related to an angle of attack of $\alpha = 2°$. Both Figures 9 and 10 show the velocity distribution around the SD7062 airfoil. Here, the cutting planes were set to 50 % of the wingspan and it is obvious that the velocities are almost equal due to the same geometries, which only differ in the winglet. From an aerodynamics point of view, the higher velocity at the upper side of the wing is generating lift.

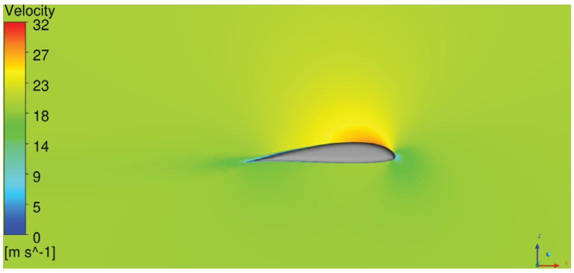


Figure 9: Velocity distribution of the wing without winglets for $\alpha = 2°$ [23]

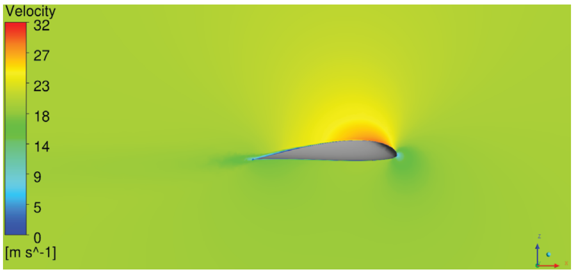


Figure 10: Velocity distribution of the wing with winglets for $\alpha = 2°$ [23]

Figure 11 represents the velocity streamlines of the wing with winglets in black and the pressure contour of the wing with an attached scale of magnitude. The twisted airflow along the wing tip, which causes vortices and consequently lift-induced drag, is visualised.

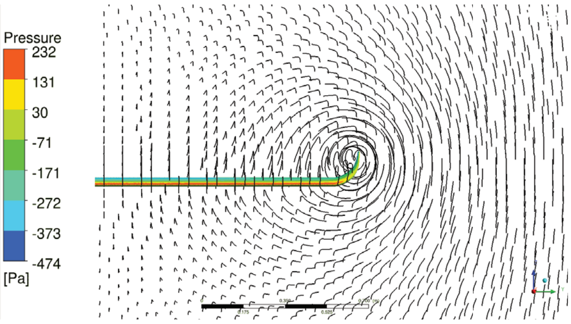


Figure 11: Velocity streamlines of the wing with winglets for $\alpha = 2°$ [23]

The pressure distribution is presented separately in Figure 12. However, it is noticeable that the pressure at the lower side of the wing is greater than at the upper side. This pressure difference is responsible for generating lift as well.

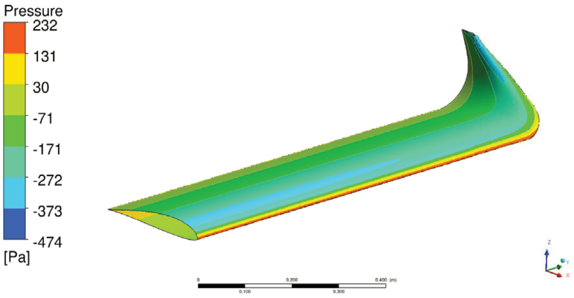


Figure 12: Pressure distribution of the wing with winglets for $\alpha = 2°$ [23]

Finally, this lift force gets visible when looking at the force distribution in Figure 13. There are several force vectors which act on the upper side of the wing. As these vectors point upwards, it is obvious that they visualize the lift force.

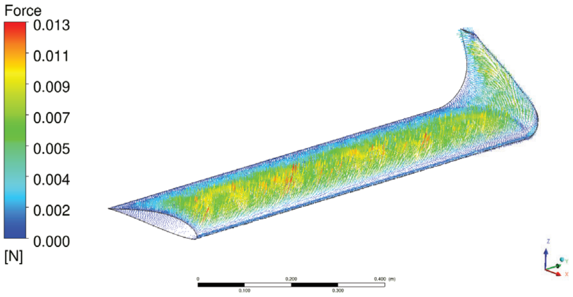


Figure 13: Force distribution of the wing with winglets for $\alpha = 2°$ [23]

## 3.2 Matlab® plots

Here, the MATLAB® plots which were generated within the routine are presented. Special focus is put on the comparison of the aerodynamic coefficients of the two different wing configurations. The values for the winglet configuration are displayed in red and the values for the no-winglet configuration are represented in black.

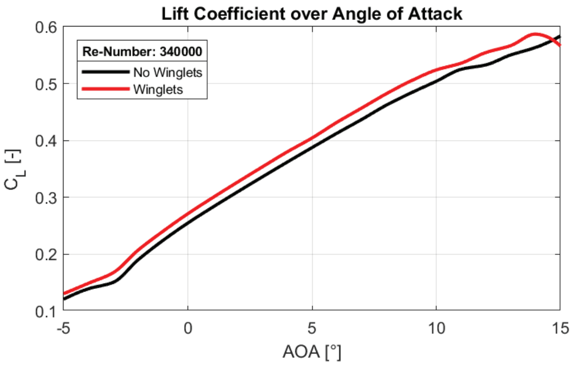


Figure 14: Lift coefficient over angle of attack for both configurations [20]

Figure 14 and 15 visualize the calculated lift and drag coefficients of the three-dimensional SD7062 wing profile for the Reynolds number Re $\approx$ 340000. Here, the aerodynamic coefficients are plotted over the angle of attack range from $-5° \leq \alpha \leq +15°$. It is noticeable that the winglet configuration shows better results in general, because of the greater lift coefficients. Surprisingly, both configurations have nearly the same drag coefficients.

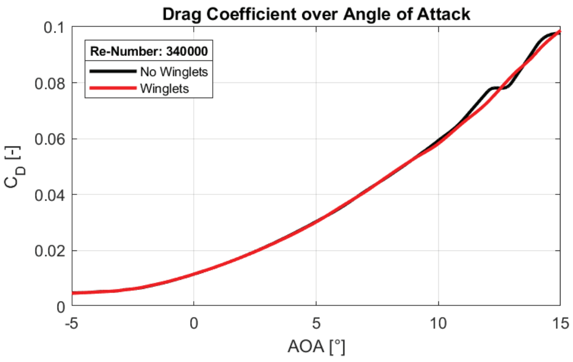


Figure 15: Drag coefficient over angle of attack for both configurations [20]

The famous drag curve is plotted in Figure 16, which shows the lift coefficients as a function of the drag coefficients. It is perceptible that the winglet configuration has slightly higher values for the lift coefficients at given drag coefficients. This result shows the advantage of winglets, especially for increasing angles of attack.

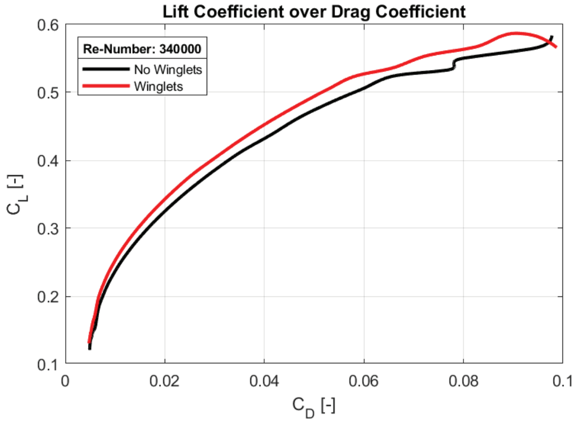


Figure 16: Lift coefficients over drag coefficients (drag curves) of both configurations [20]

Lastly, the results of the lift-induced drag calculations for the winglets configuration are presented in Figure 17. The total drag with its two parts is visualised in black. Firstly, there is the zero-lift drag in red and secondly the lift-induced drag in blue. It is observable that the skin friction and the form drag account for the main part (79 %), whereas the induced drag due to vortices represent a small part of the drag (21 %).

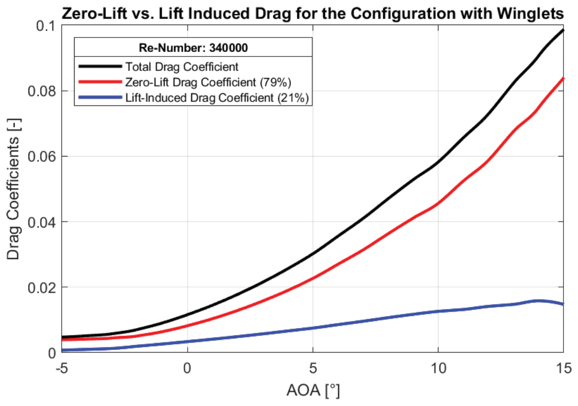


Figure 17: Components of the total drag for the configuration with winglets [20]

# 4 Discussion

As a discussion, the results of the configuration with winglets gained with 3D CFD simulations are compared with 2D data of X-Foil. On „airfoiltools.com“ plots of the lift and drag coefficients for the SD7062 airfoil are available. [10]

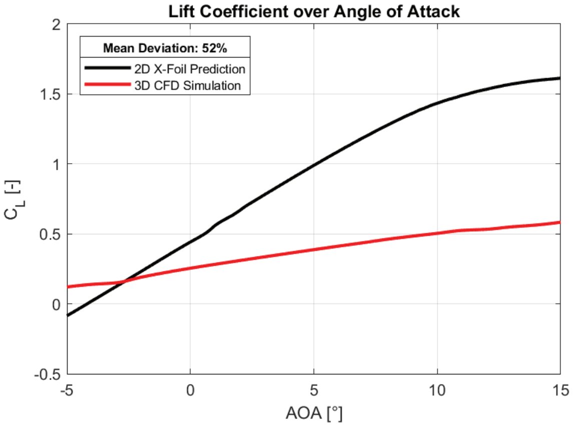


Figure 18: Validation of the lift coefficients gathered by CFD simulations [20]

The Figures 18 and 19 show the X-Foil values in black and the CFD coefficients in red. The curves do not match in a good way, neither for the lift (52 % deviation) nor for the drag coefficients (122 % deviation). It seems that the simulations underestimate the lift and overestimate the drag, but also 3D effects play a decisive role, when investigating aerodynamic coefficients.

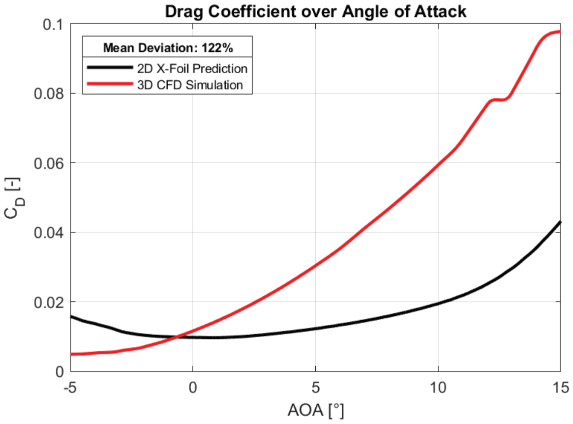


Figure 19: Validation of the drag coefficients gathered by CFD simulations [20]

# 5 Conclusion

To sum up, the lift-induced drag due to vortices is based on the lift coefficient of the wing, the aspect ratio and the sweep angle. The induced drag decreases with an increase in the aspect ratio, which means that a wing should be narrow with a large wingspan. For such a wing, the lift decreases from the root to the tip and it will generate an elliptic force distribution and small-intensity vortices, as presented in Figure 20.

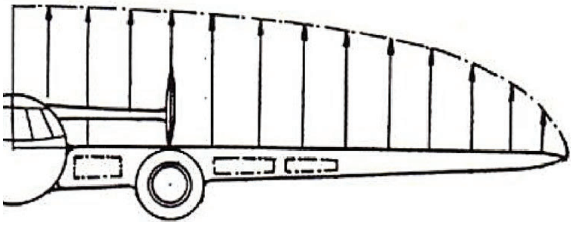


Figure 20: Elliptic lift distribution on a wing with finite elongation [7]

As a conclusion, it can be stated that it is not that easy to generate an aerodynamic model, which fully represents the real flow when investigating it with numerical software. Nevertheless, it is obvious that the configuration of the SD7062 wing profile with winglets shows not only higher lift forces with nearly the same drag, but has also a higher lift coefficient and a better lift-to-drag ratio in general. As the phenomenon of stall occurs for angles of attack of more than 15°, these results were not published here. In order to overcome the problem with the stall-simulations, it would be better to carry out various transient flow simulations instead of steady-state simulations. An improvement would be to calculate the critical angle of attack, which is possible with ANSYS software. Furthermore, also the accuracy of the results could be improved by drastically reducing the y-plus values, in order to reach values for the lift and the drag coefficients which are comparable to those from X-Foil. Although the calculated aerodynamic coefficients differ from the experimental values, it is a fact that winglets can reduce the wing tip vortices and the resulting lift-induced drag.

## Acknowledgments

This paper and the research behind it would not have been possible without the exceptional support of FH Joanneum GmbH, which makes the very expensive software required for this project available to students.

## References

[1] Slooff, J., *Computational methods for subsonic and transonic aerodynamic design. Proceeding of the first conference on inverse design concepts and optimization in engineering*, 1983.

[2] Narayan, G. and John, B., *Effect of winglets induced tip vortex structure on the performance of subsonic wings*, Aerosp Sci Technol, 2016.

[3] McLean, D., *Wingtip Devices: What they Do and How They Do it*, Boeing Conference, 2005.

[4] Zhang, L., *Optimization and analysis of winglet configuration for solar aircraft*, Chinese Journal of Aeronautics, 2019.

[5] Whitcomb, R., *A design approach and selected wind tunnel results at high subsonic speeds for wing-tip mounted winglets*, NASA Langley Research Center, 1976.

[6] Gratzer, L., *Blended winglet*, Google, 1994.

[7] Recalenda, E., *Analysis of three different Winglet models with comparison between CFD simulation and wind tunnel testing*, Politecnico di Torino, 2020.

[8] Aeronautics, J., *Aircraft Design Report*, Joanneum Aeronautics, 2022.

[9] CATIA, *Computer Aided Three-Dimensional Interactive Application*, Dassault Systems, 2019.

[10] Airfoiltools, "Airfoil Plotter," Website: `http://airfoiltools.com///`, last access: 29.01.2023.

[11] Radespiel, R., *Tragflügelaerodynamik*, Technische Universität Braunschweig, 2007.

[12] ANSYS, *Meshing Users Guide*, 2018.

[13] Engmorph, "Skewness Calculation for 2D Elements," Website: `https://www.engmorph.com/skewness-finite-elemnt`, last access: 29.01.2023.

[14] ANSYS, *CFX-Pre Users Guide*, 2018.

[15] Barros, A. and Camara, F., *Use of gamma theta model to determine the aerodynamics charactersitics of NACA 23012*, Universidade Federal do Rio Grande, 2017.

[16] Laurien, E. and Oertel, H., *Numerische Strömungsmechanik: Grundgleichungen und Modelle - Lösungsmethoden - Qualität und Genauigkeit*, Springer Verlag, 2013.

[17] Simscale, "Y-plus," Website: `https://www.simscale.com/forum/t/what-is-y-yplus`, last access: 29.01.2023.

[18] Fichter, W., *Statische Stabilität der Längsbewegung*, Universität Stuttgart, 2015.

[19] ANSYS, *CFX Introduction*, 2018.

[20] MATLAB, *MATrix LABoratorium*, MathWorks, Inc., 2021.

[21] Foysi, H., *Auftriebs- und Widerstandsmessung an einem Tragflügelprofil*, Universität Siegen, 2020.

[22] Saravanan, R., *Design of Parametric Winglets and Wing tip devices – A Conceptual Design Approach*, Linkoping University, 2012.

[23] ANSYS, *CFD Post Users Guide*, 2018.

[24] Xfoil, "Xfoil - Subsonic Airfoil Development System," Website: `https://web.mit.edu/drela/Public/web/xfoil/`, last access: 29.01.2023.

# Numerische Simulation von Tragflächenvereisungen

Luckeneder Sarah, BSc

FH JOANNEUM - University of Applied Sciences, Institut Luftfahrt/Aviation, Alte Poststraße 149, 8020 Graz, Österreich

## Zusammenfassung

In dieser Arbeit wird das Vereisungstool FENSAP-ICE für die Vorhersage der Eisschicht an Tragflächen von Flugzeugen genutzt. Zuerst wird das Modell in CATIA erstellt und die Vernetzung in ICEM CFD durchgeführt. Die Vereisungssimulation findet jeweils mit den Umweltbedingungen aus Appendix C, gefrierender Nieselregen und gefrierender Regen statt. Eine Simulation kann genutzt werden um die Änderung der aerodynamischen Kräfte, hervorgerufen durch die Eisschicht, vorherzusagen. Die Flügelprofile NACA 0012, NACA 23012 und SD 7062 werden verwendet. Es ergibt sich jeweils eine Strömungs-, eine Tröpfchen- und eine Vereisungslösung. Diese Lösung ist stark abhängig von der Feinheit des verwendeten Netzes.

## 1 Einführung

Die Arbeit Numerische Simulation von Tragflächenvereisungen basiert auf den Ergebnissen der Bachelorarbeit Analysis of Numerical Simulations of Icing on Airfoils, die nun weiterverfolgt wird. [Luc22] Durch Vereisungen an Tragflächen während eines Fluges werden die Eigenschaften stark verändert. Durch die Eisschicht kann es zu erhöhtem Widerstand und verringertem Auftrieb kommen. [Eur15] Tests und Evaluierungen, die für die Zertifizierung eines Luftfahrzeuges notwendig sind, bleiben unverzichtbar. Dennoch hat die Bedeutung von Simulationen in den letzten Jahren stark zugenommen. [Ban20]

### 1.1 CFD Simulationen

Mit Hilfe von Computational Fluid Dynamics (CFD) kann die Strömung um verschiedene Körper berechnet werden. Durch den Einsatz während der Konstruktionsphase können zum Beispiel die aerodynamischen Eigenschaften von Autos oder Flugzeugen verbessert werden. [And95] Für eine Simulation müssen die richtigen Randbedingungen und Vereinfachungen getroffen werden. Die Geometrie wird in einem computer-aided design (CAD) Programm nachkonstruiert. Anschließend wird ein Netz erzeugt und die gewünschten Lösungsansätze ausgewählt. Danach kann eine Strömungslösung berechnet werden. [Mon19]

### 1.2 Parameter

Wichtige Parameter in der Berechnung von Strömungs-, Tröpfchen- und Vereisungslösungen sind die Courant-Friedrichs-Lewy (CFL) Nummer und der dimensionslose Wandabstand y$^+$. Beide Parameter sind für die Stabilität und Genauigkeit der berechneten Lösung verantwortlich. [And95] In der Nähe von Wänden entstent eine sogenannte viskose Unterschicht. Die turbulenten Schwankungen sind hier vernachlässigbar. Die Formel für den dimensionslos gemachten Wandabstand lautet wie folgt.

$$y^+ = \frac{u_\tau}{\nu} y \tag{1}$$

Die Wanddschubspannungsgeschwindigkeit wird hierbei durch die kinematische Viskosität geteilt und mit der y-Position multipliziert. Die viskose Unterschicht gilt ungefähr bis zu einem

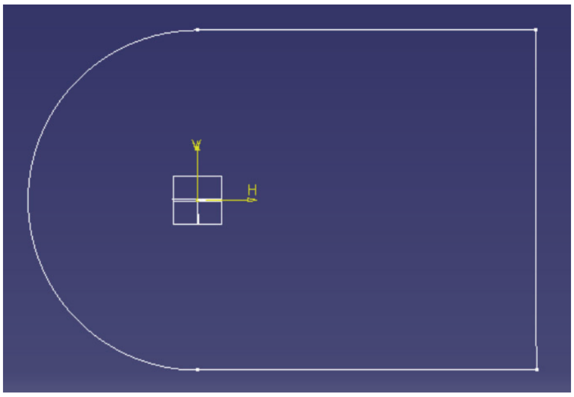


Abbildung 1: Geometrie in CATIA

Wert von $y^+ = 5$. [Kuh14] In dieser Arbeit ist es wichtig, dass der dimensionslose Wandabstand an der Oberfläche des Flügels möglichst kleine Werte annimmt, um realistische Lösungen zu erhalten. Dies wiederum bedeutet, dass die Größe der Elemente im Netz an dieser Stelle sehr klein gewählt werden muss.

# 2 Verwendete Methoden

Zuerst wurde ein CAD-Modell der Tragfläche mit Hilfe von CATIA erzeugt. Die Punkte des Profils wurden von der Webseite Airfoiltools heruntergeladen. [Air23] Eine Vereinfachung, die bei allen Profilen getroffen wurde, war die Verwendung einer kantigen statt einer spitzen Flügelhinterkante. Dadurch konnte das Netz später besser angepasst werden. Rund um den Flügel wurde eine Box erzeugt, um den Raum des Fluids einzugrenzen. Dies kann in Abbildung 1 gesehen werden. Das Profil befindet sich hierbei im Koordinatenursprung und das Verhältnis von Box zu Flügeltiefe beträgt 50:1. Das verwendete Vernetzungsprogramm war ICEM CFD. Hier konnte das zuvor erzeugte CAD-Modell importiert werden und alle nötigen Einstellungen für eine akkurate Vernetzung wurden getroffen. Die 2D-Geometrie wurde extrudiert um eine Spannweite zu erzeugen.

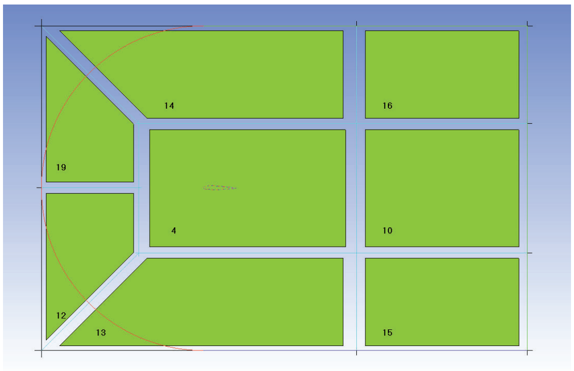


Abbildung 2: Blöcke in ICEM CFD [Luc22, S.10]

## 2.1 Netzerzeugung

In der Bachelorarbeit Numerical Simulations of Icing on Airfoils wurden Berechnungen mit unterschiedlich feinen Netzen durchgeführt. [Luc22] In dieser Arbeit wurde nun mit bereits passenden Parametern und Elementgrößen gearbeitet. In ICEM CFD wurde ein Block erzeugt, der in 8 kleinere Blöcke geteilt wurde. Diese sind in Abbildung 2 dargestellt. Anschließend wurden die Ecken und Kanten der Blöcke mit der Geometrie verbunden.

Die Abstände der Knotenpunkte der einzelnen Elemente wurden nahe der Flügeloberfläche möglichst gering gewählt. Dies ist in Abbildung 3 ersichtlich. Das zweidimensionale Netz wurde extrudiert und somit in Richtung Spannweite der Tragfläche verlängert.

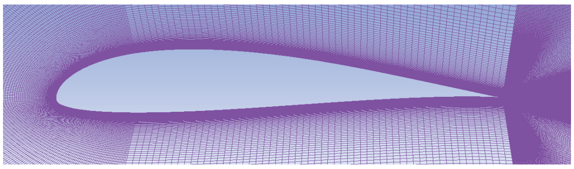


Abbildung 3: Netz entlang des Tragflügelprofils

Im Beispiel SD 7062 entstand so eine Gesamtzahl an Elementen von 739957. Der Einlass, die zwei Symmetrieebenen und die Wand der Tragfläche wurden als Einzelteile definiert.Wie in Abbildung 4 zu sehen ist, liegen alle $y^+$-Werte an der Oberfläche des Flügels unter 0.2.

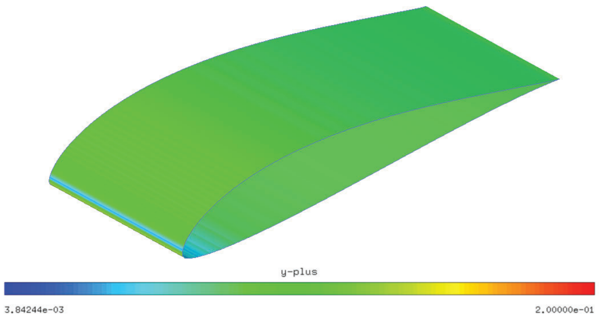


Abbildung 4: $y^+$-Werte an der Oberfläche des Flügels

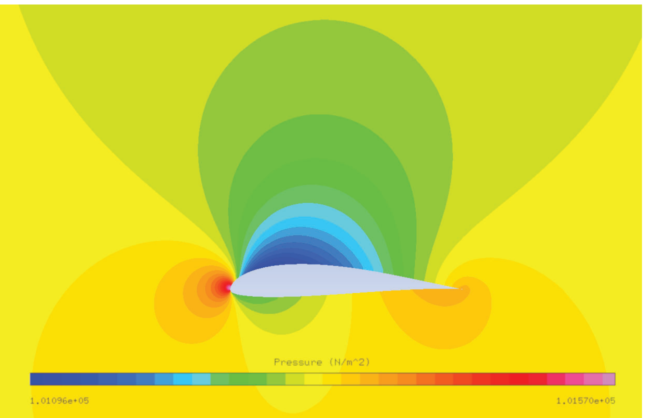


Abbildung 5: Druckprofil SD 7062 bei 0° Anstellwinkel

## 2.2 Berechnungen in FENSAP-ICE

Das in ICEM CFD erstellte Netz wurde in das Airflow Modul in FENSAP-ICE geladen. Anschließend konnten die Einstellungen getroffen werden. Die Strömungslösungen an allen Profilen wurden mit den Navier-Stokes-Gleichungen und dem Spalart-Allmaras Turbulenzmodell berechnet. Für das SD 7062 Profil wurde eine Umströmungsgeschwindigkeit von 20 m/s gewählt. Die Flügeltiefe betrug 0.3 m und Druck und Temperatur wurden nach der internationalen Standardatmosphäre bei 15°C Umgebungstemperatur gewählt. So ergab sich eine Reynolds-Zahl von 410450. Es wurde mit einer CFL-Nummer von 50 gerechnet. Es wurden 500 Iterationen durchgeführt, damit die Lösung konvergiert. In Abbildung 5 ist das Druckprofil am SD 7062 Profil bei einem Anstellwinkel von 0° ersichtlich.

Nachdem eine Strömungslösung berechnet wurde, konnte diese für die Berechnung der Tröpfchenlösung verwendet werden. Hier wurden die Parameter der Strömungslösung importiert. Die CFL-Nummer war 20 und es wurden 300 Iterationen durchgeführt. Die Berechnung der Vereisungslösung wurde mit ähnlichen Bedingungen wie in der Arbeit Analysis of Numerical Simulations of Icing on Airfoils durchgeführt. Die hierfür benötigten Parameter sind in Tabelle 1 ersichtlich.

Vereist wurde unter den in Appendix C enthaltenen Bedingungen. Die Details hierzu sind im CS-25 Dokument der European Union Aviation Safety Agency (EASA) zu finden. Diese Bedingungen müssen für die Zertifizierung erfüllt werden. [Eur21] Die Geschwindigkeit der Luft war 20 m/s und der Anstellwinkel betrug 0°. Die statische Lufttemperatur (static air temperature) betrug -10°C. Der durchschnittliche Durchmesser der Tröpfchen (median volume diameter) betrug 20 $\mu$m und der Anteil an flüssigem Wasser in den Tröpfchen (liquid water content) wurde als 0.5 g/m³ gewählt. Die gewählte Dauer der Vereisung war 1600 s. [Luc22]

Tabelle 1: Bedingungen für die Vereisungssimulation in FENSAP-ICE

| Profil | SD 7062 |
|---|---|
| Vereisungsbedingung | Appendix C |
| Geschwindigkeit | 20 m/s |
| Anstellwinkel | 0° |
| SAT | -10°C |
| MVD | 20 $\mu$m |
| LWC | 0.50 g/m³ |
| Dauer | 1600 s |
| Mach | 0.06 |

Nach einer Vereisungszeit von 1600 Sekunden ergab sich die in Abbildung 6 ersichtliche Eisschicht. Es ist erkennbar, dass sich das meiste Eis an der Flügelvorderkante ansammelte.

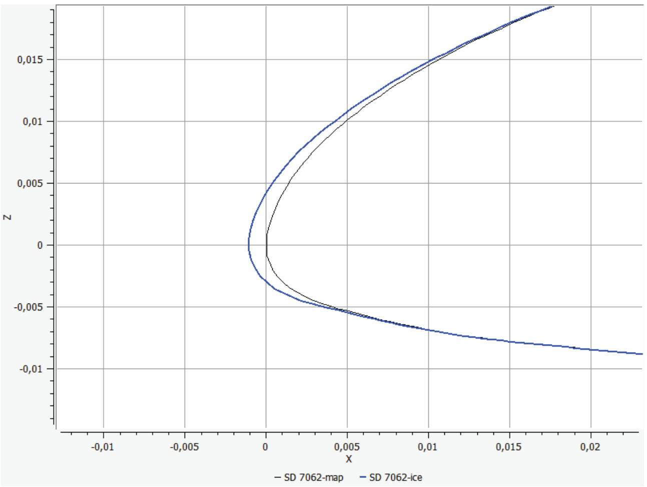


Abbildung 6: Eisschicht am SD 7062 Profil

## 3 Zusammenfassung und Ausblick

In FENSAP-ICE können mit einem gewissen Grundwissen schnell und einfach Vereisungssimulationen durchgeführt werden. Die Ergebnisse müssen anschließend kritisch betrachtet und auf Richtigkeit geprüft werden. Bei der Vereisung des SD 7062 Profils war die Auflösung des Netzes an der Oberfläche sehr fein, wodurch die Genauigkeit der Lösung verbessert wurde. Das meiste Eis hat sich in der Zeit von 1600 s an der Vorderkante des Flügels angesammelt.

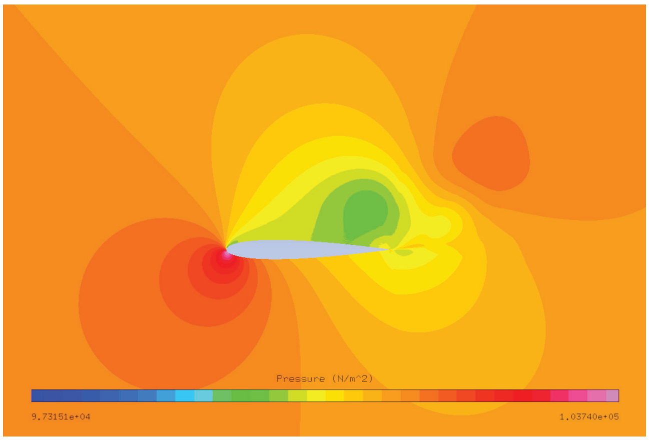


Abbildung 7: Druckprofil NACA 0012 bei 15° Anstellwinkel [Luc22, S.22]

In Zukunft kann mit kritischen Parametern gerechnet werden. Ein Beispiel hierfür ist in Abbildung 7 zu sehen. Die Anströmung erfolgte hierbei in einem Winkel von 15° auf ein NACA 0012 Profil. Weiters könnte eine Lösung mit der Berücksichtigung von integrierten Vorrichtungen zur Wärmeübertragung berechnet werden.

## Literatur

[Air23] Airfoiltools.com, 2023. [13.02.2023].

[And95] John D. Anderson. *COMPUTATIONAL FLUID DYNAMICS: The Basics with Applications.* McGraw-Hill Book Co., New York, USA, 1995.

[Ban20] Stephan E. Bansmer. *Aircraft icing: A challenging problem of fluid mechanics.* Cuvillier Verlag, Göttingen, Germany, 1 edition, 2020.

[Eur15] European General Aviation Safety Team. In flight icing: For general aviation pilots. European Aviation Safety Agency, Juli 2015. [10.02.2023].

[Eur21] European Union Aviation Safety Agency. *Certification Specifications and Acceptable Means of Compliance for Large Aeroplanes (CS-25)*, November 2021.

[Kuh14] Hendrik Kuhlmann. *Strömungsmechanik: Eine kompakte Einführung für Physiker und Ingenieure.* Pearson Deutschland GmbH, Hallbergmoos, Germany, 2 edition, 2014.

[Luc22] Sarah Luckeneder. Analysis of numerical simulations of icing on airfoils. FH JOANNEUM Luftfahrt/Aviation, 2022.

[Mon19] Francesco Montomoli, editor. *Uncertainty Quantification in Computational Fluid Dynamics and Aircraft Engines.* Springer International Publishing AG, Cham, Switzerland, 2 edition, 2019.

# Three-dimensional simulation of in-flight icing on an aerofoil using Ansys FENSAP-ICE

Maximilian Angermayr[1]

[1]FH JOANNEUM – University of Applied Sciences, Institute Luftfahrt/Aviation, Alte Poststraße 149, 8020 Graz, Austria

## Abstract

In this paper the icing tool FENSAP-ICE is inspected and used for predicting the three-dimensional in-flight ice growth on the airfoil NACA 23012 exposed to two different conditions. The airfoil profile is first created using the CATIA V5 design program. Then, Ansys ICEM CFD is used to add the geometry of the icing wind tunnel (IWT) and to generate a mesh for the three-dimensional system. In Ansys FENSAP-ICE, the two icing conditions are simulated using the singleshot method. The results are evaluated and plotted in MATLAB. In addition, the lift as well as drag coefficient of the airfoil NACA 23012 with 1.5° and 3° angle of attack (AOA) are calculated and compared with the analysis software XFOIL. The results from the icing simulations are compared with those from the real tests, which were carried out in a IWT and were made available by the JOICE research project.

## 1 Introduction

The icing of an aircraft in flight is a significant safety problem in aviation. The article by [CDG00] analyses, that there had been no significant advances in theoretical studies of ice formation on an aircraft until the early 1970s with the advent of the computer age, although this problem exists since the beginning of aviation. Growing ice on an aircraft can affect the flight dynamics in several ways. On the one hand, it affects the airfoil shape, which can result in unusual behaviour in terms of stall. On the other hand, control elements such as the trims can be negatively affected. The overall weight of the aircraft is also increased by additional ice. Therefore, it is of great importance to keep the aircraft free of ice even during flight.

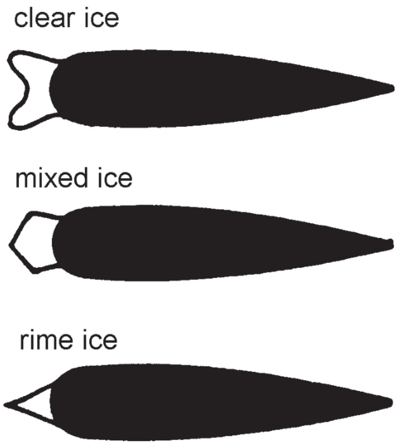


Figure 1: Ice accumulation on the leading edge of the wing [Sch13, p. 3]

On the ground, a chemical liquid is already applied when there is an increased risk of icing. In the air, a distinction is made between electrical, pneumatic, chemical and thermal deicing processes. In order to use such systems efficiently and to be able to make predictions about different wing geometries and their behavior during icing conditions, real experiments in the air by flight tests, on the ground by using IWT or analyses by numerical flow calculations are necessary. The calculations are now discussed in this paper.

### 1.1 Icing conditions

Ice formation is explained in [Sch13] by supercooled water droplets. An aircraft may encounter several forms of ice-producing precipitation. In reality, water droplets condense on solid nuclei to form a cloud. These cloud droplets can exist in the supercooled state down to a temperature of -20 °C due to their relatively small size, and very small droplets can exist down to a temperature of -40 °C. The dimension of supercooling depends on the purity of the water, the size of the droplet, and whether ice particles are already present in the cloud. Thus, water droplets can still exist in the liquid phase below 0 °C, which is why the term supercooled water was introduced. During the cooling process of the water droplets, they remain undisturbed, so that ice formation is not initiated. If the supercooled droplets encounter a solid body, they change their aggregate state and from then on appear in solid phase. In addition, they adhere to the object which triggered the aggregate state by contact. Fig. 1 shows the formation of ice on an airfoil, which strongly influences the geometry of the airfoil.

Depending on the prevailing external conditions, different ice forms can form. Clear ice forms at temperatures down to -10 °C and causes a change in the cross-section of the wing. Rime ice forms at -20 °C. This state leads to a pointed leading edge of the wing and thus changes the aerodynamic properties of the wing. Between these two forms, mixed ice is formed, which combines the properties of clear ice and rime ice. In fig. 1, the three different ice shapes and their effect on the wing cross-section can be viewed.

### 1.2 Computational Fluid Dynamics

According to [JGL18], the Computational Fluid Dynamics (CFD) is the computer-aided calculation for the analysis of physical phenomena of fluids in a system under consideration. It arises from various disciplines of fluid mechanics, thermodynamics, mathematics as well as computer science and is used in several fields such as process, chemical, civil and environmental engineering. In most problems that can be solved with simulation, a fluid flow is studied by numerical simulations using computer programs or software packages. In this process, the calculations are performed on high-performance computers to obtain a numerical solution.

During this process, the mathematical equations of fluid mechanics are solved numerically in an attempt to obtain an approximately correct result. For the discretization, initial and boundary conditions have to be predefined. Finally, the system of equations is solved with the help of a matrix solver. Computer programs support the user with regard to the input of parameters, execution of the calculation and presentation of the results.

## 2 Methods

When generating ice, it is important to consider several methods for ice generation on one object in order to be able to subsequently compare them with each other. In the following, the two test setups Design Point (DP) 42 and DP 66 with the parameters that can be taken from tab. 1 are now considered for the wing profile NACA 23012. For this purpose, a suitable airfoil is first created in a Computer Aided Design (CAD) software and then a computational mesh is generated in Ansys ICEM CFD. The actual calculation takes place in FENSAP-ICE and the results are evaluated and compared by using MATLAB.

### 2.1 Creation of geometry using CATIA V5

In order to perform a simulation, a geometry of the airfoil and the IWT must first be prepared. The dimensions of the experimental setup are given in fig. 3. The chord length of the airfoil is 1.2 m and the airfoil width is 2 m. The data points for the airfoil NACA 23012 were generated using [Air22] and saved in an EXCEL sheet. The

Table 1: Input parameters for the simulations

| **DP** | **Icing State** | **Speed** | **AoA** | **SAT** | **TAT** | **MVD** | **LWC** | **RH** | **Duration** |
|---|---|---|---|---|---|---|---|---|---|
| | | m/s | deg | K | K | microns | g/m$^2$ | % | s |
| 42 | App. C | 60.0 | 1.5 | 268.05 | 269.95 | 20 | 0.52 | 92.70 | 1800 |
| 66 | FZRA | 80.0 | 3 | 262.45 | 266.35 | 535 | 0.25 | 79.30 | 900 |

geometry for the wind tunnel was added later in the mesh generator. In CATIA V5, the Digitized Shape Editor was used to import the data points from the Excel spreadsheet. It was necessary to take care of the correct arrangement and separation of the coordinates in the file, because standardized both the X-coordinates and the Y-coordinates are listed in a common column. After the data points were imported, they could be connected to a spline in the Generative Shape Design menu. The profile trailing edge was merged into a point. [Air22] creates a profile with a length of 100 mm. Therefore, this profile must be scaled to the desired chord length of 1200 mm. To generate the AOA for the two test setups, the scaled profile was rotated around the Z-axis. The result can be seen in fig. 2. In total, two geometries were created, extruded to the specified width and saved as a *step* file.

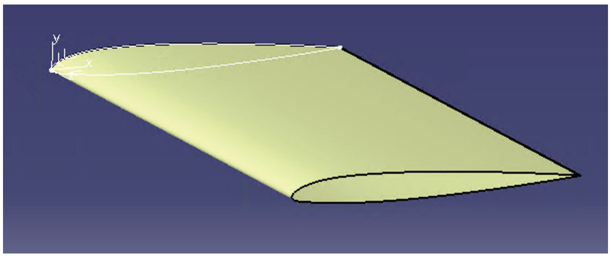


Figure 2: Three-dimensional airfoil created with CATIA V5

## 2.2 Mesh generation in Ansys ICEM CFD

The previously generated *step* file can now be loaded as a profile into the meshing tool ICEM CFD. Since the AOA for both airfoils is minimally different, the meshing could be performed equally for both geometries. The geometry of the IWT was created using the shape of a cuboid. To mesh the geometry, a block was first created for the entire geometry. The block type chosen was *3D Bounding Box* since it was a three dimensional geometry. For further block division, the *Ogrid Block* option was used, as it is best suited for meshing a wing. This generates multiple blocks around the wing that are arranged in a C formation. The generated side edges of the blocks were associated with the lines of the airfoil. Depending on the location of the block edges, different meshing settings were made and multiple cells were generated for the entire system. After the mesh quality settings were made, it was necessary in ICEM CFD to convert the first created block structured hexahedral mesh into an unstructured mesh. In order to save the created mesh as a *GRID*, the solver *Fluent* had to be chosen first, since it has the best compatibility with FENSAP-ICE. Fig. 4 shows a detailed view of the completed mesh at the leading edge of the wing for the IWT. For calculation and ice generation in FENSAP-ICE, it is important to create a particularly fine mesh near the airfoil to get more accurate results for the ice accumulation. Since a three-dimensional geometry was also considered, 6.4 million cells were created during the creation of the mesh.

## 2.3 Simulation of the 3D geometry with the singleshot method

Three-dimensional meshes can have a large number of elements, which can significantly increase the computation time. For this reason, the singleshot method was chosen for the calculation with the three-dimensional geometries.

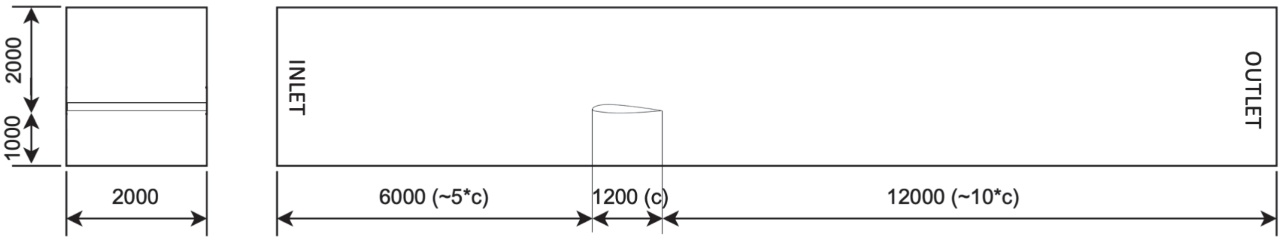


Figure 3: Dimensions of the icing wind tunnel including the airfoil

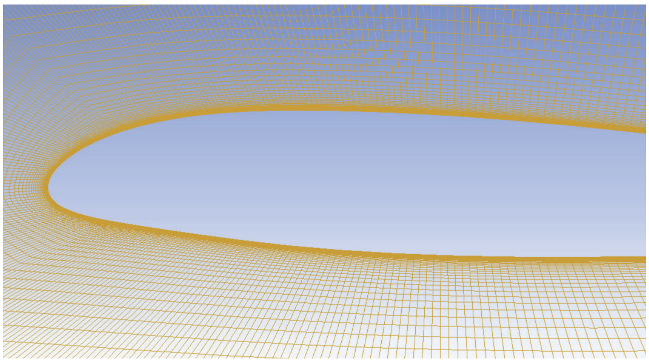


Figure 4: Detailed view of the mesh near the profile

FENSAP-ICE also offers the possibility to divide the total icing time into smaller intervals. This creates a new calculation with the previously generated ice shape after each iteration. This procedure is also known as the multishot method. The procedure for the calculation of DP 66 and DP 42 is generally the same. As can be seen from tab. 1, different input parameters were used for both simulations.

After a new project was created, the simulation could be set up. First, the calculation was performed with the FENSAP flow solver. In this simulation, a flow solution is generated using the continuity equations for the calculation of the mass, the Navier-Stokes equation for the momentum and the conservation of energy for the energy. To do this, the previously created mesh had to be imported into FENSAP-ICE. During the import, a submenu opened where the boundary conditions were specified. The smaller side surfaces were defined as *Inlet* with the inlet velocity given in tab. 1 and *Exit* with the pressure 0 Pa. All remaining geometries were assigned the boundary condition $WALL$ with the no-slip condition. After the import, the settings were made. Through the configuration icon, the submenu of FENSAP-Flow opened, where the parameter settings for the flow solutions can be made. In the first tab $Model$, the physical basic equations as well as the turbulence model were selected. For these calculations, the Spalart-Allmaras turbulence model, published by [SA92], with relaxation factor 1 was used. In the second tab, the parameters could be entered as they can be seen in tab. 1. Depending on the chord length, the air temperature and the ambient pressure, which was 101 325 Pa, FENSAP-ICE automatically calculates the Mach number (Ma), the Reynolds number (Re) and the total air temperature (TAT). The calculated values can be obtained from tab. 2.

Table 2: Calculated values from the flow simulation

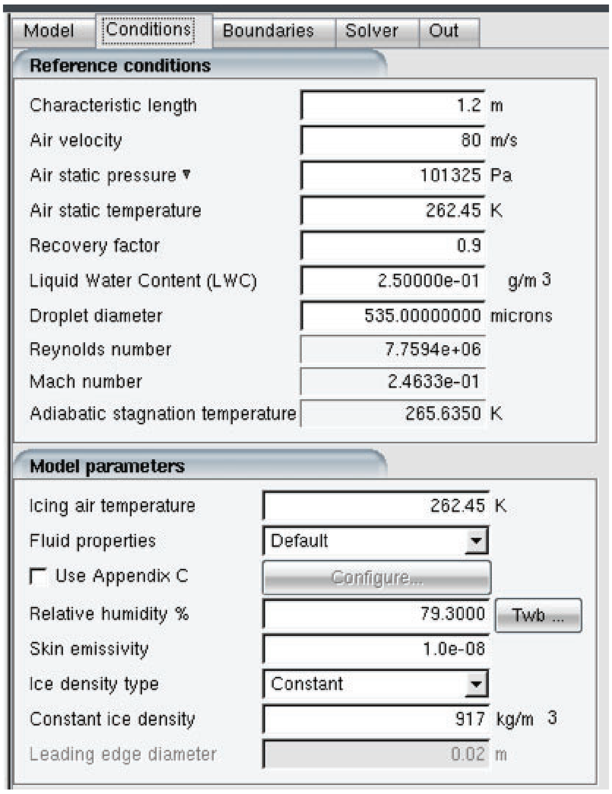
| **DP** | **Re** | **Ma** | **TAT**<br>K |
|---|---|---|---|
| 42 | $5.8169{\cdot}10^6$ | 0.185 | 264.241 |
| 66 | $7.7594{\cdot}10^6$ | 0.246 | 265.635 |

The next simulation was computed using the DROP3D solver, which computes based on the previously generated flow solution by dragging and dropping the file. The droplet distribution is calculated using the Navier-Stokes equations, supplemented by the continuity and momentum equations of the droplets. Most settings can be left in the default settings. The reference conditions for the droplets were taken according to tab. 1. As a special feature, a Langmuir-E droplet distribution for Freezing Rain (FZRA) was considered in the calculation for DP 42 according to tab. 3, as well as Appendix C for DP 66. The distributions published by [LBU46] were used by the National Advisory Committee for Aeronautics (NACA), later National Aeronautics and Space Administration (NASA), to determine the mean volumetric diameter cited in Appendix C, which is used for aircraft icing certification. Appendix C, according to [Jec02], is the data published in Appendix C of Part 25-Airworthiness Standards: Transport Category Airplanes in Title 14-Aeronautics and Space of the U.S. Code of Federal Regulations, which has been used since 1964 to select values of icing-related cloud variables for the design of in-flight anti-icing systems for aircraft.

Table 3: Droplet distribution for DP 42 with FZRA

| Droplet diameter<br>microns | Percentage<br>% |
|---|---|
| 10 | 15,0 |
| 50 | 11,0 |
| 200 | 13,0 |
| 500 | 28,5 |
| 1000 | 25,0 |
| 1500 | 7,5 |

For the icing, a simulation was performed with the solver ICE3D, which calculates the mass conservation and the energy conservation via two partial differential equations. This was based on the previously calculated flow and droplet solutions. Fig. 5 illustrates the settings that were made in ICE3D. Once all three calculations were successfully completed, the results could be analysed graphically in the *VIEWMERICAL* subroutine.



Reference conditions

Characteristic length 1.2 m

Air velocity 80 m/s

Air static pressure ▾ 101325 Pa

Air static temperature 262.45 K

Recovery factor 0.9

Liquid Water Content (LWC) 2.50000e-01 g/m 3

Droplet diameter 535.00000000 microns

Reynolds number 7.7594e+06

Mach number 2.4633e-01

Adiabatic stagnation temperature 265.6350 K

Model parameters

Icing air temperature 262.45 K

Fluid properties Default

□ Use Appendix C

Relative humidity % 79.3000

Skin emissivity 1.0e-08

Ice density type Constant

Constant ice density 917 kg/m 3

Leading edge diameter 0.02 m

Figure 5: Settings in ICE3D for DP 66

## 2.4 Importing the results into MATLAB

All results were presented in a Matlab script. The required points from the generated ice film were imported from FENSAP-ICE. The ice solution was opened with *VIEWMERICAL*. In this menu the points can be generated and saved. The points were saved in a TXT file. Since ice also formed on the walls of the IWT, these points had to be removed. In MATLAB, the files were imported with the *readtable* command. The next step was to convert the tables into arrays using the *table2array* command to facilitate the evaluation of the matrix. The same procedure was performed with the data points of the wing profile. To plot the results, the ice layer and the airfoil were generated in the same plot. One plot was generated for each of DP 42 and DP 66.

### 2.5 Calculation of the aerodynamic coefficients

To obtain the lift coefficient $c_l$ and drag coefficient $c_w$ of the airfoil NACA 23012 with 1.5° and 3° AOA, a simulation was also performed in FENSAP-ICE with a different mesh, which can be seen in fig. 6. Since the coefficients are determined by the pressure distribution, a larger area around the wing profile had to be created in order to achieve a better pressure distribution than that calculated with the mesh in fig. 4. This mesh has 120000 cells. FENSAP-ICE can calculate the aerodynamic coefficients using the flow solver, taking into account the reference surface. After the simulation, the value was chosen which was calculated during the last iteration step. The results from FENSAP-ICE were compared using XFOIL. XFOIL is an analysis tool for airfoils, wings and airplanes provided by [Dre00] as an open source package. In XFOIL those parameters were used which were applied for the calculation of DP 42 and DP 66. In addition, the Re calculated by FENSAP-ICE and shown in tab. 2 was used.

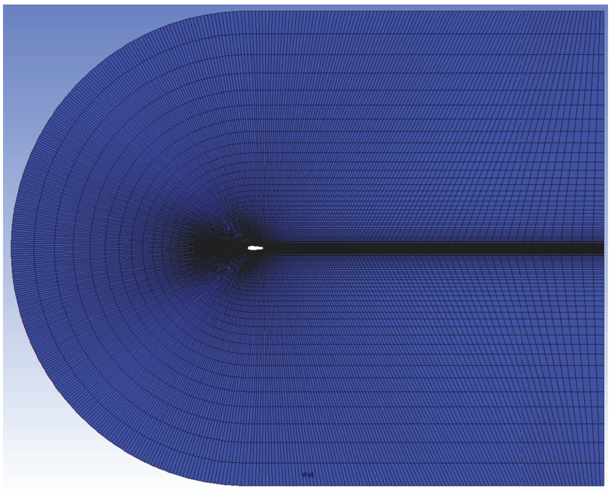


Figure 6: Mesh for the calculation of the lift and drag coefficient

## 3 Results

The aim of this paper was to visualise the ice growth on the wing profile NACA 23012 by a numerical flow calculation. All results collected during the simulations have been visualised using MATLAB. In the graphs presented, the two-dimensional view is used. For the calculations that took place in 3D, the cross section was placed in half of the profile. In the following section, the results generated by this work are now compared with the results published in [BDH$^+$22]. The IWT-tests as well as the calculations by the simulation programs TAC2 from [Aer22], ICEAC2DV3 from [Has19] and FENSAP-ICE from Ansys will be considered. For the aerodynamic coefficients, the results of XFOIL, FENSAP-ICE without and with ice accumulation for DP 66 and the results published by the [UIU23] database are compared.

### 3.1 NACA 23012 with DP 42

The first simulation was performed with the parameters of DP 42. The lift as well as the drag, which are defined by the aerodynamic coefficients, can be determined via the pressure distribution on the wing. The results can be seen in tab. 4. The wing profile without ice accumulation is considered.

Table 4: Aerodynamic coefficients for DP 42

| | $c_l$ | $c_w$ | Ice |
|---|---|---|---|
| **FENSAP-ICE** | 0,327 | 0,028 | no |
| **XFOIL** | 0,304 | 0,006 | no |
| **UIUC** | 0,380 | 0,013 | no |

In fig. 8 the mentioned methods for DP 42 are shown in a graph. For the calculation with TAC2, a temperature 0.5 °C lower than that which can be taken from tab. 1 was selected, as well as a Langmuir-E droplet distribution. For ICEAC2DV3, a monodisperse droplet distribution was used. In this work, due to this framework, a calculation with the Langmuir-E distri-

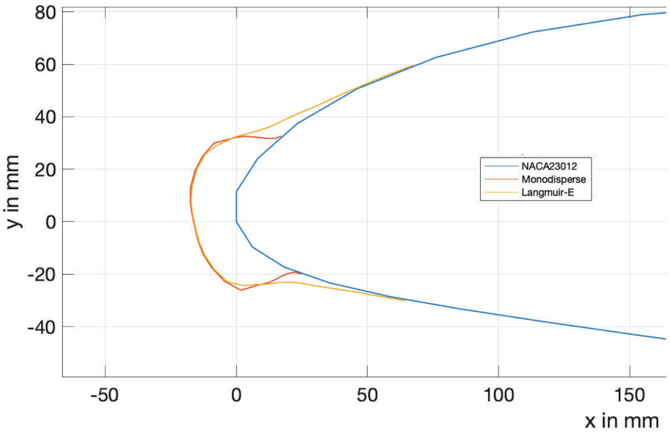


Figure 7: Ice accretion with DP 42 considering different droplet distribution in FENSAP-ICE

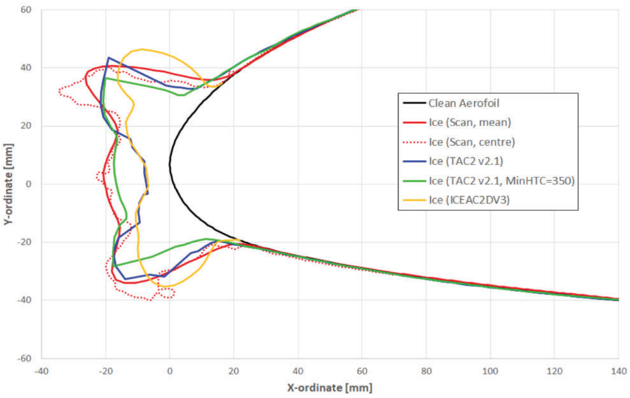


Figure 8: Ice accretion with DP 42 [BDH+22, p. 14]

bution was performed in addition to the simulation in DROP3D with a monodisperse distribution. A direct comparison can be seen in fig. 7. Both the monodisperse distribution and Langmuir-E show a similar icing pattern. In fig. 8, the blue line represents the result from the TAC2 calculation. In addition, a simulation was performed with a convective heat transfer coefficient, which is represented by the green line. The yellow line shows the ICEAC2DV3 prediction. The solid red line is the mean ice shape determined over the entire span, while the red dotted line represents the cross-section of the centerline at the IWT test. The calculations by FENSAP-ICE show the closest agreement with the thickness and position of the ice shape. However, there is no strong formation of horns in the ice.

## 3.2 NACA 23012 with DP 66

In the next step, DP 66 is considered. First, the aerodynamic coefficients were calculated, which can be taken from tab. 5.

Table 5: Aerodynamic coefficients for DP 66

| | $c_l$ | $c_w$ | Ice |
|---|---|---|---|
| **FENSAP-ICE** | 0,497 | 0,029 | yes |
| **FENSAP-ICE** | 0,477 | 0,009 | no |
| **XFOIL** | 0,489 | 0,005 | no |
| **UIUC** | 0,490 | 0,012 | no |

Furthermore, in fig. 10 the red line shows the ice shape by the real test, the yellow line shows the result of the simulation with ICEAC2DV3 and the blue line shows the result of the simulation with TAC2. The black line represents the wing profile. In fig. 9 the result of the calculation by the program FENSAP-ICE with the singleshot method is shown. The comparison shows a very good agreement with the ice thickness and position. However, similar to DP 42, the formation of ice grains does not occur. In addition, it should be mentioned that a thicker ice layer is formed at the bottom of the profile in the result of this work than in all results from fig. 10. This can be explained by the fact that the AOA was modified from 3° to 1.5° in the latter results. However, in [BDH+22] it is described that for the real tests no data on the pressure distribution were available, so that the actual aerodynamic AOA could not be determined.

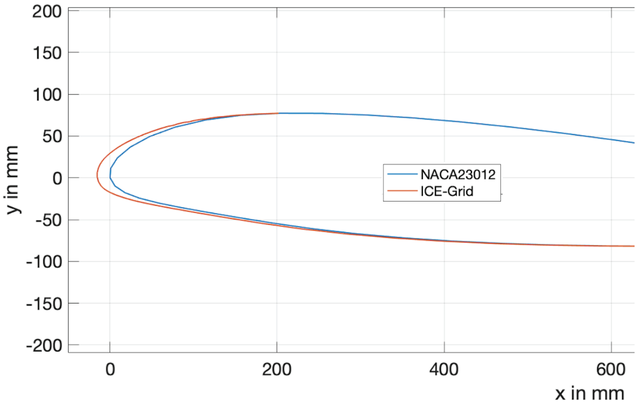


Figure 9: Ice accretion with DP 66 in FENSAP-ICE

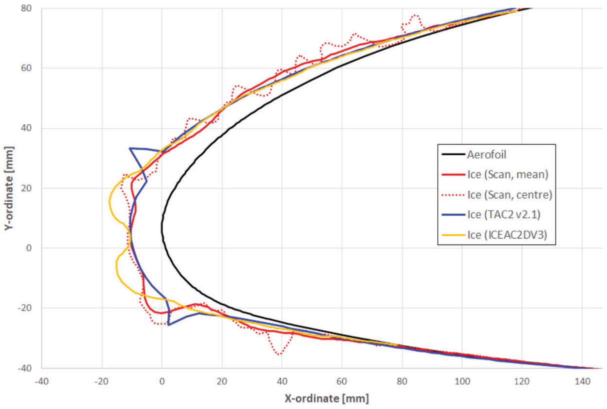


Figure 10: Ice accretion with DP 66 [BDH+22, p. 13]

## 4 Summary and Outlook

Three-dimensional numerical simulations require a considerable amount of computing power. Decisive for the required computing power and duration is the created mesh and the number of cells as well as nodes. Thereby the number of nodes increases strongly if several cells are created in the depth of the profile. Due to this fact, an attempt was made to minimize the number of cells for the three-dimensional mesh in order to improve the computing performance. In order to still generate a meaningful result, a high quality mesh was created near the airfoil, especially at the leading edge of the wing. For regions farther from the wing, the mesh quality was intentionally degraded. It should be noted that under uniform conditions, different results occur both in the different numerical simulations and in the real tests in the IWT. Despite modern technology, both ice generation in a wind tunnel and computer-aided calculations may still produce results which may differ. In [Ban20], the main reason given for this problem is turbulence. Due to its chaotic behavior, it is difficult to capture it and make a prediction. Therefore, several methods are applied and compared with each other. For this reason, wind tunnel tests remain an important part for generating results, especially in certification processes. For this procedure, a large geometry of the channel is necessary to minimize effects caused by the tunnel. Furthermore, the documentation of the generated ice layer, due to the transparency of the ice is another challenge, which is why own measurement methods have been developed for this purpose. The high technical effort and costs lead to the fact that many simulations are carried out in computer programs. At the moment, the difficulty is to analyze the parameters that are difficult to determine experimentally and to include them correctly in the calculation. [Ban20] assumes that there will be significant progress in this field in the next decades.

## References

[Aer22] AeroTex UK. *TAC2 – Trajectory and Catch*. 2022. URL: http://www.aerotex.co.uk/aircraft-icing/tac2/. [cited: 28.12.2022].

[Air22] Airfoil Tools. *NACA 23012 12%*. 2022. URL: http://airfoiltools.com/airfoil. [cited: 11.12.2022].

[Ban20] S. E. Bansmer. *Aircraft icing: A challenging problem of fluid mechanics*. 1st ed., Cuvillier Verlag, Göttingen, Deutschland, 2020.

[BDH$^+$22] W. Breitfuß, S. Diebald, W. Hassler, S. Humpel, P. Kollmann, D. Kozomara, R. Moser, T. Neubauer, R. Puffing, B. Reinholz, S. Schweighart, and A Tramposch. *Generation of Validation Data for an Electrothermal Ice Protection System*. AIAA AVIATION Forum, Chicaco, 2022.

[CDG00] J.T. Cansdela, N. P. Dart, and R. W. Gent. *Aircraft icing*. Philosophical Transactions of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences, London, 2000. Vol. 9, pp. 2873 – 2911.

[Dre00] M. Drela. *XFOIL: Subsonic Airfoil Development System*. 2000. URL: https://web.mit.edu/drela/Public/ web/xfoil/. [cited: 07.01.2023].

[Has19] W. Hassler. *A Refined In-Flight Icing Model and its Numerical Implementation*. SAE Technical Paper 2019-01-1937, 2019. URL: https://doi.org/10.4271/2019-01-1937. [cited: 28.12.2022].

[Jec02] R. K. Jeck. *Icing Design Envelopes (14 CFR Parts 25 and 29, Appendix C) Converted to a Distance-Based Format*. U.S. Department of Transportation, Federal Aviation Administration, 2002.

[JGL18] T. Jiyuan, H. Y. Guan, and Chaoqun L. *Computational Fluid Dynamics: A Practical Approach*. Butterworth-Heinemann, Oxford, UK, 2018.

[LBU46] I. Langmuir, K. Blodgett, and United States. Army Air Forces. *A mathematical investigation of water droplet trajectories.* Army Air Forces Headquarters, Air Technical Service Command, Washington, D.C, USA, 1946.

[SA92] P. Spalart and S. Allmaras. *A One-Equation Turbulence Model for Aerodynamic Flows.* AIAA AVIATION Forum, Chicaco, 1992. Vol. 439.

[Sch13] D. Scholz. *Flugzeugsysteme: Eis- und Regenschutz. ATA 30.* Vorlesungsskript. Hamburg University of Applied Sciences, Hamburg, 2013.

[UIU23] UIUC Applied Aerodynamics Group. *UIUC Airfoil Data Site*. 2023. https://m-selig.ae.illinois.edu/ads/coord/database .html [cited: 28.03.2023].

## MERKUR INNOVATION LAB

Im Herbst 2020 gründete die Merkur Versicherung ihr konzerneigenes Start-up. Seitdem kümmert sich das junge Team um Machine Learning-Lösungen und Künstliche Intelligenz, streckt seine Fühler aber auch extern aus. Partnerschaften mit Forschungseinrichtungen und die breite Vernetzung in der Szene lassen das Start-up weiterwachsen. Damit entwickelt sich die Ideenschmiede aus Graz auch immer mehr zu einem Ausbildungshub für Fachkräfte im Bereich Data Science.

„*Unsere Aufgabe besteht in erster Linie darin, für unsere Kunden aus vorhandenem Datenschatz und Erfahrungswissen einen Mehrwert zu schaffen*“, erklärt Daniela Pak-Graf, Geschäftsführerin vom Merkur Innovation Lab. „*Wir nehmen die verschiedenen Datenquellen und vernetzen sie, so bekommen wir Einblicke, die niemand vorher hatte. Kunden wünschen sich neue Dienstleistungen, für die bisher bloß keiner die Idee hatte. Und in den Daten stecken die Lösungen.*“

Aus einer Handvoll Mitarbeitern wurde rasch ein dynamisch wachsendes Team, das mittlerweile zu zehnt in der Datenwerkstatt tüftelt. Und die Reise geht weiter. Daniela Pak-Graf sucht bereits nach neuen Data Scientists, nach Mathematikern und IT-Systementwicklern, die Zukunftsthemen auf Businesschancen durchleuchten. Damit schafft das Start-up neue Jobs, baut Talente auf und hat sich zu einem Ausbildungshub entwickelt, das Data Scientists in die wirtschaftliche Praxis holt.

Auch der Markt reagiert: Was als Prototyping-Plattform begonnen hat, entwickelt sich kontinuierlich weiter. Ging es dem Innovation Lab anfangs ausschließlich um hauseigene Versicherungsprodukte und Services, will man künftig das Know-how auch extern anbieten.

*©iStock*

# Scientific Computing 2023

## Track: Mathematik

*Fabian Tschofenig, Daniel Reisinger und Georg Jäger:* **Exploring complex contagions on progressively sparse networks: A study of non-linear graph reduction** ........ 126

# Exploring complex contagions on progressively sparse networks: A study of non-linear graph reduction

Fabian Tschofenig[1], Daniel Reisinger[1], and Georg Jäger[1]

[1]University of Guaz, Institute of Environmental Systems Sciences

## Abstract

A contagion can be conceptualized as the process of spreading an entity such as a virus, emotion, or information through a group or network. Depending on the kind of contagion, not all connections may contribute equally to the spreading success of a contagion. In this regard, it can be distinguished between a simple and a complex contagion where the peculiarity of a complex contagion is that the potential reachability of nodes depends not only on direct connections but also on the presence of local common neighborhoods. This makes it difficult to understand the pathway as well as the extent to which a complex contagion spreads through a network. Simplifying the network structure in order to preserve only feasible connections seems therefore highly rational. In this paper, we propose a novel graph reduction method that reduces a network to contain only connections relevant to the spreading phenomena of a complex contagion process. Results show that this reduced network can effectively inform about the spreading potential of a complex contagion prior to its outbreak.

## 1 Introduction

Contagion processes describe the spread of viruses, malware, behavior, emotions, memes, information, and many other entities [Bar16]. These processes can occur in various settings and can have a significant impact on communities and societies in the form of epidemics [CWW$^+$03, BHG$^+$09, HBG04], cyber-attacks [SGRL11, ZJ12, WGHB09], and social movements [SA15, MSFL17], among others [CF07, ZC19, Cen21]. Understanding the mechanisms behind contagion processes and how they can be controlled is thus crucial for preventing negative outcomes and addressing associated challenges.

Contagion processes are largely determined by two factors: the spreading mechanism of a given entity such as a virus, emotion or information, and the underlying network structure on which the entity spreads [Bar16]. Depending on the specifics of the spreading mechanism, we can distinguish between a simple and a complex contagion [CM07, CEM07, GC21].

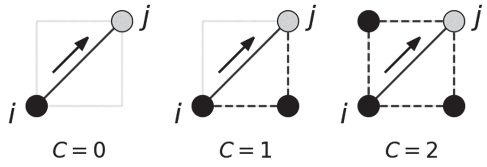


Figure 1: Simple versus complex contagion. The minimum number of common neighbors required for spreading is denoted by the complexity $C$. A simple contagion (left, $C = 0$) requires only a single connection to spread a virus from an infected node $i$ (black) to a susceptible node $j$ (gray). A complex contagion requires one (center, $C = 1$), two (right, $C = 2$), or more additional infected common neighbors (black) between node $i$ and node $j$ before further spreading occurs.

In simple contagions, such as the spread of viruses, a single connection between two individuals is sufficient to spread the virus from one individual to another [IPBL19]. Contrary to this, complex contagions, such as the spread of a social movement, require a susceptible individual to be connected to multiple infected spreaders simultaneously before spreading continues [IPBL19]. A visual example emphasizing the difference between simple and complex contagions is shown in Fig. 1. Due to the restrictive nature of complex contagions, individual connections alone do not guarantee spreading success. Nodes reachable through direct connections in a network may not be part of any feasible path in a complex contagion process as spreading requires additional spreaders in the local common neighborhood between two nodes. In this context, [CM07] speak of the need for wide enough bridges to facilitate complex spreading. This makes it difficult to understand the exact pathway as well as the extent to which a complex contagion spreads through a network.

In addition, many network theoretical metrics are not implemented to account for the complex paths observed in complex contagions. For example, the size of the giant component is an informative metric for the spreading potential of a simple contagion as it indicates how much of the network will ultimately be infected should the contagion start in any part of the network's giant component [Bar16]. It is, however, not an informative metric for the spreading success of a complex contagion as some parts of the giant component might not be supported by wide enough bridges to facilitate further spreading.

Similarly, many centrality measures such as betweenness, closeness, and eigenvector centrality turn out to be poor predictors for capturing so-called superspreaders in complex contagion processes [GC21]. This is due to their reliance on the concept of shortest path length which is currently not well defined for complex contagions. As a result, centrality measures that use the conventional definition of shortest path length are prone to incorrectly identifying the most important contributors to the spreading success of a complex contagion [GC21].

It appears evident that there is a need to bridge the gap between network theoretical concepts based on simple paths and their applicability to complex contagions. Simplifying a given network structure in order to preserve only feasible connections seems therefore highly rational. In this paper, we propose a novel graph reduction method that reduces a network to contain only connections that can potentially be utilized during the spreading process of a complex contagion. The complexity reduced network may then serve again as informative basis for the application of conventional network theoretic concepts.

# 2 Methodology

The potential benefit of introducing complexity reduced networks is evaluated using the following methodology: First, we formalize our method of complex graph reduction. Second, we present a simplistic model to simulate contagion dynamics on various graphs. Third, we describe a procedure to generate progressively sparse graphs ranging from high to low connectivity. Fourth and final, we describe our simulation setup with which we investigate how conventional network properties calculated on complex graph reductions compare to complex contagion simulations as network connectivity is shifted.

## 2.1 Complex Graph Reduction

The transformation operator $T(\mathbf{A})$ transforms the adjacency matrix $\mathbf{A}$ of a given graph into $\mathbf{A}^C$, the adjacency matrix adjusted for a complex contagion of complexity $C$.

$$\mathbf{A}^C = T(\mathbf{A}) \tag{1}$$

In simple graphs, the adjacency matrix $\mathbf{A}$ of a graph with number of nodes $n$ is of dimension $n \times n$ with $\mathbf{A}_{ij} = 1$ if node i is adjacent to j, else $\mathbf{A}_{ij} = 0$. Two nodes are adjacent to each other if they are connected via a common edge. By introducing the Kronecker delta and the Heaviside step function, the transformation of $\mathbf{A}$ to $\mathbf{A}^C$

can be defined for every matrix element $\mathbf{A}^C_{uv}$ as shown in Eq. 4, where the complexity $C$ represents the minimum required number of common neighbors [Pop08, Dav02].

$$\delta_{ij} = \begin{cases} 1, & \text{if } i = j, \\ 0, & \text{if } i \neq j. \end{cases} \tag{2}$$

$$\mathcal{H}(x) = \begin{cases} 1, & \text{if } x > 0, \\ 0, & \text{if } x \leq 0. \end{cases} \tag{3}$$

$$\mathbf{A}^C_{uv} = \mathcal{H}\left[\mathbf{A}_{uv}\Big(-C + \sum_{k=1}^{dim(\mathbf{A})} \mathbf{A}_{ku}\delta_{\mathbf{A}_{kv},1}\Big)\right] \tag{4}$$

Figure 2 shows an example of the graph reduction operator for various complexities $C$ applied to a given graph with adjacency matrix $\mathbf{A}$.

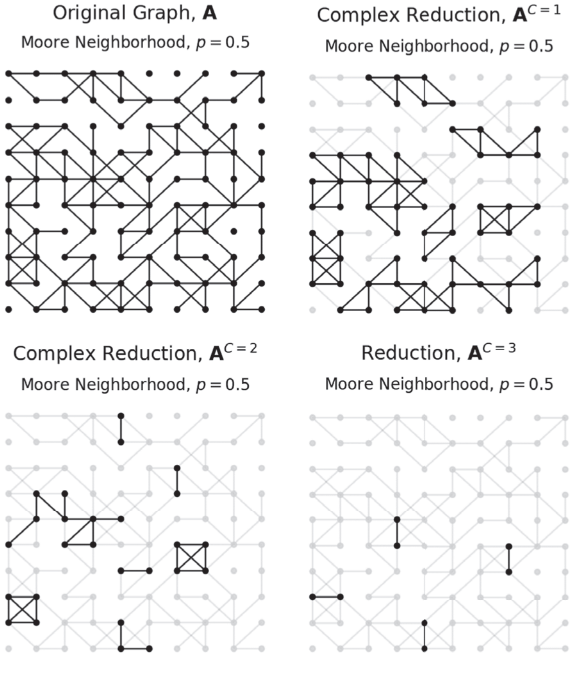

Original Graph, **A**
Moore Neighborhood, $p = 0.5$
Complex Reduction, $\mathbf{A}^{C=1}$
Moore Neighborhood, $p = 0.5$
Complex Reduction, $\mathbf{A}^{C=2}$
Moore Neighborhood, $p = 0.5$
Reduction, $\mathbf{A}^{C=3}$
Moore Neighborhood, $p = 0.5$

Figure 2: Complex graph reduction. Shows preserved edges (black) and removed edges (gray) after applying the complex graph reduction operator of complexity $C$ to a Moore neighborhood network in which half of the edges were deleted (denoted by $p = 0.5$) prior to the graph reduction.

## 2.2 Contagion Model

To simulate contagion dynamics we use a simplistic model that assumes that a susceptible node in a network becomes infected when connected to other infected nodes. Once a node is infected, it remains infected and cannot recover. A contagion simulation is initialized by infecting nodes in the leftmost column of a Moor neighborhood network and the simulation ends when no new nodes can be infected (see Fig. 3). The spreading success of a contagion process of complexity $C$ is expressed by $d_C$ which denotes the portion of infected nodes in a given network at the end of simulation. In the example in Fig. 3, the simple contagion, i.e. $C = 0$, spreads to 96 percent of the network. In comparison, the complex contagion requiring one additional infected common neighbor, i.e. $C = 1$, reaches just 25 percent of the network. The spreading potential lowers even further for contagions of higher complexities with $C = 2$ only reaching 12 percent of the network and $C = 3$ showing no spreading at all.

## 2.3 Bernoulli Percolation

To simulate contagion processes in a variety of differently connected network structures, we use a procedure called Bernoulli percolation in which every edge is a Bernoulli random variable that can either be open or closed, where closed refers to the edge being removed [DC18]. More specifically, we start with a Moore neighborhood network where every node is connected to its 8 surrounding neighbors and assign every edge a random number $r$ in the half-open interval $[0, 1)$. An edge is closed or removed if $r$ is smaller than a chosen probability threshold $p$, i.e. $r < p$. Varying $p$ from 0 to 1, hence, changes the network structure from a fully connected Moore neighborhood to isolated nodes. [1]

[1] In the context of contagion dynamics, this shift in connectivity is associated with a sudden state transitions between a state where all nodes become infected and a state where almost none of the nodes become infected.

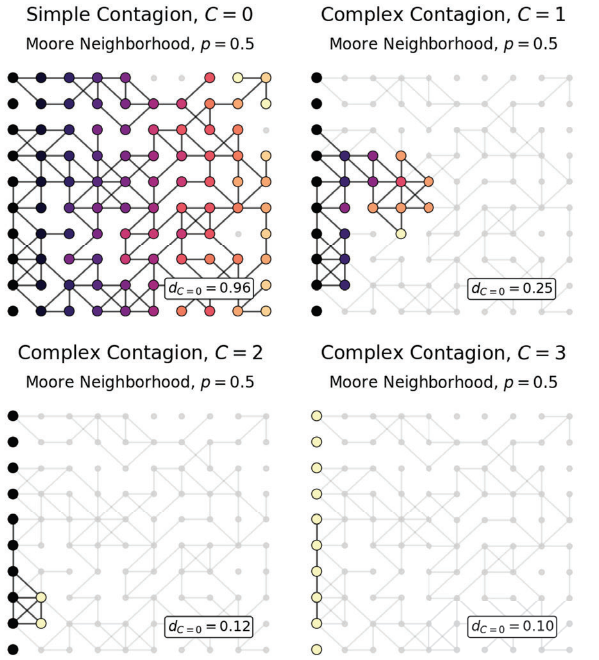

Simple Contagion, $C = 0$
Moore Neighborhood, $p = 0.5$
$d_{C=0} = 0.96$
Complex Contagion, $C = 1$
Moore Neighborhood, $p = 0.5$
$d_{C=0} = 0.25$
Complex Contagion, $C = 2$
Moore Neighborhood, $p = 0.5$
$d_{C=0} = 0.12$
Complex Contagion, $C = 3$
Moore Neighborhood, $p = 0.5$
$d_{C=0} = 0.10$

Figure 3: Contagion model. Shows contagion simulations of different complexities $C$. The spreading simulations are initialized in the left-most column. The temporal component of the contagion corresponds to the dark to light color gradient.

### 2.4 Simulation Setup

Our simulation setup can be outlined as follows: We begin by constructing a Moore neighborhood network containing 30 by 30 nodes and assign a random number $r$ in the half-open interval $[0, 1)$ to every edge in the network. We then construct modified networks by iteratively increasing a probability threshold $p$ from 0 to 1 in 100 equal sized steps and remove edges from the original Moore neighborhood network if $r < p$. For every network constructed in this fashion[2], we (a) simulate simple and complex contagion dynamics for which we calculate the percentage of infected nodes $d_C$, and (b) construct corresponding complex reduced networks for which we calculate a list of network metrics. The metrics calculated are the size of the giant component $G$, the number of components $n$, the mean component size $\mu$, and the standard deviation of component sizes $\sigma$. We then examine, in the form of correlation tables, whether the network metrics in a complex reduced graph of complexity $C$ inform about the spreading success of the corresponding complex contagion of equal complexity $C$ in the original network. The whole simulation setup is performed 100 times and results are averaged.

## 3 Results

Our results show that complexity reduced networks can effectively aid in informing about the spreading potential of a complex contagion prior to its outbreak. In particular, we find that network metrics calculated on a complexity reduced network $\mathbf{A}^C$ correlate well with the percentage of infected nodes $d_C$ of a complex contagion with equal complexity $C$. This is shown in Fig. 4 in the form of graphs and in Tables 1 through 3 in the form of Pearson correlation coefficients [SBS18]. We find that the giant component $G$, the mean component size $\mu$, and the standard deviation of component sizes $\sigma$ of a complexity reduced network $\mathbf{A}^C$ show high correlation with the spreading potential $d_C$ of a complex contagion of equal $C$.

In this regard, the giant component $G$ seems to be especially informative about the spreading potential of a complex contagion. This is particularly interesting considering that an infection of some initial cluster in the complexity reduced giant component does not necessarily result in an infection of the whole complexity reduced giant component. Simple contagion processes spread along direct connection in the network. Hence, if the contagion is initialized in any part of the simple giant component, the contagion will ultimately infect every node in that component.

[2] By using complex graph reductions on a variety of networks derived by Bernoulli percolation, we can benchmark our network parameters on a wide range of structures. This allows us to analyze its effectiveness across a wide range of configurations, despite the fact that the Moore neighborhood network is the same for each simulation.

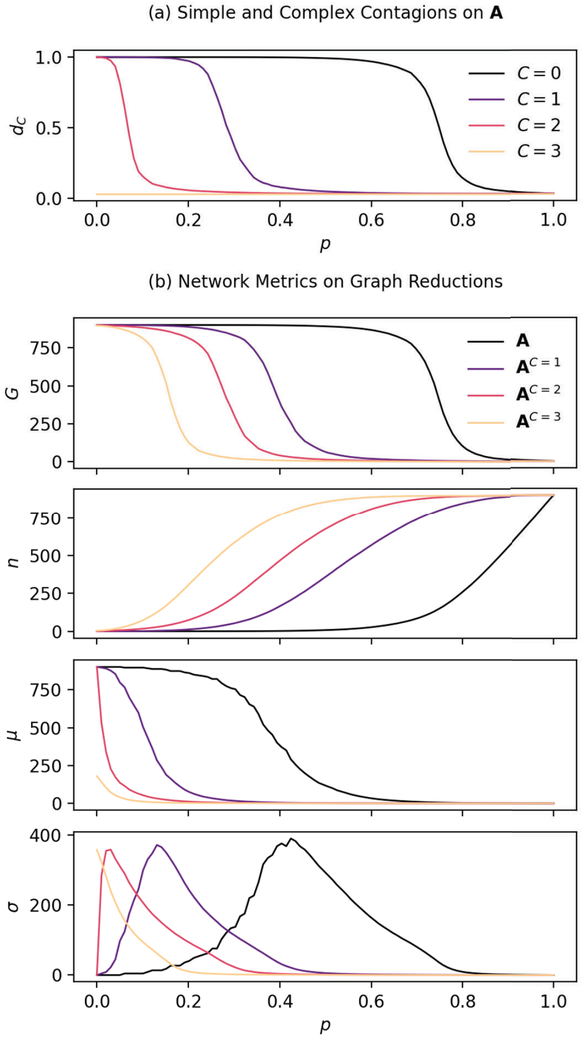


Figure 4: Comparison of contagion dynamics with network metrics of complexity reduced graphs. Contagion dynamics are expressed by the percentage of infected nodes $d_C$. Network metrics include the size of the giant component $G$, the number of components $n$, the mean size of the components $\mu$, and the standard deviation of the component sizes $\sigma$.

Table 1: Correlation table for **A**

| | $d_{c=0}$ | $d_{c=1}$ | $d_{c=2}$ |
|---|---|---|---|
| $p$ | -0.81 | -0.84 | -0.53 |
| $G$ | 1.00 | 0.46 | 0.23 |
| $n$ | -0.92 | -0.40 | -0.20 |
| $\mu$ | 0.59 | 0.92 | 0.49 |
| $\sigma$ | 0.51 | -0.39 | -0.28 |

Table 2: Correlation table for $\mathbf{A}^{C=1}$

| | $d_{c=0}$ | $d_{c=1}$ | $d_{c=2}$ |
|---|---|---|---|
| $p$ | -0.81 | -0.84 | -0.53 |
| $G$ | 0.58 | 0.90 | 0.46 |
| $n$ | -0.82 | -0.77 | -0.39 |
| $\mu$ | 0.29 | 0.73 | 0.93 |
| $\sigma$ | 0.42 | 0.82 | 0.03 |

Table 3: Correlation table for $\mathbf{A}^{C=2}$

| | $d_{c=0}$ | $d_{c=1}$ | $d_{c=2}$ |
|---|---|---|---|
| $p$ | -0.81 | -0.84 | -0.53 |
| $G$ | 0.45 | 1.00 | 0.59 |
| $n$ | -0.65 | -0.92 | -0.50 |
| $\mu$ | 0.18 | 0.44 | 0.83 |
| $\sigma$ | 0.35 | 0.83 | 0.77 |

This does not hold for complex contagions which is due to the fact that a complex graph reduction does not consider the spreading direction of a contagion process. It is symmetric in the sense that it assumes that a contagion could potentially start and progress from any part of the network. A complexity reduced network may therefore not result in a complete overlap with a complex contagion dynamic. This small discrepancy between a symmetric complexity reduced network and a directed complex contagion process emphasizes the intricacies of dealing with complex contagion. We would also like to point out that the giant component $G$ always outperforms the correlation with control parameter $p$.

Additionally, the standard deviation of component sizes $\sigma$ exhibits intriguing behavior in that all contagion complexities nearly have the same maximum standard deviation over all $p$. The dropping parts after the peaks, in other words the relaxation part, have very similar form even if the growing side on the left side of the peaks shows a substantial change in shape throughout the different complexities.

Note that the imposed restrictions of a complex contagion with complexity $C = 3$ are too strict to facilitate any spreading given the initial seed of infected nodes (e.g. as shown in Fig. 3). In this case, a meaningful correlation analysis is not possible and corresponding correlation values are omitted.

## 4 Summary and Outlook

This paper presents a new method that is able to reduce a given network to only connections relevant for a contagion process of certain complexity. The method is applied to increasingly sparse Moore neighborhood networks and network metrics calculated on the network reductions are compared to complex contagion simulations. Results show that metrics, such as the giant component, the mean component size, and the standard deviation of component sizes, calculated on the reduced network correlate well with the complex spreading phenomenon on the original network.

For future work, it would be interesting to apply the complex graph reduction method to other network types, possibly real network structures, where scale-free degree distributions, small worlds, and preferential attachment dictate the path of a contagion. In this regard, first tests on real world sample social networks have already been performed and show promising results.

Additionally, transferring the non-linear operator to Tensor operations may provide a significant gain in calculation performance. Another point to look at is whether the idea of using network reducing operators on the adjacency matrix is applicable to other contagion processes as well as global network metrics.

## Acknowledgments

The authors thank the reviewers for their thoughtful comments which greatly improved this manuscript.

## References

[Bar16] Albert-László Barabási. *Network science.* Cambridge University Press, 2016.

[BHG+09] Duygu Balcan, Hao Hu, Bruno Goncalves, Paolo Bajardi, Chiara Poletto, Jose J Ramasco, Daniela Paolotti, Nicola Perra, Michele Tizzoni, Wouter Van den Broeck, et al. Seasonal transmission potential and activity peaks of the new influenza a (h1n1): a monte carlo likelihood analysis based on human mobility. *BMC medicine*, 7:1–12, 2009.

[CEM07] Damon Centola, Víctor M Eguíluz, and Michael W Macy. Cascade dynamics of complex propagation. *Physica A: Statistical Mechanics and its Applications*, 374(1):449–456, 2007.

[Cen21] Damon Centola. *Change: How to make big things happen.* Hachette UK, 2021.

[CF07] Nicholas A Christakis and James H Fowler. The spread of obesity in a large social network over 32 years. *New England journal of medicine*, 357(4):370–379, 2007.

[CM07] Damon Centola and Michael Macy. Complex contagions and the weakness of long ties. *American journal of Sociology*, 113(3):702–734, 2007.

[CWW+03] Elizabeth L Corbett, Catherine J Watt, Neff Walker, Dermot Maher, Brian G Williams, Mario C Raviglione, and Christopher Dye. The growing burden of tuberculosis: global trends and interactions with the hiv epidemic. *Archives of internal medicine*, 163(9):1009–1021, 2003.

[Dav02] Brian Davies. *Integral transforms and their applications*, volume 41. Springer Science & Business Media, 2002.

[DC18] Hugo Duminil-Copin. Sixty years of percolation. In *Proceedings of the International Congress of Mathematicians: Rio de Janeiro 2018*, pages 2829–2856. World Scientific, 2018.

[GC21] Douglas Guilbeault and Damon Centola. Topological measures for identifying and predicting the spread of complex contagions. *Nature communications*, 12(1):4430, 2021.

[HBG04] Lars Hufnagel, Dirk Brockmann, and Theo Geisel. Forecast and control of epidemics in a globalized world. *Proceedings of the national academy of sciences*, 101(42):15124–15129, 2004.

[IPBL19] Iacopo Iacopini, Giovanni Petri, Alain Barrat, and Vito Latora. Simplicial models of social contagion. *Nature communications*, 10(1):2485, 2019.

[MSFL17] Bjarke Mønsted, Piotr Sapieżyński, Emilio Ferrara, and Sune Lehmann. Evidence of complex contagion of information in social media: An experiment using twitter bots. *PloS one*, 12(9):e0184148, 2017.

[Pop08] Christopher Pope. Geometry and group theory. *Publisher: Texas A and M University*, 2008.

[SA15] Bogdan State and Lada Adamic. The diffusion of support in an online social movement: Evidence from the adoption of equal-sign profile pictures. In *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing*, pages 1741–1750, 2015.

[SBS18] Patrick Schober, Christa Boer, and Lothar A Schwarte. Correlation coefficients: appropriate use and interpretation. *Anesthesia & analgesia*, 126(5):1763–1768, 2018.

[SGRL11] Seungwon Shin, Guofei Gu, Narasimha Reddy, and Christopher P Lee. A large-scale empirical study of conficker. *IEEE Transactions on Information Forensics and Security*, 7(2):676–690, 2011.

[WGHB09] Pu Wang, Marta C González, César A Hidalgo, and Albert-László Barabási. Understanding the spreading patterns of mobile phone viruses. *Science*, 324(5930):1071–1076, 2009.

[ZC19] Jingwen Zhang and Damon Centola. Social networks and health: New developments in diffusion, online and offline. *Annual Review of Sociology*, 45:91–109, 2019.

[ZJ12] Yajin Zhou and Xuxian Jiang. Dissecting android malware: Characterization and evolution. In *2012 IEEE symposium on security and privacy*, pages 95–109. IEEE, 2012.

# Scientific Computing 2023

## Track: Mobilität

*Sabine Ascher, Mirjam Klammsteiner und Michael Kohlegger:* **Can you trust in me? A qualitative literature review on validation techniques for multi-sensory environments in autonomous driving** ........ 136

*F. Thilker, N. Paulitsch, A. Hofer und F. Giuliani:* **CFD based exploration of parametric 3D swirler** ........ 142

*Nico Kunz, Benjamin Pahl, Angelo Teichert, Wolfgang Oertel, Falk Schmidsberger, Robert Manthey, Alexander Kühn, Matthias Vodel, Christian Roschke und Marc Ritter:* **Entwicklung und Evaluation einer virtuellen Verkehrssimulation in Kombination mit einem Expertensystem** ........ 152

*Stefan Posch, Clemens Gößnitzer, Franz Rohrhofer, Bernhard C. Geiger und Andreas Wimmer:* **Finding the Optimum Design of Large Gas Engines Prechambers using CFD and Bayesian Optimization** ........ 160

*Johannes Pieber und Marlena V. Hofer:* **First development steps towards autonomous driving** ........ 170

*Georg Jäger:* **Using AI to transform Traffic Models into Mobility Models** .. 178

# Can you trust in me? A qualitative literature review on validation techniques for multi-sensory environments in autonomous driving

Sabine Ascher[1], Mirjam Klammsteiner[2], and Michael Kohlegger[1, 2]

[1]Data Science Unit, University of Applied Sciences Kufstein, Kufstein
[2]Josef Ressel Centre Vision2Move, University of Applied Sciences Kufstein, Kufstein

## Abstract

This paper provides a comprehensive literature review of data validation techniques for multi-sensor environments in the context of autonomously driving vehicles (AV). It analyzes existing methods for detecting anomalies and evaluates their applicability, with a focus on understanding the strengths and limitations of different approaches and their suitability for different types of data. The review is based on a thorough examination of 36 articles published after or in 2018, and uses Mayring's structuring content analysis technique to analyze the content and answer six key questions about the validation cases, types of data, models used, operation of the models, performance, and real-world applicability of each approach. The goal of the paper is to advance the field of anomaly detection and contribute to the safety and efficiency of AV by presenting an overview of data validation strategies.

## 1 Introduction

Autonomously driving vehicles (AVs) as a research topic has gained a lot of interest in recent years [TVS+22][OWJK21]. There are several research branches that are currently pushing forward in this regard, e.g., (*i*) AV technology [CZX+22], (*ii*) AV environments [PČY+16], or (*iii*) AV communication infrastructure [VBF+22]. While there is much bottom-up research on AVs, legislations have also prepared structures that regulate the use of AVs in specific countries [Bun21]. These new regulations pose additional requirements on the development, implementation, and use of AVs, e.g., the requirement to audit AV systems, or to guarantee compliance with certain regulations. Regarding the latter, the trustworthiness of sensor data is an important aspect that needs to be fulfilled to guarantee secure AV operation.

To achieve such trustworthiness in sensor data, the system must be able to detect anomalies reliably and quickly [FYP12]. By detecting unusual patterns and deviations from expected behaviors, autonomous vehicles can identify potential issues such as obstacles, malfunctions, or incorrect sensor readings. This helps prevent accidents, minimize downtime, and enhance overall system performance [HZC+22]. The sheer volume and velocity of data [CCS12] that is, however, produced by AVs and AV environments make it necessary to deploy working autonomous models that can assess data trustworthiness without human aid. Human operators would no longer be capable of manually screening (i.e., looking over) and assessing (i.e., deciding upon) all the produced data. Working autonomous models, in contrast, are scalable and can handle both large volumes and high velocities of data.

As our research shows, there are already many promising methods for detecting anomalies in data. This research area is characterized by approaches in the field of statistical models and machine learning algorithms. Literature dealing with the representation of the operating algorithms is widespread [SWP22]. In many cases, however, authors lack to elaborate on their overall approaches that the algorithms and their models are embedded into in detail. Therefore, it is essential to thoroughly evaluate these approaches and their potential for utilization in the area of AVs, e.g in terms of potential application areas, type of validation, usable input data, internal setup, as well as their produced outputs. This requires a comprehensive understanding of each approach. The ability to effectively compare different methods and evaluate their performance in various contexts is crucial to advance the field of anomaly detection in AVs and ensure the safety and efficiency of these systems. Therefore, this paper addresses the research question of how different approaches to data validation differ and for which use cases they are suitable. This will lead to a better understanding of anomaly detection methods, which will later help to improve the performance of AVs and increase their level of safety for passengers and other road users.

In this paper, we are presenting a literature review on data validation strategies. We have been doing research on already developed approaches in this field and have analyzed them according to various dimensions. We aim to find out how the approaches to data validation differ and for which types of data they are suitable for.

## 2 Methods

We have conducted a thorough literature review. The search process was based on the approach proposed by Webster and Watson [WW02]. We started with a keyword search on Google Scholar. There we looked for various combinations of the search terms sensor, validation, anomaly, monitoring and time series. We also used variations of these terms and combined them with search engine logic (e.g., “`[time series OR 'Time series'] AND [validation OR anomaly OR monitoring]`“). In addition to that, we have been using the references of the identified articles to conduct a complementing snowball search. This approach yielded a total number of 65 articles that were put to an initial evaluation using only their titles. To guarantee up-to-dateness we have only considered articles that were published after or in 2018 for further analysis, which yielded a corpus of 36 articles (see Figure 1).

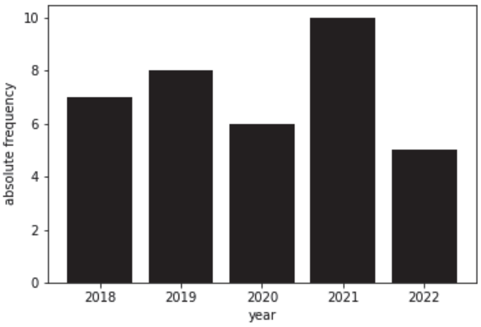


Figure 1: Number of publications per year

We have been using Mayring's structuring content analysis technique [May19], to further analyze the identified articles. We applied deductive coding (i.e., coding with a predefined set of codes) as well as inductive coding (i.e., coding with codes elicited from the material) to structure the articles' content. The coding was conducted by a team of three researchers and was guided by the following six questions: (*i*) Which validation case is supported? (*ii*) Which kind of data is supported? (*iii*) Which kind of approach has been used? (*iv*) How is the used model operated? (*v*) What is the definition of faulty data? (*vi*) What is the model output? These questions acted as initial codes (deductive coding) and were refined during the analysis process (inductive coding). We have boiled down the questions to (*i*), (*ii*), (*iii*) and (*vi*) as part of our analysis process. To ensure high inter-coder reliability, we have been conducting two meetings where the used codes as well as the intermediate findings have been discussed. Each meeting was followed by a re-coding session, where the taxonomy of codes - that Mayring calls category system - was refined and unified.

Table 1: Comparison of validation approaches

| Name | Sector | Validation Case | Supported Input | Machine Learning Strategies | Supported Output |
|---|---|---|---|---|---|
| **[SXJ+21]** | Self-driving vehicle monitoring | A | geo | Extended Kalman filtering with Gaussian distribution | Faulty score |
| **[CLH21]** | Wireless sensor networks | A | time series | Hypergrid based adaptive detection of faults method | Binary classification |
| **[FLS+18]** | Wireless sensor networks | A | time series | Very fast decision trees | Binary classification |
| **[SLS+18]** | Manufacturing | A, B | other | Classical machine learning methods | Quality rating |
| **[BPR+19]** | Cloud computing | C | other | Deep learning | Trust score |
| **[EMAN20]** | Building monitoring | C | time series | Auto-associative neural network | Binary classification |
| **[CGP+20]** | Research | C | json | Deep learning | Binary classification |
| **[DB19]** | Waste water monitoring | A | time series | Deep auto encoders combined with ARIMA and OCSVM | Binary classification |
| **[VWWKM19]** | Self-driving vehicle monitoring | C | time series | Deep learning combined with classical machine learning and Kalman filtering | Binary classification |
| **[KCC+22]** | Research | A | time series | Deep learning | Trust score |
| **[XWWL21]** | Research | A | time series | Anomaly transformer | Binary classification |
| **[SLK20]** | Research | A | time series | Dilated recurrent neural networks | Binary classification |
| **[ZWD+20]** | Research | A, B | time series | Deep learning | Trust score |
| **[MSDA18]** | Research | A | time series | Deep auto-encoder | Binary classification |
| **[KCD+21]** | Self-driving vehicle monitoring | B | geo | Deep learning | Trust score |
| **[BBL+19]** | High performance computing | A, B | other | Deep learning | Trust score |
| **[CBM+19]** | Vehicle movement analysis | C | trajectories | Deep learning | Binary classification |
| **[KMPM+20]** | Self-driving vehicle monitoring | B | time series | Internal cross-correlation parameters with dynamic Bayesian networks | Binary classification |
| **[WMK20]** | Vehicle movement analysis | B | geo | Model-based signal filtering | Binary classification |
| **[CAFG21]** | Research | A | time series | Deep learning | Trust score |
| **[LXX21]** | Research | A | time series | Self-supervised deep learning | Binary classification |
| **[Hoo19]** | Communication network monitoring | A | time series | Auto-regressive–moving-average model | Binary classification |
| **[SZZ+21]** | Research | A, B | time series | Hidden Markov model combined with 1D convolutional neural networks | Binary classification |
| **[WDL+22]** | Rail transit monitoring | C | time series | Long short term memory models | Binary classification |
| **[TPT+22]** | Manufacturing | A, B | time series | Deep learning | Trust score |
| **[LVM+19]** | Gas turbine monitoring | B | time series | Bayesian hierarchical models | Fault severity measurement |
| **[WGW+21]** | Research | B | time series | Gated recursive units and long shirt term memory models | Binary classification |
| **[HSY+22]** | Research | A | time series | Multiresolution self-supervised discriminative network | Binary classification |
| **[JRA+21]** | Research | C | time series | Deep learning | Trust score |
| **[JS22]** | Research | C | time series | Multi-layer perceptrons | Anomaly score |
| **[CZD19]** | Research | A | other | Progressive ensemble method using Isolation Forest and Local Outlier Factor | Binary classification |
| **[ZSM+18]** | Research | A | other | Deep auto-encoder combined with a gaussian mixture model | Binary classification |
| **[LIPJ21]** | Research | A, B | time series | Clustering algorithms | Binary classification |
| **[ZLK18]** | Research | B | other | Adaptive kernel density-based anomaly detection | Trust score |
| **[CMC18]** | Research | A | other | Deep learning based one-class neural network | Binary classification |
| **[CSHS18]** | System log monitoring | A | other | Deep convolutional auto-encoder | Binary classification |

# 3 Results

Our main findings are that most approaches to anomaly detection are currently experimental and have, at best, been tested with real-world data only. There is an ongoing debate on a unified definition of faulty data, which can be explained by the fact that the development of anomaly detection methods has been pushed forward by various different research domains. Our coding table (see Table 1) gives a good overview of these research domains, as can be seen in the *Sector* column. Many approaches have yet only been used in research scenarios. However, there are also applied fields that have great interest in anomaly detection (e.g., self-driving vehicle monitoring, or wireless sensor networks). Across these fields, we can see different approaches to anomaly detection, but there is a fairly stable set of algorithms that are used across all domains [SWP22].

We can see that current work supports three different *Validation Cases*, i.e., (*i*) the validation of a single stream of data by detecting outliers and anomalies (validation case A), (*ii*) the validation of a single stream of data by comparing it with a second data stream (validation case B) and (*iii*) the validation of a single stream of data by comparing it with a representative historic sample (validation case C). Most of the literature deals with validation case A, followed by validation case B and C. Articles that deal with validation case B are also likely to deal with case A.

As can be seen in the *Approach* column in Table 1, most recent works rely on deep learning models for anomaly detection. These models are very flexible with regard to their degrees of freedom and can be applied in a supervised as well as an unsupervised fashion (e.g., auto-encoders). The challenge with these models, however, is that their predictions are difficult to trace back to the input data. This is why they might not be suitable for applications where explainability is an important requirement. Still, classical machine learning approaches are only rarely used (e.g. LightGBM in outlier detection).

With regard to *Machine Learning Strategies*, we can see that both supervised as well as unsupervised approaches are covered by the analyzed articles. While most works rely on supervised learning, which requires labeled data that can be expensive to get, there are also approaches that work without labels. The latter approaches are especially promising if the data that is subject to anomaly detection has never been used before, wherefore there is not yet a ground truth of labels that could be used for training.

When looking at the *Supported Data* column, we can see that most works on anomaly detection focus on time series data, with some exceptions. Here, the second-most present group is the one focusing on anomaly (also often called outlier) detection with panel data. These data do not follow a random walk [Pea05] pattern, wherefore anomalies cannot be detected by looking at their time behavior but only by their exposure with respect to other data points.

Finally, we have examined how distinct approaches forecast anomalies. As can be seen in the *Supported Output* column, most papers use a classification approach with two classes (normal/abnormal). Some papers, however, rely on models that predict an anomaly score. These models treat their input data in a more fine-grained way, which can be beneficial. On the other hand, these models need an additional threshold value to make normative predictions.

# 4 Conclusion

In this paper, we have been reporting our findings from a comprehensive literature review of data validation techniques for multi-sensor environments. We have been analyzing 36 articles that have been published after or in 2018 and that were dealing with data validation strategies. By providing insights into the various methods for detecting anomalies in data and their specific requirements and constraints, the paper contributes to the development of reliable and effective methods for detecting anomalies. This research paper addresses the question of how different approaches to data validation differ and for which use cases they are suitable.

Our main findings are that (*i*) most state-of-the-art approaches rely on deep learning as it offers a high degree of freedom with regard to model performance but comes at the cost of low explainability; (*ii*) most approaches use binary classification to predict anomalies, rather than using a scoring system; (*iii*) most approaches have been evaluated with real-world data but are still in a research state; (*iv*) anomaly detection is mostly used with time series data and (*v*) anomaly detection can be done using an unsupervised as well as a supervised strategy.

This study has two main limitations. (*i*) Only literature published in 2018 or earlier is considered. However, the field of anomaly detection has been a research subject for more than five years. Approaches and algorithms that became public before 2018 may also be valid. This would also increase the current processed validation status of 36 papers, which would lead to a broader outcome. (*ii*) The findings were validated by three researchers. While conducting the study, classifying the validation cases led to disagreements. Further researchers as well as a re-coding could contribute to a better classification.

## 5 Acknowledgments

This research has partially been financed by the German Federal Ministry of Digital and Transport under the ALMODA project and by the Austrian Federal Ministry for Digital and Economic Affairs and the National Foundation for Research, Technology and Development, as well as the Christian Doppler Research Association under the Josef Ressel Center Vision2Move.

## References

[BBL+19] Andrea Borghesi, Andrea Bartolini, Michele Lombardi, Michela Milano, and Luca Benini. Anomaly detection using autoencoders in high performance computing systems. In *Proceedings of the AAAI Conference on artificial intelligence*, volume 33, pages 9428–9433, 2019.

[BPR+19] Eric Breck, Neoklis Polyzotis, Sudip Roy, Steven Whang, and Martin Zinkevich. Data validation for machine learning. In *MLSys*, 2019.

[Bun21] Deutscher Bundestag. Straßenverkehrsgesetz. STVG, 2021.

[CAFG21] Chris U Carmona, François-Xavier Aubet, Valentin Flunkert, and Jan Gasthaus. Neural contextual anomaly detection for time series. *arXiv preprint arXiv:2107.07702*, 2021.

[CBM+19] Damian Campo, Mohamad Baydoun, Pablo Marin, David Martin, Lucio Marcenaro, Arturo de la Escalera, and Carlo Regazzoni. Learning probabilistic awareness models for detecting abnormalities in vehicle motions. *IEEE Transactions on Intelligent Transportation Systems*, 21(3):1308–1320, 2019.

[CCS12] Hsinchun Chen, Roger HL Chiang, and Veda C Storey. Business intelligence and analytics: From big data to big impact. *MIS quarterly*, pages 1165–1188, 2012.

[CGP+20] Emily Caveness, Paul Suganthan GC, Zhuo Peng, Neoklis Polyzotis, Sudip Roy, and Martin Zinkevich. Tensorflow data validation: Data analysis and validation in continuous ml pipelines. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*, pages 2793–2796, 2020.

[CLH21] Lingqiang Chen, Guanghui Li, and Guangyan Huang. A hypergrid based adaptive learning method for detecting data faults in wireless sensor networks. *Information Sciences*, 553:49–65, 2021.

[CMC18] Raghavendra Chalapathy, Aditya Krishna Menon, and Sanjay Chawla. Anomaly detection using one-class neural networks. *arXiv preprint arXiv:1802.06360*, 2018.

[CSHS18] Yu Cui, Yiping Sun, Jinglu Hu, and Gehao Sheng. A convolutional auto-encoder method for anomaly detection on system logs. In *2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 3057–3062. IEEE, 2018.

[CZD19] Zhangyu Cheng, Chengming Zou, and Jianwei Dong. Outlier detection using isolation forest and local outlier factor. In *Proceedings of the conference on research in adaptive and convergent systems*, pages 161–168, 2019.

[CZX+22] Guangzhen Cui, Weili Zhang, Yanqiu Xiao, Lei Yao, and Zhanpeng Fang. Cooperative perception technology of autonomous driving in the internet of vehicles environment: A review. *Sensors*, 22(15):5535, 2022.

[DB19] A Disch and F Blumensaat. Messfehler oder prozessanomalie?–echtzeit-datenvalidierung für eine zuverlässige prozessüberwachung in kanalnetzen. *REGENWASSER WEITERDENKEN*, page 73, 2019.

[EMAN20] Mariam Elnour, Nader Meskin, and Mohammed Al-Naemi. Sensor data validation and fault diagnosis using auto-associative neural network for hvac systems. *Journal of Building Engineering*, 27:100935, 2020.

[FLS+18] Simon Fong, Jiaxue Li, Wei Song, Yifei Tian, Raymond K Wong, and Nilanjan Dey. Predicting unusual energy consumption events from smart home sensor network by data stream mining with misclassified recall. *Journal of Ambient Intelligence and Humanized Computing*, 9:1197–1221, 2018.

[FYP12] Jiajie Fan, Kam-Chuen Yung, and Michael Pecht. Anomaly detection for chromaticity shift of high power white led with mahalanobis distance approach. In *2012 14th International Conference on Electronic Materials and Packaging (EMAP)*, pages 1–5. IEEE, 2012.

[Hoo19] Bryan Hooi. Anomaly detection in graphs and time series: Algorithms and applications. *Published PHD Thesis*, 2019.

[HSY+22] Desen Huang, Lifeng Shen, Zhongzhong Yu, Zhenjing Zheng, Min Huang, and Qianli Ma. Efficient time series anomaly detection by multiresolution self-supervised discriminative network. *Neurocomputing*, 491:261–272, 2022.

[HZC+22] Xingshuo Han, Yuan Zhou, Kangjie Chen, Han Qiu, Meikang Qiu, Yang Liu, and Tianwei Zhang. Ads-lead: Lifelong anomaly detection in autonomous driving systems. *IEEE Transactions on Intelligent Transportation Systems*, 2022.

[JRA+21] Deokwoo Jung, Nandini Ramanan, Mehrnaz Amjadi, Sankeerth Rao Karingula, Jake Taylor, and Claudionor Nunes Coelho Jr. Time series anomaly detection with label-free model selection. *arXiv preprint arXiv:2106.07473*, 2021.

[JS22] Kyeong-Joong Jeong and Yong-Min Shin. Time-series anomaly detection with implicit neural representation. *arXiv preprint arXiv:2201.11950*, 2022.

[KCC+22] Siwon Kim, Kukjin Choi, Hyun-Soo Choi, Byunghan Lee, and Sungroh Yoon. Towards a rigorous evaluation of time-series anomaly detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 7194–7201, 2022.

[KCD+21] Gour Karmakar, Abdullahi Chowdhury, Rajkumar Das, Joarder Kamruzzaman, and Syed Islam. Assessing trust level of a driverless car using deep learning. *IEEE Transactions on Intelligent Transportation Systems*, 22(7):4457–4466, 2021.

[KMPM+20] Divya Kanapram, Pablo Marin-Plaza, Lucio Marcenaro, David Martin, Arturo de la Escalera, and Carlo Regazzoni. Self-awareness in intelligent vehicles: Experience based abnormality detection. In *Robot 2019: Fourth Iberian Robotics Conference: Advances in Robotics, Volume 1*, pages 216–228. Springer, 2020.

[LIPJ21] Jinbo Li, Hesam Izakian, Witold Pedrycz, and Iqbal Jamal. Clustering-based anomaly detection in multivariate time series data. *Applied Soft Computing*, 100:106919, 2021.

[LVM+19] Enzo Losi, Mauro Venturini, Lucrezia Manservigi, Giuseppe Fabio Ceschini, and Giovanni Bechini. Anomaly detection in gas turbine time series by means of bayesian hierarchical models. *Journal of Engineering for Gas Turbines and Power*, 141(11), 2019.

[LXX21] Minhao Liu, Zhijian Xu, and Qiang Xu. Deepfib: Self-imputation for time series anomaly detection. *arXiv preprint arXiv:2112.06247*, 2021.

[May19] Philipp Mayring. Qualitative content analysis: Demarcation, varieties, developments. In *Forum: Qualitative Social Research*, volume 20, pages 1–26. Freie Universität Berlin, 2019.

[MSDA18] Mohsin Munir, Shoaib Ahmed Siddiqui, Andreas Dengel, and Sheraz Ahmed. Deepant: A deep learning approach for unsupervised anomaly detection in time series. *Ieee Access*, 7:1991–2005, 2018.

[OWJK21] Daniel Omeiza, Helena Webb, Marina Jirotka, and Lars Kunze. Explanations in autonomous driving: A survey. *IEEE Transactions on Intelligent Transportation Systems*, 23(8):10142–10162, 2021.

[PČY+16] Brian Paden, Michal Čáp, Sze Zheng Yong, Dmitry Yershov, and Emilio Frazzoli. A survey of motion planning and control techniques for self-driving urban vehicles. *IEEE Transactions on intelligent vehicles*, 1(1):33–55, 2016.

[Pea05] Karl Pearson. The problem of the random walk. *Nature*, 72(1865):294–294, 1905.

[SLK20] Lifeng Shen, Zhuocong Li, and James Kwok. Timeseries anomaly detection using temporal hierarchical one-class network. *Advances in Neural Information Processing Systems*, 33:13016–13026, 2020.

[SLS+18] Sebastian Schelter, Dustin Lange, Philipp Schmidt, Meltem Celikel, Felix Biessmann, and Andreas Grafberger. Automating large-scale data quality verification. *Proceedings of the VLDB Endowment*, 11(12):1781–1794, 2018.

[SWP22] Sebastian Schmidl, Phillip Wenig, and Thorsten Papenbrock. Anomaly detection in time series: a comprehensive evaluation. *Proceedings of the VLDB Endowment*, 15(9):1779–1797, 2022.

[SXJ+21] Yanqing Shen, Chao Xia, Zhiqiang Jian, Shitao Chen, and Nanning Zheng. An integrated localization system with fault detection, isolation and recovery for autonomous vehicles. In *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, pages 84–91. IEEE, 2021.

[SZZ+21] Zijing Shang, Yingjun Zhang, Xiuguo Zhang, Yun Zhao, Zhiying Cao, and Xuejie Wang. Time series anomaly detection for kpis based on correlation analysis and hmm. *Applied Sciences*, 11(23):11353, 2021.

[TPT+22] Theodoros Tziolas, Konstantinos Papageorgiou, Theodosios Theodosiou, Elpiniki Papageorgiou, Theofilos Mastos, and Angelos Papadopoulos. Autoencoders for anomaly detection in an industrial multivariate time series dataset. *Engineering Proceedings*, 18(1):23, 2022.

[TVS+22] Behrad Toghi, Rodolfo Valiente, Dorsa Sadigh, Ramtin Pedarsani, and Yaser P Fallah. Social coordination and altruism in autonomous driving. *IEEE Transactions on Intelligent Transportation Systems*, 23(12):24791–24804, 2022.

[VBF+22] Ovidiu Vermesan, Roy Bahr, Mariano Falcitelli, Daniele Brevi, Ilaria Bosi, Anton Dekusar, Alexander Velizhev, Mahdi Ben Alaya, Carlotta Firmani, Jean-Francois Simeon, et al. Iot technologies for connected and automated driving applications. In *Internet of Things–The Call of the Edge*, pages 255–306. River Publishers, 2022.

[VWWKM19] Franco Van Wyk, Yiyang Wang, Anahita Khojandi, and Neda Masoud. Real-time sensor anomaly detection and identification in automated vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 21(3):1264–1276, 2019.

[WDL+22] Yujie Wang, Xin Du, Zhihui Lu, Qiang Duan, and Jie Wu. Improved lstm-based time-series anomaly detection in rail transit operation environments. *IEEE Transactions on Industrial Informatics*, 18(12):9027–9036, 2022.

[WGW+21] Yuhang Wu, Mengting Gu, Lan Wang, Yusan Lin, Fei Wang, and Hao Yang. Event2graph: Event-driven bipartite graph for multivariate time-series anomaly detection. *arXiv preprint arXiv:2108.06783*, 2021.

[WMK20] Yiyang Wang, Neda Masoud, and Anahita Khojandi. Real-time sensor anomaly detection and recovery in connected automated vehicle sensors. *IEEE transactions on intelligent transportation systems*, 22(3):1411–1421, 2020.

[WW02] Jane Webster and Richard T Watson. Analyzing the past to prepare for the future: Writing a literature review. *MIS quarterly*, pages xiii–xxiii, 2002.

[XWWL21] Jiehui Xu, Haixu Wu, Jianmin Wang, and Mingsheng Long. Anomaly transformer: Time series anomaly detection with association discrepancy. *arXiv preprint arXiv:2110.02642*, 2021.

[ZLK18] Liangwei Zhang, Jing Lin, and Ramin Karim. Adaptive kernel density-based anomaly detection for nonlinear systems. *Knowledge-Based Systems*, 139:50–63, 2018.

[ZSM+18] Bo Zong, Qi Song, Martin Renqiang Min, Wei Cheng, Cristian Lumezanu, Daeki Cho, and Haifeng Chen. Deep autoencoding gaussian mixture model for unsupervised anomaly detection. In *International conference on learning representations*, 2018.

[ZWD+20] Hang Zhao, Yujing Wang, Juanyong Duan, Congrui Huang, Defu Cao, Yunhai Tong, Bixiong Xu, Jing Bai, Jie Tong, and Qi Zhang. Multivariate time-series anomaly detection via graph attention network. In *2020 IEEE International Conference on Data Mining (ICDM)*, pages 841–850. IEEE, 2020.

# CFD based exploration of parametric 3D swirler

F. THILKER [1], N. PAULITSCH[2], A. HOFER[2], and F. GIULIANI[2]

[1]FH JOANNEUM – University of Applied Sciences, Institute Aviation/Luftfahrt, Eggenberger Allee 11, 8020 Graz, Austria

[2]COMBUSTION BAY ONE, Engineering, Research and Development, Ruckerlberggasse 13, 8010 Graz, Austria

## Abstract

The development of using sustainable fuels in the aviation section comes with big challenges. Not only by making the resources available but also by using them efficiently and safely. Therefore, different simulations and analysis of the combustion behavior need to be done, in order to operate these fuels safely and under specific conditions. One indicator, if technical changes need to be done on the given combustion design, is the Wobbe index. This index indicates the interchangeability of fuel gases, such as natural gas, and shows if two different gases can operate in the same system without making further technical changes. A conventional gas like $CH_4$ has a higher Wobbe index of 53.28, compared to $H_2$ with 48.23 [Too03]. Therefore, one important part of a premixed swirl-stabilized flame, is the swirler design which can improve the outcome and stability of the flame [KG11]. In this paper, the flow field of different premixing swirler geometries are discussed and improved by implementing an air swirl for stronger mixing of the flow, inspired by the design of the MOeBIUS project [10.22], and less sudden changes of the change of state. These results will be displayed, compared and approximated for the use of future fuels.

## 1 Introduction

In order to operate different fuels which come with different conditions, the geometries and boundary conditions of commercial combustion need to be modified. The geometry and properties have to be adapted to increased temperatures and flow rates. High expectations are being held on hydrogen changing commercial fuel for reaching a zero emission operation in terms of $CO_2$. In order to achieve efficient and safe operation, the flame and burning conditions need to be analyzed. Several studies [S.M02, Gup97] address swirl-stabilized flames and their flow fields, which can lead to vortex breakdowns or other unstable conditions of flame reactions due to pressure losses and negative axial velocities beneath the flame base tip. In the case of a premixing swirler, a geometry which turns a straight flow into a spinning/swirling flow, two fluids are mixed together as fast as possible by giving the fuel-air mixture a swirl [KG11, MASB09]. This avoids the flame having an eventual breakdown. On the other side, an overly increased swirl for the mixture can cause a rapid expansion at the entrance of the combustion, which leads to an adverse axial pressure gradient and a rapid vortex breakdown/flashback [LNO01]. In this study, three geometries of a swirler are designed and analyzed, in order to achieve a more stable combustion by changing the amount of swirl flow. This will result in different temperature, velocity and pressure distribution which can indicate the amount of stability for these flows.

# 2 Boundary conditions

In order to simulate and compare the created geometries, standard boundary conditions such as velocity, wall thickness and temperature conditions are used. In addition to that, a cylindrical combustion form is specified by the premixer's inlet diameter with a ratio of 1:4. The simulations are done with a pure air flow for the swirler and cylinder inlet. The inlet temperatures have a difference of 500K (Tab. 2) in order to have a clearer visibility of the temperature distribution along the axial flow and also to display experimental examples of a premixer like in the review from [HY09, Lei78]. For the velocity of the inlet, the $\Delta$V is based on practical values which are common for such a dimension of combustion.

In the simulation done by AnsysFluent2022 R2, the Standard k-epsilon turbulence model is taken for the viscous model which is also the most commonly used model in computational fluid dynamics (CFD) to simulate mean flow characteristics for turbulent flow conditions. The mesh size has been adapted to the complexity of the geometry [Hua03]. Since the Flower configuration shows the more complex design along the walls and the outlet of the swirler, the mesh grid has to be more accurate to ensure the most exact solutions without expanding the calculation time to much. Therefore, the number of cells, faces and nodes has been accommodated to the outlet of the swirler, which is the most important area for the analysis.

Table 1: Boundary conditions

| Properties | Inlet swirler | Ambient Air Geometry |
|---|---|---|
| Fluid | Air | Air |
| Temperature [K] | 800 | 300 |
| Velocity [m/s] | 9 | 1 |

With the mesh sizes defining the accuracy of the calculation, the node and element number are not chosen constantly over the geometry in order achieve a higher accuracy of the important section without high calculation times. The used meshing method is 'contact sizing', which meshes the contact region of bodies in a more detailed manner. This creates a higher accuracy in outflow and wall section. Therefore, the following properties are defined for each swirler and presented in table 2.

Table 2: Nodes and meshing elements for all swirler configurations

| | | Nodes | Elements |
|---|---|---|---|
| Reference Swirler | Ambient Air Geometry | 48098 | 276809 |
| | Swirler Geometry | 17824 | 16616 |
| 4-Star Swirler | Ambient Air Geometry | 36221 | 204611 |
| | Swirler Geometry | 10368 | 8959 |
| Flower Swirler | Ambient Air Geometry | 59532 | 340943 |
| | Swirler Geometry | 7800 | 6156 |

Since the method of the meshing is kept the same for all geometries (contact sizing), the setup is the same. Depending on the complexity of the swirler, more or less nodes and elements are used. For the ambient air geometry (cylinder), the amount of elements increases for the flower geometry, but reduces for the 4-Star geometry due to contact elements between the two bodies (swirler and cylinder).

## 2.1 Mathematics to Shape

The principle 'mathematics to shape' describes the idea of creating geometries by using mathematical functions in order to calculate the best shape for the problem. Therefore, analytical approaches are used in combination with new manufacturing procedures such as 3D printing in order to generate these complex shape solutions [BCBM21]. By starting with the improvement of the reference geometry to higher performance or adjustment to changing conditions, the wanted solutions need to be clarified. In case of this study, the new geometries need to generate more swirl by keeping the pressure difference as small as possible. Therefore, vanes to guide the axial flow need to be implemented with consideration for the pressure difference, which has a high impact on the swirler's performance. With these geometrical boundaries, the optimum shape can be generated not only with common empirical but also with analytical approaches, which can be used to make the ideal geometrical solution. Because of today's manufacturing procedures (3D printing, laser milling, etc.), these complex mathematical shapes are able to be produced and thus provide higher performance solutions [10.18].

## 2.2 Swirler Geometries

In order to analyze the impact of the premixer's geometry to the out-coming flow, three parametric configurations are created and compared. The geometry properties displayed in table 3 are chosen by a common size of combustion premixer such as [Lei78] and are kept equally. The standard configuration (see Fig. 1) is set as the standard and reference premixer for the other two geometries. The outlet of the swirler is 31% smaller than the inlet, which means a velocity rise is expected at the end of the outlet.

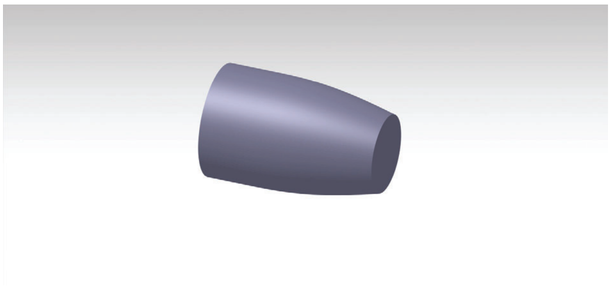


Figure 1: Reference geometry

Table 3: Swirler properties

| | Inlet Ø | Outlet Ø | Length |
|---|---|---|---|
| Standard | 26 mm | 18 mm | 40 mm |
| 4-Star | 26 mm | max. 26 mm | 26 mm |
| Flower | 26 mm | max. 26 mm | 26 mm |

An addition to that, table 4 shows the surfaces for the in- and outlet of the three configurations. For the 4-Star and Flower geometry, the outlet surface is reduced due to the rotation of the profile to generate the swirl. Overall, all surface ratios of all configurations are kept close together in order to have a real comparison on the performance.

Table 4: Swirler surface properties

| | Inlet Surface [mm$^2$] | Outlet Surface [mm$^2$] |
|---|---|---|
| Standard | 5,3027e-004 | 2,5287e-004 |
| 4-Star | 5,3027e-004 | 2,4375e-004 |
| Flower | 5,3027e-004 | 2,5599e-004 |

The second configuration is called '4-Star' (Fig. 2) because of the 90° rotation from a cylinder to a 4-Star shape. The length of the swirler is kept at 40 mm. The outlet has a maximum diameter of 26 mm and is kept equally with the inlet diameter in order to reach the same area as the inlet. This helps to reduce the pressure difference and therefore, more stability of the outlet flow.

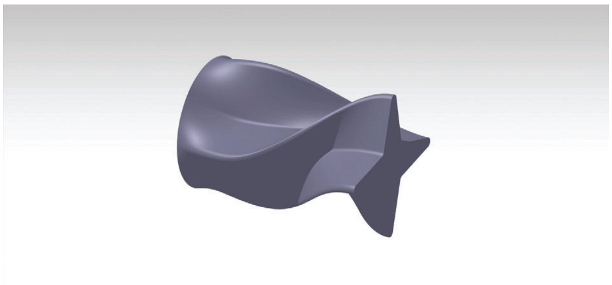


Figure 2: Swirler geometry in 4-Star form with progressive twist

The last configuration 'Flower' (Fig. 3) continues with the idea of a four prong outlet but adds a bigger outlet area and a 180° twist. Therefore the area of the flow, which produces no swirl, gets decreased. This area is the axial flow area which is not guided by the outside swirl and can directly flow from the inlet to the outlet of the geometry. In order to reduce this axial area, the four stars are designed more like a leaf shape and turned by 180°. This leaf shape, keeps the amount of prongs but increases the mass flow.

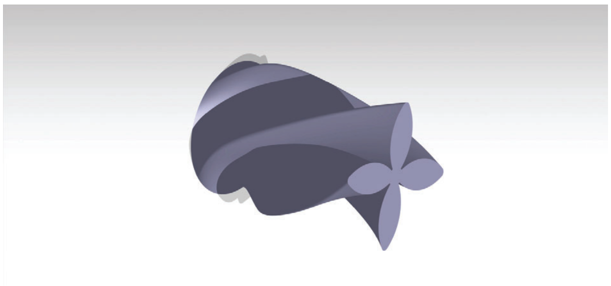


Figure 3: Swirler geometry in flower form with constant twist

## 3 Results and Discussion

The simulation results are generated by Ansys Fluent. All solutions and values are based on the exported solution data from the simulation itself. Therefore, the analytical calculations such as the pressure difference, k-factor and swirl number are done with the simulated data.

### 3.1 Velocity Magnitude

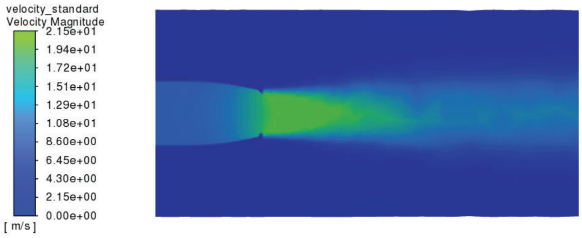


Figure 4: Velocity magnitude for the cone premixer geometry

The inlet-velocity of all simulations is kept constant with 9 m/s. When changing geometry parameters the velocity also changes due to smaller cross-sections or complex swirler contours. This means that the outlet velocity is different between the three configurations. For the reference geometry (Fig. 4), the inlet velocity rises up to 21.5 m/s at the swirler outlet and reduces down to 10.8 m/s by the length of two times the inlet diameter. In the inner section of the premixer, no swirl is generated, which means the reduced outlet diameter causes the velocity rise. Therefore, a pure axial flow jet occurs [S.M02]. In comparison to that, you see a much higher velocity rise in the '4-Star' configuration (Fig. 5), due to the 4-star shaped prongs and the reduction of the inner cross-section at the half of the premixers length. Here, the flow accelerates up to a maximum of 53.5 m/s. Only close to the outlet, the velocity slows down to 35.2 m/s which is caused by the sudden diameter expansion. This also leads to a reduction of the velocity cone length after the outlet to 1.5 times the premixers inlet diameter. Therefore, the axial flow, which is not guided by the vanes, causes a strong rise of the velocity magnitude. To combine both of the advantages of the two previous geometries, the 'Flower' configuration has a reduced mid axial flow section (Fig. 6) which means that almost all of the fluid flows through the four flower vanes and causes a higher swirl. By that, the maximum velocity is reduced to 30.5 m/s which is almost a 40% reduction compared to the 4-Star geometry. In addition to that, the velocity cone after the outlet is shorter and adapts better to the outer flow, which also means a sudden velocity change is avoided.

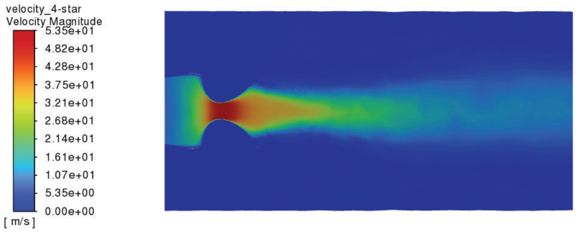


Figure 5: Velocity magnitude for the 4-Star premixer geometry

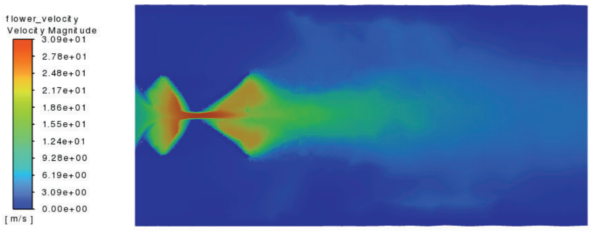


Figure 6: Velocity magnitude for the flower premixer geometry

Table 5: Swirl number calculation parameters

| | |
|---|---|
| Tangential Velocity [m/s] | $V_\theta$ |
| Radial Velocity [m/s] | r |
| Axial Velocity [m/s] | u |
| Density [kg/m$^3$] | $\rho$ |
| Maximum Inlet radius [m] | $r_{max}$ |

## 3.2 Swirling Gradient

In the development of a swirler, different flow characteristics are included. Y. Huang and V. Yang [HY09] describe one of the most common and important ones, the vortex breakdown. This phenomenon is an abrupt change in the core of the slender vortex and can develop towards the downstream in a so called 'bubble' or a spiral pattern. Leibovic [Lei78] and others [Gup97, LNO01] have reviewed most of the breakdowns and observed, in this case, the most interesting one for low swirl numbers. With small swirl numbers, which are defined as the ratio of the axial flux of angular momentum to the axial flux of linear momentum [F.G22], usually the spiral mode occurs. In this case, the approximation for the created geometries is within a swirl number below 0.6 wherefore, a vortex breakdown is expected.

For the analytical comparison of the swirler, the swirl number is defined for each configuration. The used integral in polar-form is defined as [F.G22]:

$$S = \frac{2\int\limits_{0}^{2\pi}\int\limits_{a}^{r_{max}} \rho u V_\theta r^2 dr d\theta}{D\int\limits_{0}^{2\pi}\int\limits_{a}^{r_{max}} \rho u^2 r dr d\theta} \tag{1}$$

This number describes the characteristics of the swirl created by the geometries. A swirl number $< 0.6$ means, the swirl can be interpreted as axial flow jet. Lucca-Negro and O'Doherty [LNO01] have noted, that the recirculating flow emerges in the swirling annular flow when the swirl number is higher than 0.6, causing an adverse flow direction, whereas a number of 0.6 or higher is seen as a tangential flow jet. For the solution of the swirl numbers, the calculation plane is about 2*D from the swirler outlet, whereas D represents the swirler inlet diameter. In this case it is 80 mm from the swirler inlet. Since the swirler inlet diameter does not change for the geometries, it is always at the same position. With defining the section, the simulated values of the x, y and z velocity with the related coordination's are exported and converted into a polar coordinates in order to use the Swirl Number formula (1). If the tangential velocity vectors face clockwise, the result of an negative swirl number will appear due to the rotation direction of the polar angle. By evaluating the vectors of the velocity in x,y and z direction the mesh size and boundary conditions are an important factor in the accuracy of the results. For the standard geometry, a swirl number of 0 is expected, since it produces no swirl of the fluid and the tangential velocity is relatively small compared to the 'Flower' geometry. In the case of the 4-Star, the swirl number rises up to 0.0676, which is caused by the 90° turn of the star outlet. This number is still significantly smaller compared to the 'Flower' configuration, which is caused by the inner section of the swirler. This cylindrical section is still causing a stream without swirl, which prevents the fluid to mix faster with the outer flow. In the 'Flower' configuration you can see in Fig. 6, the cylindrical mid section, which causes no swirl is highly reduced, which therefore causes a higher tangential velocity. After the revision of the meaning of the small numbers (Tab. 6), the value of 0.336 is still below the named 0.6, which means there is still not much adverse flow direc-

Table 6: Swirl number results

| | |
|---|---|
| Standard | 0 |
| 4-Star | 0.068 |
| Flower | 0.336 |

tion. In combination with the dynamic pressure at the inlet of the swirler, the flower geometry shows a high swirl with low pressure difference to the ambient pressure. This also means that this configuration is not a ful jet flow type anymore, compared to the other ones. In order to achieve an even higher swirl than the Flower configuration, the inner cross-section needs to be even more reduced and the prong number and area increased, so the fluid can be completely guided by the swirl. This needs to be combined with a small pressure difference, which rises when to much swirl is applied.

### 3.3 Pressure Distribution

The pressure distribution of a premixing swirler is one indicator of the stability of the flame. That means by sudden pressure changes, breakdowns or instabilities of the flame shape or position can occur.

Therefore, the simulation shows the pressure distribution for the three geometries along the flow and indicates how the swirl can influence that.

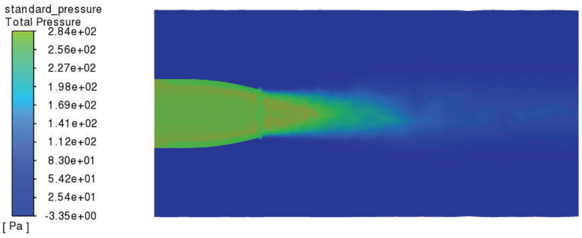


Figure 7: Pressure diagram for the standard premixer geometry

For the reference geometry, small pressure distributions are expected due the reference geometry. This can be seen by simulation results in Fig. 7. With the velocity being inversely proportional to the pressure, the pressure rises towards the smaller outlet and drops within the velocity cone outside of the swirler. For the 4-Star configuration, a much higher pressure change is shown in Fig. 8. With the geometry changing suddenly after the inlet, the velocity drops and the pressure rises. This leads to an outlet pressure of 650 Pa, which decreases to 300 Pa at the end of the pressure cone. That indicates that the low amount of swirl and the diameter changes within the geometry cause a sudden pressure rise and therefore, the possibility of an instability of the flow shape. Therefore, sudden area changes need to be avoided in order to combine an accurate swirl and a small pressure difference. The Flower configuration already shows some of these modifications based on the Fig. 9. The incoming pressure of around 300 Pa stays almost constant throughout the boundaries of the geometry, until it drops to 75 Pa at the end of the pressure cone. This shows that by increasing the vanes area, less barriers lose velocity and more area produces enough swirl.

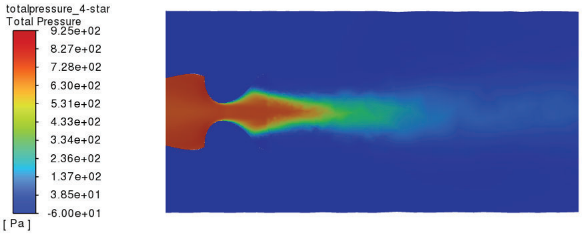


Figure 8: Pressure diagram for the 4-Star premixer geometry

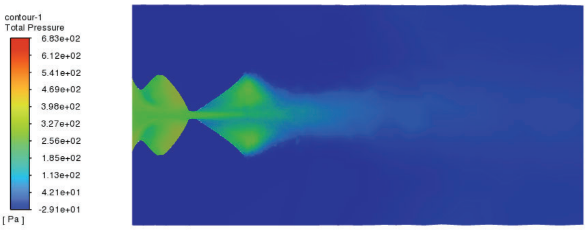


Figure 9: Pressure diagram for the flower premixer geometry

That also indicates, that it is possible to generate an stable combustion, combined with a fast mixing flow, which can be applied to further analysis with different fuel compounds.

### 3.4 Calculated Singular Pressure Loss

As described, the pressure drop caused by the swirler is also an indicator of efficiency and performance. The velocity inlet with 9 m/s is kept equal to all geometries. Since the pressure loss is not linear along the axial core because there is no wall friction, like in standard pipe calculations, the assumption of a proportional dynamic pressure for valves and nozzles calculation is made. That means, the pressure difference is defined by the dynamic pressure from the inlet of the swirler to the ambient conditions (300 K, 101325 Pa).

The exported data from the simulations show the total and static pressure of the inlet. With the array values from the simulation, the mean total and static pressure is calculated (Tab. 7). By computing the dynamic pressure with,

$$p_{dyn} = p_{total} - p_{static} \tag{2}$$

the first approximation of the performance regarding the pressure stability can be made. Therefore, $\mathrm{p_{dyn}}$ represents the dynamic pressure, $\mathrm{p_{stat}}$ the static pressure and $\mathrm{p_{total}}$ the total pressure of the flow.

The reference geometry shows the less pressure difference of all swirlers. This fits into the expectation regarding the former simulation results and experimental reviews as [MASB09]. For the recirculation zones, which can occur for swirl numbers smaller than 0.6, significant radial pressure gradients appear at any axial position due to centrifugal effects. But because there is no coupling between axial and tangential velocity components at lower swirl numbers, the axial pressure gradient does not rise significantly and is therefore not strong enough for axial recirculation. [Gup97]

Table 7: Mean pressure from the exported data and the calculated dynamic pressure

| | $\mathrm{p_{stat.}}$ [Pa] | $\mathrm{p_{dyn.}}$ [Pa] | $\mathrm{p_{total}}$ [Pa] |
|---|---|---|---|
| Standard | 232.738 | 49.583 | 282.320 |
| 4-Star | 731.424 | 49.583 | 786.172 |
| Flower | 389.275 | 59.749 | 449.023 |

In order to evaluate the pressure difference the K-value is calculated (Tab. 8). This value displays the ratio between the static and dynamic pressure of the geometry. Based on the fact that no swirl is applied, the reference geometry should have the least amount of pressure difference. For the 4-Star and Flower configuration higher K ratios are expected due to the swirl and changing area of the outlet. With the exported simulation data from all configurations, the K-value has been defined and confirms the assumption. The lowest pressure difference is shown for the reference geometry. As in former results of the temperature and velocity distribution, the performance of the 4-Star swirler is, also in this case, the worst of all three configurations. The K-value is therefore more than three times higher than the reference geometry, which indicates that the flow given by the vanes causes a high increase in pressure in the core. As for the Flower configuration, an unexpectedly low value is shown. That means, that the high amount of swirl still causes a low pressure difference. Therefore, by increasing the size of the outlet area and reducing the area where no swirl is applied, the pressure difference decreases. Therefore, the singular pressure loss depends o the geometrical shape of the swirler, which has been designed by CAD. In addition, there is no big compressor needed in order to generate these pressures.

Table 8: K values

| | K |
|---|---|
| Standard | 4.694 |
| 4-Star | 13.359 |
| Flower | 6.515 |

### 3.5 Temperature Distribution

The Temperature distribution already gives the first impression of the swirler's performance. After reviews done by Y. Huang and V. Yang [HY09], higher swirl numbers cause a faster temperature drop after the outlet, which is due to the mixing process of the hot and cold fluids. For the standard configuration, where the swirler creates no swirl, the hot fluid does not mix as fast and creates a long, cone-shaped temperature zone, whereas experimental observations done by [KG11, LNO01] show swirlers with higher swirl numbers, create a shorter but wider cone directly after fluid leaves the outlet of the swirler. This fits to the displayed solutions in Fig. 4 for the reference standard swirler. Figure 10 shows more temperature distribution compared to the standard swirler geometry. The maximum fluid temperature of 800 K decreases earlier in axial direction which means the hot fluid mixes earlier with the cold air because of its swirl and the reduction of the axial velocity. This means the geometry does not only mixes and reduces the axial velocity, but also has a higher impact on the pressure loss. For the 4-Star configuration, the temperature cone at the end of the swirlers outlet is shorter, which leads to a temperature drop, earlier in axial direction (Fig. 11). This is caused by the velocity drop and a faster mixing of the two fluids. In addition to that, the temperature drops down to close to 450 K at the end of the cylinder, which, compared to the reference configuration, is almost 100 K less. This still does not lead to a efficient swirler and that is why more fluid needs to be guided by the swirl vanes.

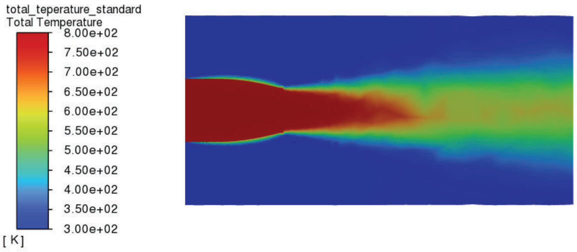


Figure 10: Temperature diagram for the cone premixer geometry

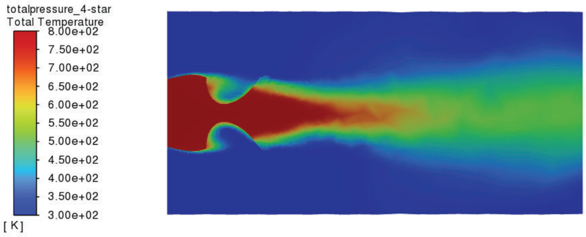


Figure 11: Temperature diagram for the 4-Star premixer geometry

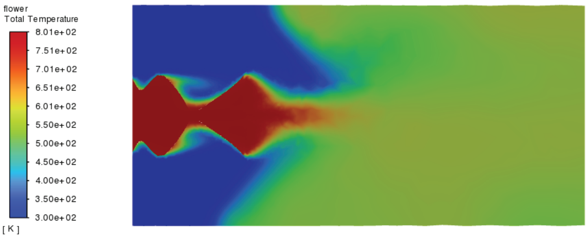


Figure 12: Temperature diagram for the flower premixer geometry

With the Flower geometry, the prong area has been increased which causes a more swirled flow and therefore, more temperature distribution. In figure 12, the temperature section shows the distribution of the temperature and therefore, the effectiveness of the swirler itself. More fluid is guided by the geometry and less flow enters the swirl cross-section without following the 180° twist. Therefore, the temperature drops down to 550 K after a length of 1*D behind the outlet. This already describes the good mixing process of the swirled hot with the cold bypass flow.

## 4 Summary and Outlook

Overall this study shows, which modifications for swirler shapes are effective to generate a stable and safe premixing flow. The CFD simulation allows to generate first impressions and results, which can help to make better technical decisions. Changing the geometry in order to achieve a faster mixing of the fluids can result in higher temperatures, velocity distributions and pressure differences. The swirl can cause the temperature to drop closer to the swirlers outlet and creates a shorter and wider temperature cone right after the flow leaves the swirler [Gup97]. The numerical solutions of the swirl number and the k-value indicate, among others, the effectiveness of the three configurations. The two created geometries, 4-Star and flower, both show unexpected results when it comes to the temperature and pressure distribution. The 4-Star shows the highest pressure difference with a minimum amount of swirl. Compared to that, the Flower geometry shows a good performance when it comes to the combination of a high swirl number combined with a low pressure difference, and excellent mixing of the fluids. By looking at future fuel-air mixtures such as hydrogen-air combustion, results from this simulation can be translated and helpful to understand the flow behavior of swirler. Of course different viscous equations need to be applied in order to have the correct mixing boundaries for these two fluids, but it shows the most important weighting factors.

## Acknowledgments

The author is grateful to the institute of Aviation, FH Joanneum Graz for valuable contributions and the opportunity of this paper. The single-surface swirling premixer is a detail of the MOeBIUS technology (patent AT523924B1; PCT/AT2021/060270) which effectively forces the interaction between fresh reactants and burnt gases prior to combustion, taking advantage of the flow's tangential momentum, hereby considered with an approach of type mathematics to shape. The direct application of this work will be evaluated experimentally in the crystAIr project, which uses similar technology combined with additive manufacturing. We thank the FFG, the Austrian Research Promotion Agency, for funding the MOeBIUS (call 2019, contract 881041) and crystAIr (call 2021, contract 894105) projects within the framework of the aeronautics Take-Off program.

## References

[10.18] *An Assessment on the Benefits of Additive Manufacturing Regarding New Swirler Geometries for Gas Turbine Burners*, volume Volume 4A: Combustion, Fuels, and Emissions of *Turbo Expo: Power for Land, Sea, and Air*, 06 2018. V04AT04A008.

[10.22] *Flow Design Using CFD for a Constant-Section Recursive Sequential Combustor*, volume Volume 10C: Turbomachinery — Design Methods and CFD Modeling for Turbomachinery; Ducts, Noise, and Component Interactions of *Turbo Expo: Power for Land, Sea, and Air*, 06 2022. V10CT32A029.

[BCBM21] Manik Chandra Biswas, Samit Chakraborty, Abhishek Bhattacharjee, and Zaheeruddin Mohammed. 4d printing of shape memory materials for textiles: Mechanism, mathematical modeling, and challenges. *Advanced Functional Materials*, 31(19):2100257, 2021.

[F.G22] F.Giuliani. Gas turbine combustion - technology, dimensioning diagnostics. *Combustion Bay One e.U. advanced combustion management Schuetzenhofgasse 22, A 8010 Gra*, 2022.

[Gup97] A.K. Gupta. Gas turbine combustion: Prospects and challenges. *Energy Conversion and Management*, 38(10):1311–1318, 1997. International Symposium on Advance Energy Conversion Systems and Related Technologies.

[Hua03] Weicheng Huang. Dynamical computing power balancing for adaptive mesh refinement applications. In K. Matsuno, A. Ecer, N. Satofuka,

J. Periaux, and P. Fox, editors, *Parallel Computational Fluid Dynamics 2002*, pages 411–418. North-Holland, Amsterdam, 2003.

[HY09] Ying Huang and Vigor Yang. Dynamics and stability of lean-premixed swirl-stabilized combustion. *Progress in Energy and Combustion Science*, 35(4):293–364, 2009.

[KG11] Ahmed E.E. Khalil and Ashwani K. Gupta. Distributed swirl combustion for gas turbine application. *Applied Energy*, 88(12):4898–4907, 2011.

[Lei78] S Leibovich. The structure of vortex breakdown. *Annual Review of Fluid Mechanics*, 10(1):221–246, 1978.

[LNO01] O Lucca-Negro and T O'Doherty. Vortex breakdown: a review. *Progress in Energy and Combustion Science*, 27(4):431–481, 2001.

[MASB09] Seoksu Moon, Essam Abo-Serie, and Choongsik Bae. Air flow and pressure inside a pressure-swirl spray and their effects on spray development. *Experimental Thermal and Fluid Science*, 33(2):222–231, 2009.

[S.M02] S.Menon. Simulation of combustion dynamics in gas turbine engines. *Parallel Computational Fluid Dynamics*, 2002.

[Too03] Engineering ToolBox. Fuel gases - wobbe index - technology, dimensioning diagnostics. 2003.

# Entwicklung und Evaluation einer virtuellen Verkehrssimulation in Kombination mit einem Expertensystem

Nico Kunz[1], Benjamin Pahl[1], Angelo Teichert[1], Wolfgang Oertel[2], Falk Schmidsberger[1], Robert Manthey[1], Alexander Kühn[1], Matthias Vodel[1], Christian Roschke[1], und Marc Ritter[1]

[1]Hochschule Mittweida - University of Applied Sciences, Fakultät Angewandte Computer- und Biowissenschaften, Technikumpl. 17, 09648 Mittweida, Deutschland
[2]Hochschule für Technik und Wirtschaft Dresden - University of Applied Sciences, Fakultät Informatik/Mathematik, Friedrich-List-Platz 1, 01069 Dresden, Deutschland

## Zusammenfassung

Dieser Artikel beschreibt, wie mithilfe einer Spiele-Engine und eines Expertensystem-Tools eine Software aufgebaut wird, die zur Simulation und Analyse von Straßenverkehrssituationen verwendet werden kann. Dafür wird eine verkehrsrelevante dreidimensionale Infrastruktur einer Stadt simuliert, die wechselseitig über Funktionsaufrufe und Datenübergabe mit einem Expertensystem kommuniziert.

In einer Evaluation mit verschiedenen Probanden wurde die Funktionalität des Expertensystems und des Simulationssystems begutachtet. Diese ergab, dass der gewählte Ansatz hinsichtlich Systemarchitektur, Programmierung, raumzeitlicher und wissensbasierter Modellierung, Nutzerinterface sowie Flexibilität tragfähig ist. Gleichzeitig traten jedoch auch die Grenzen, was den Ressourcenbedarf und den Implementierungsaufwand betrifft, zutage.

## 1 Einleitung

Der Gedanke, Künstliche Intelligenz (KI) für die Überwachung von Verkehr einzusetzen, bietet einen guten Lösungsansatz, um Unfälle, Staus und kritische Situationen vorzubeugen. Heutzutage gibt es schon mehrere Wege, wie der Verkehr mithilfe künstlicher Intelligenz überwacht werden kann [MMJAG20]. Auch das autonome Fahren von Autos kann nur mit verschiedenen KI-Softwaresystemen funktionieren, wobei Softwarefehler verhindert werden, aber nicht auszuschließen sind. Da bereits ein einzelner Fehler in diesem Kontext zu erheblichen Folgen führen kann, ist es ratsam, Tests mit verschiedenen KI-Modellen durchzuführen, um herauszufinden, welches mit Anforderungen und auftretenden Problemen am besten umgehen kann. Dafür wird ein Blick auf Expertensysteme geworfen. Diese bilden das Wissen von Fachleuten ab und speichern dieses formalisiert in Regeln und Fakten, die daraufhin in Inferenzverfahren zur Lösung verschiedener komplexer Probleme in einem begrenzten Fachgebiet genutzt werden können. Dadurch könnten sie sich optimal eignen, um das präzise formulierte Wissen des Straßenverkehrs nachzubilden. In diesem Beitrag wird folglich getestet, wie gut ein Expertensystem dafür geeignet ist. Mithilfe der etablierten professionellen Software CLIPS[1] (C Language Integrated Production System) wurde ein Expertensystem erstellt und dieses mit Wissen zum Straßenverkehr gefüllt. Um Daten zum Testen zu erhalten, wurde sich dazu entschieden, den Verkehr zu simulieren, anstatt reale Verkehrsszenen zur Datengewinnung zu nutzen. Das erleichtert das Annotieren der Daten und dadurch auch die Überprüfung, ob ein Expertensystem für die Verkehrsüberwachung infrage kommen kann. Außer-

[1] https://clipsrules.net/

dem ist es möglich, gezielt kritische Situationen nachzubilden, die im realen Verkehr nur selten auftreten. Da die grafische Darstellung mithilfe von 3D-Modellen und die Möglichkeiten zur einfachen Erweiterung der Simulation ein wichtiger Grundstein der Software sind, wurde die für Spiele entworfenen Laufzeit- und Entwicklungsumgebung Unity[2] genutzt, um die 3D-Simulation des Straßenverkehrs zu erschaffen. Diese umfasst dabei verschiedenste Teilnehmer des Straßenverkehrs und auch weitere Verkehrsobjekte, sodass ein breites Stadtbild nachgebildet wird. Dadurch können Fakten über die Position der verschiedenen Verkehrsteilnehmer direkt aus der Simulation gewonnen werden. Das Expertensystem arbeitet mit diesen Informationen und kann mittels des in ihm gespeicherten Wissens Ergebnisse zu den vier Themenbereichen *Verhalten, Verstoß, Halten und Unfall* liefern. Ein Ergebnis ist hierbei eine Situation, welche das Expertensystem anhand der Fakten erkannt hat. Darunter kann zum Beispiel ein Unfall oder ein Rot-Licht-Verstoß fallen. Ein großer Vorteil ist hierbei die Flexibilität und Anpassungsfähigkeit, da der Benutzer selbst Verkehrsszenarien erzeugen und somit unterschiedlichste Szenarien simulieren und auswerten lassen kann. Zudem wird kein Vorwissen im Bereich KI und Expertensysteme vorausgesetzt.

Um die Effektivität und Richtigkeit des erstellten Systems zu bestimmen, wurde in einer Evaluation mit Studierenden, wissenschaftlichen Mitarbeitern und Professoren aus zwei verschiedenen Hochschulen getestet, ob die vom Expertensystem generierten Schlussfolgerungen mit denen der Probanden übereinstimmen. Anhand dieser Ergebnisse werden am Ende des Papers die Vor- und Nachteile zur Nutzung von Expertensystemen im Straßenverkehr diskutiert.

Der Rahmen dieser Forschung wurde durch das Modul „Wissenschaft und Wirtschaft“ der Hochschule Mittweida gelegt. In diesem werden Gruppen von Studierenden verschiedene weitreichende Aufgaben gestellt, die das Entwickeln eines Programmes beinhalten. Dieser Prozess wird von einer Vielzahl an Betreuern und Stakeholdern begleitet und somit professionell unterstützt.

## 2 Verwandte Arbeiten

Das System CLIPS wird für Expertensysteme in einer Vielzahl von Branchen bereits benutzt und ist in diesen anerkannt [TWK+16]. In der Medizin wird es z.B. bei der Diagnose verwendet. In der Forschung von Samhan et. al. wurde ein Expertensystem erstellt, welches Knieprobleme erkennen soll [SAAN21]. Es wurde eine Handyapp erschaffen, bei welcher die Nutzer Fragen mit *Ja* oder *Nein* beantworten. Aus diesen Antworten kann das Expertensystem selektieren, welches Problem das Knie haben könnte. Ein sehr einfaches Prinzip, welches jedoch gut funktioniert und dadurch eine grobe medizinische Prognose ermöglicht.

Ein anderes Gebiet, in dem Expertensysteme benutzt werden, ist die freie Wirtschaft, um für Touristen einen optimalen Urlaubsort unter Berücksichtigung zugehöriger Bedürfnisse aufzufinden [KS12].

Auch CLIPS wurde bereits in Verkehrsszenarien, speziell bei Autobahnen getestet, um Unfälle zu erkennen und dementsprechend zu handeln [CH93]. Expertensysteme arbeiten mithilfe von Künstlicher Intelligenz. In [KKB06] dient sie u.a. dazu, Unfälle zu erkennen und zu verhindern.

Ein Fall, bei dem eine andere Künstliche Intelligenz im Straßenverkehr verwendet wurde, stellt die Arbeit [NHN+22] dar. In dieser wurde eine KI erstellt, welche über einen festgelegten Zeitraum hinweg die Anzahl an Fahrzeugen oder Passanten zählt. Das generelle Thema der Verkehrsanalyse ist hierbei sehr ähnlich zu dem Vorhaben dieser Arbeit, jedoch unterscheidet es sich hauptsächlich im Bereich der Annotation. Während in dieser Arbeit Fakten aus einer 3D-Umgebung entnommen werden, wird in jenem Projekt ein Video annotiert.

Verkehrssimulationen sind ein wesentlicher Bestandteil, um Straßenverkehrssituationen zu analysieren. Beispielsweise kann CBLabs den Ver-

[2] https://unity.com

kehrsfluss eines Straßennetzes mit bis zu 10.000 Kreuzungen analysieren, und stellt dieses in einer zweidimensionalen Vogelperspektive dar [LHL+22]. Es gibt jedoch auch dreidimensionale Verkehrssimulationen, die sich im Schwerpunkt auf die Analyse des Verkehrsflusses konzentrieren und ebenfalls mit Unity arbeiten [SQSL22].

## 3 Systemarchitektur und Implementierung

**Systemarchitektur** Abbildung 1 zeigt einen Überblick der Systemarchitektur und stellt dar, wie die Simulation mit dem Expertensystem verbunden ist.

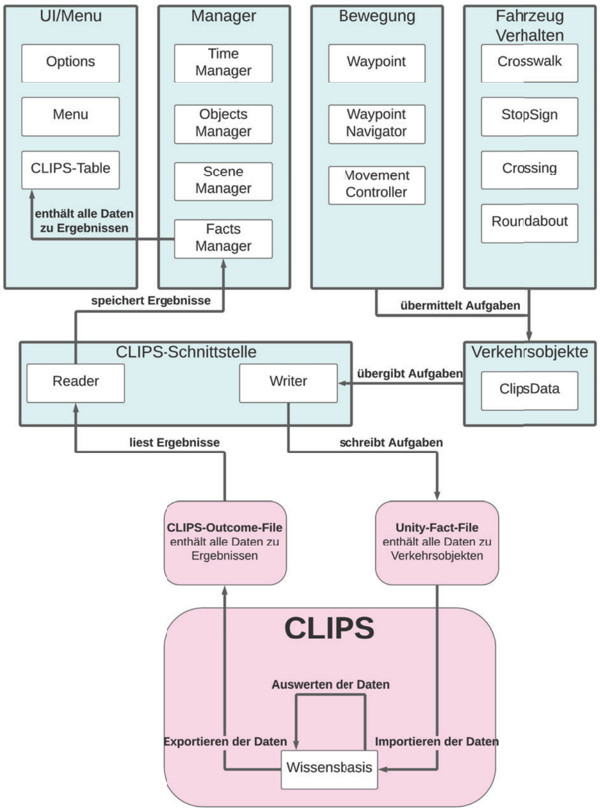


Abbildung 1: Systemarchitektur

**Software** Zum Erstellen des Projektes wurden mehrere verschiedene Programme genutzt. Die Spiele-Engine Unity wurde als Laufzeit- und Entwicklungsumgebung zur Simulation des Verkehrs gewählt, wobei die Programmiersprache C# mit den IDEs Visual Studio[3] und Jetbrain Rider[4] Einsatz fand. CLIPS wurde zum Erstellen und Testen des Expertensystems genutzt. Es wurden fast alle 3D-Modelle der Simulation mit den 3D-Modelling-Werkzeugen Blender[5] und Maya[6] selbstständig erstellt und dann mithilfe der Software Substaince Painter[7] texturiert. 2D-Assets wurden mit Adobe Photoshop[8] erzeugt. Zur gleichzeitigen und verteilten Nutzung fand Git[9] Verwendung.

**Unity** Die Unity Engine dient als Entwicklungsumgebung, um alles visuelle darstellen zu können. Wie in Abb. 2 zu sehen, ist ein wesentlicher Bestandteil die 3D-Umgebung. Diese setzt sich aus der Stadt zusammen mit Assets wie Straßen, Gebäuden, Bäumen und Fahrzeugen. Zum anderen gibt es die Benutzeroberfläche, die den Anwender durch die Software begleitet. Hierbei ist anzumerken, dass die gesamte Simulation des Verkehrs selbst implementiert wurde. Das gesamte Straßennetz besteht aus mehreren, mit einander verknüpften Wegpunkten, zwischen welchen sich die Verkehrsteilnehmer bewegen können. Zudem bestimmen weitere Skripte das Verhalten an Kreuzungen.

Neben dem freien Modus, in dem man sich beliebig durch die Simulation bewegen kann, gibt es diverse Szenarien, die ein direktes Ereignis innerhalb der Stadt zeigen, wie u. a. einen Rotlichtverstoß oder einen Unfall mit einem Reh. Die Szenarien sind dabei kategorisiert nach *Verhalten, Verstoß, Halten und Unfall.* Der Anwender kann zusätzlich durch die Benutzeroberfläche globale, regionale oder objektbezogene manipulative Einstellungen vornehmen. Dies umfasst die Veränderung von Geschwindigkeit, Wahrscheinlichkeit von Auffahrunfällen sowie Rotlichtverstößen.

[3] https://visualstudio.microsoft.com

[4] https://www.jetbrains.com/de-de/rider/

[5] https://www.blender.org/

[6] https://www.autodesk.de/products/maya/overview

[7] https://www.adobe.com/products/substance3d-painter.html

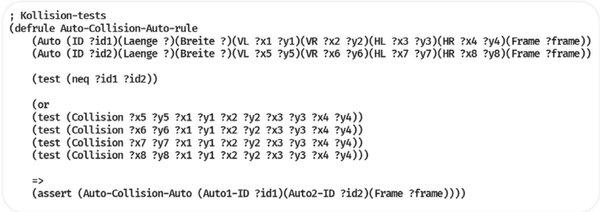
[8] https://www.adobe.com/de/products/photoshop.html

[9] https://git-scm.com/

Abbildung 2: Einblick in die Verkehrssimulation

**Regelsystem** CLIPS ist ein Werkzeug zur Erstellung von Expertensystemen, welche Regeln und Fakten beinhalten, die durch eine Inferenzmaschine ausgeführt werden. Ein Fakt repräsentiert und speichert spezifische Informationen innerhalb der Wissensbasis. Mithilfe eines Templates kann deren Aufbau strukturiert und genormt werden. Generiertes Wissen wird in Regeln repräsentiert. Diese arbeiten mit Fakten, wobei immer eine Voraussetzung erfüllt sein muss, damit die Regel ausgelöst und eine Aktion ausgeführt wird. Die Wissensbasis ist der Ort, wo alle Fakten und Regeln gespeichert werden.

Abbildung 3 zeigt den Aufbau einer Beispielregel zum Erkennen eines Autounfalls. Als Voraussetzung muss es zwei Autos mit unterschiedlicher ID geben. Zudem muss noch gelten, dass sich beide Autos berühren. Dies wird über die mathematische Funktion „Collision“ berechnet. Die Aktion wird dann nach dem „=>“ beschrieben. In diesem Fall wird ein neuer Fakt, der die Informationen des Autounfalls beinhaltet, angelegt. Alle weiteren Regeln des Projektes sind nach diesem Schema aufgebaut.



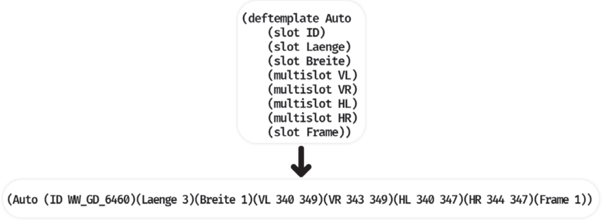
```
; Kollision-tests
(defrule Auto-Collision-Auto-rule
    (Auto (ID ?id1)(Laenge ?)(Breite ?)(VL ?x1 ?y1)(VR ?x2 ?y2)(HL ?x3 ?y3)(HR ?x4 ?y4)(Frame ?frame))
    (Auto (ID ?id2)(Laenge ?)(Breite ?)(VL ?x5 ?y5)(VR ?x6 ?y6)(HL ?x7 ?y7)(HR ?x8 ?y8)(Frame ?frame))

    (test (neq ?id1 ?id2))

    (or
    (test (Collision ?x5 ?y5 ?x1 ?y1 ?x2 ?y2 ?x3 ?y3 ?x4 ?y4))
    (test (Collision ?x6 ?y6 ?x1 ?y1 ?x2 ?y2 ?x3 ?y3 ?x4 ?y4))
    (test (Collision ?x7 ?y7 ?x1 ?y1 ?x2 ?y2 ?x3 ?y3 ?x4 ?y4))
    (test (Collision ?x8 ?y8 ?x1 ?y1 ?x2 ?y2 ?x3 ?y3 ?x4 ?y4)))

    =>
    (assert (Auto-Collision-Auto (Auto1-ID ?id1)(Auto2-ID ?id2)(Frame ?frame))))
```

Abbildung 3: Aufbau der Beispielregel Unfall

Zum Zeitpunkt der Evaluation (siehe Abschnitt 4) umfasste das Expertensystem 59 Regeln, 56 Templates sowie zwölf Funktionen für mathematische Berechnungen. Dabei ist die Wissensbasis in die vier Themenbereiche unterteilt. Jeder dieser Bereiche enthält für sich relevante Regeln und Templates. Zudem gibt es noch einen Bereich, der allgemein relevante Templates, wie zum Beispiel für Fahrzeuge oder Ampeln, enthält.

**Fakten und ihre Generierung** Damit das Expertensystem arbeiten kann, muss es viele Fakten erhalten, die zuvor aus der Simulation gewonnen werden. Jeder Fakt aus dem Simulationssystem besitzt mindestens sechs wichtige Grundbausteine. Einmal die Bezeichnung des Faktes gefolgt von Slots für die ID des Objektes und vier Punkte zur Positionsbestimmung des Objektes innerhalb der Simulation. Diese Punkte spannen ein Rechteck auf, in welchem sich das Objekt befindet. In Abb. 4 sieht man ein Beispiel für ein deftemplate, welches diese Bausteine vorgibt und aus welchem ein Fakt entsteht.

Weiterhin gibt es zwei Arten von Fakten. Die statischen Fakten müssen nur einmalig generiert werden und gehören zu Objektenklassen, welche sich in der Simulation nicht verändern (zum Beispiel Häuser, Kreuzungen, Ampeln u. v. m.). Die dynamischen Fakten werden repetitiv und ständig abgefragt, da sich die Objekte bewegen oder ihren Status ändern können (zum Beispiel Fahrzeuge oder Personen).



```
(deftemplate Auto
    (slot ID)
    (slot Laenge)
    (slot Breite)
    (multislot VL)
    (multislot VR)
    (multislot HL)
    (multislot HR)
    (slot Frame))
```

```
(Auto (ID WW_GD_6460)(Laenge 3)(Breite 1)(VL 340 349)(VR 343 349)(HL 340 347)(HR 344 347)(Frame 1))
```

Abbildung 4: Aufbau eines Faktes aus Unity. Oben ist das Template zu sehen, unten ein daraus generierter Fakt.

Das Generieren eines Faktes geschieht zur Laufzeit des Programmes. Im Grunde besitzt jedes Objekt in der Simulationsumgebung eine rechteckige Box, von welcher es umrandet wird (vgl. Abbildung 5). Das Entscheidende sind die vier Eckpunkte der Box, die zusammen ein Rechteck aufspannen. Ein Script in Unity berechnet dann diese Punkte und schreibt sie sortiert als Fakt in eine Datei. Bei dynamischen Fakten findet die Abfrage zum Generieren eines neuen Faktes sekündlich statt.

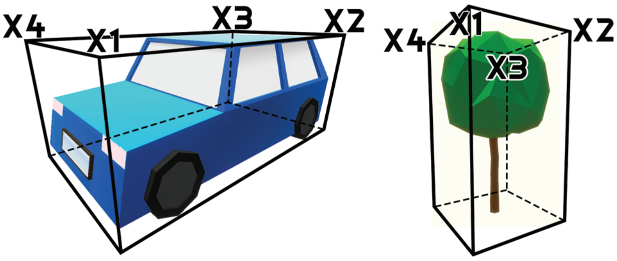
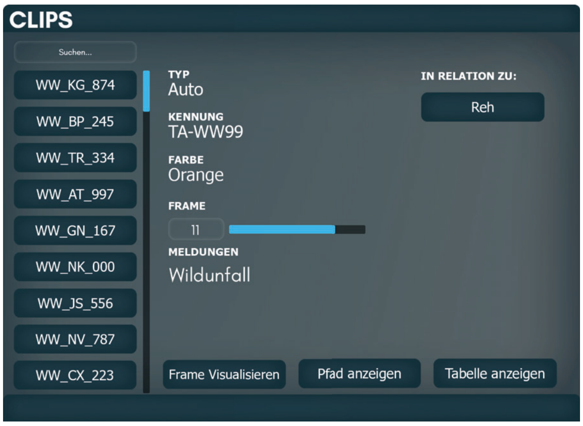


Abbildung 5: Visualisierung der Koordinaten aus Unity anhand der zwei Beispielklassen *Auto* und *Baum*

**Ereignistabelle** Innerhalb der Simulationsumgebung kann über eine Schaltfläche eine Tabelle geöffnet werden. In dieser werden Informationen zu jedem Fahrzeug angezeigt. Darüber hinaus werden auch die Ergebnisse, die das Expertensystem zuvor zum jeweiligen Fahrzeug analysiert und erkannt hat, dargestellt. In Abbildung 6 wird beispielsweise gezeigt, dass ein Auto mit der Kennung *TA-WW99* einen Unfall mit einen Reh zum Zeitpunkt des elften Frames hatte.



TYP
Auto

KENNUNG
TA-WW99

FARBE
Orange

FRAME
11

MELDUNGEN
Wildunfall

IN RELATION ZU:
Reh

Abbildung 6: Einblick auf die Ereignistabelle, welche in der Verkehrssimulation die gezeigten Fakten aus dem Expertensystem anzeigt.

**Zusammenspiel von Simulation und Expertensystem** Alle Fakten, die die Simulation generiert, werden in einer Datei gespeichert. Über eine Schaltfläche wird diese nach CLIPS exportiert. Mithilfe der neuen Fakten und der vorhandenen Wissensbasis leitet CLIPS dann Erkenntnisse und Ergebnisse zu den vier verschiedenen Themenbereichen (siehe Regelsystem) ab. Diese werden wiederum in eine Textdatei geschrieben, welche von Unity gelesen werden kann. Mithilfe eines Fakten-Managers werden die neu inferierten Fakten gespeichert und in der Ereignistabelle angezeigt. In Abbildung 1 wird dieser Vorgang grafisch dargestellt.

## 4 Evaluation und Ergebnisse

Damit qualitative Ergebnisse zum Beantworten der Forschungsfrage erfasst werden können, wurde das Expertensystem mitsamt der Simulation in einer Evaluation beurteilt.

**Ablauf und Struktur** Die Evaluation war begleitet durch eine Software-Dokumentation und Evaluationsanleitung. Die Dokumentation erklärte die Installation, die Steuerung und alle Benutzeroberflächen der Software. In der Evaluationsanleitung folgten nach einer kurzen Projektvorstellung die Aufgaben, die von den Probanden zu bewältigen waren. Als erstes sollte sich der Proband mit der Software vertraut machen, indem er sich fünf Minuten frei in der Stadt umschaut und anfallende Fehler notiert. Daraufhin sollte er mit den Werkzeugen der Software verschiedene Verkehrsszenarien, zum Beispiel einen Autounfall oder Stau, herbeiführen. Die letzte Aufgabe diente zum Überprüfen der Regeln des Expertensystems. Dafür sind Szenen in Unity eingebunden, die der Proband auswählen kann. Eines dieser Testszenarien beschreibt beispielsweise, dass ein Fahrzeug an einer Tankstelle anhält und im Anschluss weiterfährt. Der Proband soll dabei testen, ob er und das Expertensystem zur gleichen Zeit die gleichen Ereignisse erkennen.

**Durchführung** An der Durchführung der Evaluation waren Angehörige aus zwei verschiedenen Hochschulen beteiligt. Sie erfolgte in zwei Sitzungen. Die erste Sitzung wurde mit den Professoren und wissenschaftlichen Mitarbeitern durchgeführt, die zweite mit Studierenden. Für die Evaluation wurde sich in Zoom getroffen. Nach Vorstellung des Projektes bekamen die Probanden die Dokumentation und Evaluationsanleitung sowie auch einen ausführbaren Prototypen der Software übermittelt und sollten selbstständig die Aufgaben bewältigen. Für Unklarheiten sollte zuerst in die Dokumentation geschaut werden, wobei Rückfragen jederzeit gestellt werden durften.

**Ergebnisse** Abbildung 7 zeigt die Evaluation der pragmatischen und hedonischen Qualität der Software mittels AttrakDiff[10] im Sinne der DIN EN ISO 9241-11 [DIN18].

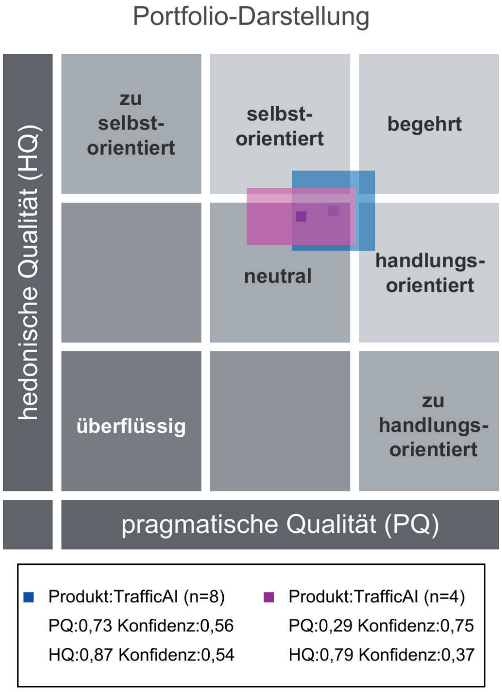


Abbildung 7: Matrix mit der durchschnittlichen Ausprägung von PQ und HQ des AttrakDiff Fragebogens - In blau die Ausprägung der Professoren und wissenschaftlichen Mitarbeiter - In violett die Ausprägung der Studenten

Sowohl die pragmatische Qualität, welche die Benutzbarkeit der Software beschreibt, als auch die hedonische Qualität, die widerspiegelt, wie stark sich der Nutzer mit der Software identifiziert, liegen im neutralen Bereich. Daraus lässt sich schlussfolgern, dass noch einige Anpassungen an der Software vorzunehmen sind, um eine optimale User Experience zu erzielen.

Zum Testen des Expertensystems sollten die Probanden bestimmte Ereignisse erkennen und dokumentieren, ob und wann das Expertensystem diese auch erkennt. In Abbildung 8 wird gezeigt, welche Differenzwerte dabei zwischen den Probanden und dem Expertensystem entstanden sind. Die Differenz wird dabei in Sekunden dargestellt.

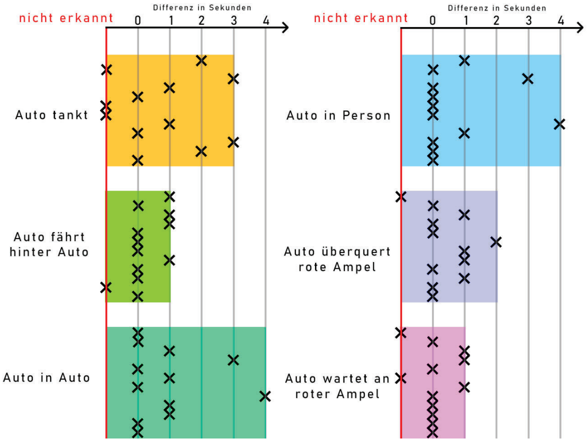


Abbildung 8: Evaluationsergebnis des Expertensystems. Auf der y-Achse werden die verschiedenen Szenen dargestellt. Die X-Achse gibt den Differenzwert zwischen Proband und CLIPS in Sekunden an.

Es ist deutlich zu erkennen, dass das Expertensystem meistens die Ereignisse erkennt. Kleinere Abweichungen können auf menschliche Ungenauigkeit oder individuelle Definitionen zurückgeführt werden. Bspw. erkennt Proband A das Tanken erst eine Sekunde später als Proband B, da er entweder unaufmerksam war oder für sich anders definiert hat, wann ein Auto tankt.

Es sind aber auch gravierende Abweichungen zu erkennen, die eindeutig auf eine Ungenauigkeit der Software hinweisen. Dabei kann ausge-

[10] https://www.attrakdiff.de/

schlossen werden, dass die Regeln fehlerhaft sind. Unter gleichen Bedingungen wird eine Regel immer ausgeführt. Bei der Regel *Auto tankt* ist die Bedingung beispielsweise *Auto hält in Tankstelle*. Da diese Regel bei mehreren Probanden erkannt wurde, ist ihre funktionale Richtigkeit damit bewiesen. Das Nicht-Erkennen der Regel ist dadurch auf Fehler in der Simulation, der Generierung der Fakten oder den Schnittstellen innerhalb der Software zu schließen.

**Diskussion** Anhand der Ergebnisse und der Simulationsumgebung wird gezeigt, dass ein Expertensystem verschiedene Ereignisse des Straßenverkehrs erkennen und analysieren kann, wodurch es sich grundlegend zur Verkehrsüberwachung eignet. Das Wichtigste hierfür ist eine funktionierende und ausgereifte Wissensbasis, die in vielen Iteration getestet und ausgebessert wird. So wurde ein Programm entwickelt, das Verkehrssituationen virtuell testen und analysieren kann, wobei keine realen Fahrzeuge zu Schaden kommen müssen. Es können somit kritische Verkehrssituationen ohne Risiken für Leben und Materialien nachgestellt und analysiert werden. Das Expertensystem ist dabei aber auf den eigenen Regelsatz und die vorhandenen Fakten limitiert. Deswegen muss durchgehend sichergestellt sein, dass es alle generierten Fakten besitzt. Das Fehlen eines einzelnen Faktes führt dazu, dass Regeln nicht ausgelöst werden. Zudem kann es keine Ereignisse erkennen, welche zuvor nicht durch Fakten und Regeln definiert wurden. Daraus folgt, dass jede Regel gründlich getestet werden muss, um eine funktionierende Wissensbasis zu gewährleisten. Gleichzeitig ist es aber auch möglich, neue Regeln hinzuzufügen, ohne Gefahr zu laufen, das schon etablierte und getestete Verhalten negativ zu beeinflussen oder gar einzubüßen. Diese Kombination aus festen bereits wohlstrukturierten Bestandteilen und flexiblen noch unvollständigen Ergänzungen macht wesentlich den Wert des insgesamt wissensbasierten Ansatzes aus. Ähnliche Aussagen lassen sich über die Definition von Wissen auf unterschiedlichen Abstraktionsniveaus treffen, woraus sich interessante Aufgabenstellungen für die Zukunft ableiten lassen.

## 5 Zusammenfassung und Ausblick

Mit dieser Arbeit sollte die Frage beantwortet werden, ob sich Expertensysteme zur Verkehrsüberwachung eignen. Diese Feststellung könnte helfen, den Freiraum für Fehler beim Analysieren des Verkehrs durch Künstliche Intelligenz zu verringern.

Dafür wurde mithilfe von CLIPS im kleinen Umfang ein Expertensystem geschaffen, das Fakten aus einer in Unity erstellten 3D-Verkehrsumgebung enthält. Dadurch können kritische Verkehrssituationen ohne Risiken auf menschliche oder materielle Schäden nachgestellt, simuliert, aber auch ausgewertet und analysiert werden. In einer Evaluation wurde die Software getestet. Das Ergebnis lieferte eine Antwort auf unsere Kernfrage: Ja, Expertensysteme eignen sich grundlegend zur Verkehrsanalyse.

Jedoch ist hierbei Folgendes zu beachten: Der Umfang der Software spiegelt nur einen kleinen Teil des eigentlichen Straßenverkehrs wider. Es gibt zudem eine Verzögerung innerhalb der Schnittstelle des Expertensystems zum Simulationssystem von 1-2 Sekunden, was den Rahmen der Echtzeitfähigkeit der aktuellen Software vorgibt. Zudem hat diese jederzeit Zugriff auf alle Daten der Simulation, was sich nicht zwingend in allen Fällen auf ein System im realen Straßenverkehr abbilden lässt. Trotz dieser Tatsachen ist nicht außer Acht zu lassen, dass sich Expertensysteme bereits in anderen Bereichen wie der Medizin etabliert haben. Dieses Paper soll motivieren, weitere Forschungen in diesem Teilbereich der KI und den möglichen Nutzen im Straßenverkehr anzuregen. So könnten Expertensysteme in naher Zukunft dazu dienen, Verkehrsteilnehmer auszubilden oder den Straßenverkehr zu analysieren und zu überwachen.

Für die weitere Entwicklung könnte der Regelsatz von CLIPS sowie die 3D-Umgebung der Software weiter ausgebaut werden, um mehr reale und komplexere Verkehrssituationen abzudecken. Eine andere Möglichkeit wäre, ein Baukastensystem zu implementieren, das dem Nutzer erlaubt, selbst Straßensysteme aufzubauen. Dadurch könnte der Nutzer gezielt Szenarien zum Analysieren erstellen.Eine ebenfalls interessante Erweiterung wäre die Schaffung eines interaktiven grafikbasierten Regel-Editors zur Arbeit mit der Wissensbasis direkt im 3D-Kontext der Anwendung.

## Danksagung

Wir danken Jens Scholz, der dem Projekt über ein Jahr beiseite stand und tatkräftig an der Verwirklichung des Projektes geholfen hat.

## Literatur

[CH93] Edmond Chin-Ping Chang and Kunhuang Huarng. Freeway incident management expert system design. *Transportation Research Record*, 1993.

[DIN18] DIN EN ISO 9241-11. DIN EN ISO 9241-11:2018-11, ergonomie der mensch-system-interaktion - teil 11: Gebrauchstauglichkeit: Begriffe und konzepte (ISO 9241-11:2018); deutsche fassung EN ISO 9241-11:2018, 11 2018.

[KKB06] Yong-Kul Ki, Jin-Woo Kim, and Doo-Kwon Baik. A traffic accident detection model using metadata registry. In *Fourth International Conference on Software Engineering Research, Management and Applications (SERA'06)*, pages 255–259, 2006.

[KS12] Hamed Khakzad and Hossein Shirazi. Tourism expert system with clips using pfc. *AISP 2012 - 16th CSI International Symposium on Artificial Intelligence and Signal Processing*, pages 299–304, 05 2012.

[LHL$^+$22] Chumeng Liang, Zherui Huang, Yicheng Liu, Zhanyu Liu, Guanjie Zheng, Hanyuan Shi, Yuhao Du, Fuliang Li, and Zhenhui Li. Cblab: Scalable traffic simulation with enriched data supporting. In *ICLR 2023*, 10 2022.

[MMJAG20] Vishal Mandal, Abdul Rashid Mussah, Peng Jin, and Yaw Adu-Gyamfi. Artificial intelligence-enabled traffic monitoring system. *Sustainability*, 12(21), 2020.

[NHN$^+$22] Nhu Nguyen, Minh Hoang, Nguyen, La Loc, Thi Truong, Thi, Duong Hieu, Thanh Le, Duy Lai, Nguyen Le, Nam Thoai, and Ho Chi. Ai-based traffic counting: A case study in vietnam. In *2022 International Conference on Advanced Computing and Analytics (ACOMPA)*, 11 2022.

[SAAN21] Lamis Samhan, Amjad Alfarra, and Samy Abu-Naser. An expert system for knee problems diagnosis. *International Journal of Academic Information Systems Research (IJAISR)*, pages 59–66, 04 2021.

[SQSL22] Sanghyun Son, Yi-Ling Qiao, Jason Sewall, and Ming C. Lin. Differentiable hybrid traffic simulation. *ACM Transactions on Graphics*, 41(6):1–10, nov 2022.

[TWK$^+$16] C.F. Tan, L.S. Wahidin, S.N. Khalil, N. Tamaldin, J. Hu, and G.W.M. Rauterberg. The application of expert system: a review of research and applications. *ARPN Journal of Engineering and Applied Sciences*, 11(4):2448–2453, 2016.

# Finding the Optimum Design of Large Gas Engines Prechambers Using CFD and Bayesian Optimization

Stefan Posch[1], Clemens Gößnitzer[1], Franz Rohrhofer[2], Bernhard C. Geiger[2], and Andreas Wimmer[1,3]

[1]LEC GmbH
[2]Know-Center GmbH
[3]Institut für Thermodynamik und nachhaltige Antriebssysteme, TU Graz

## Abstract

The turbulent jet ignition concept using prechambers is a promising solution to achieve stable combustion at lean conditions in large gas engines, leading to high efficiency at low emission levels. Due to the wide range of design and operating parameters for large gas engine prechambers, the preferred method for evaluating different designs is computational fluid dynamics (CFD), as testing in test bed measurement campaigns is time-consuming and expensive. However, the significant computational time required for detailed CFD simulations due to the complexity of solving the underlying physics also limits its applicability. In optimization settings similar to the present case, i.e., where the evaluation of the objective function(s) is computationally costly, Bayesian optimization has largely replaced classical design-of-experiment. Thus, the present study deals with the computationally efficient Bayesian optimization of large gas engine prechambers design using CFD simulation. Reynolds-averaged-Navier-Stokes simulations are used to determine the target values as a function of the selected prechamber design parameters. The results indicate that the chosen strategy is effective to find a prechamber design that achieves the desired target values.

## 1 Introduction

Internal combustion engines (ICE) will continue to play an important role in the generation of energy and the transportation of goods. The high energy density of the used fuels in combination with the robust and reliable technology, the ICE will still be the first choice for applications in which pure electrified systems make little sense as it is the case for the shipping of goods. Furthermore, ICE will be a key part for future energy management by the reconversion of chemical energy storage media such as hydrogen into electricity or to ensure grid stability which is induced by the increasing use of alternative energy sources such as wind and solar energy [GBV+19]. Hence, the research on increasing engine efficiencies under the fulfilment of stringent emission legislation will continue regardless of whether classical carbon-based or alternative carbon-free fuels are used. In the case of spark ignited (SI) engines, a common approach is the operation under lean conditions which increases the engine efficiency by increasing the specific heat capacity of the working fluid, reducing heat losses through the cylinder walls, and reducing pumping losses in part load operation [ZALY22]. However, the operation at lean conditions comes with the drawback of unstable combustion and thus, high cyclic fluctuations that can even lead to misfiring cycles. The enhancement of the ignition energy by increasing the performance of the ignition system helps to overcome these issues. A well-established concept to increase the ignition efficiency of lean mixture SI engines is the turbulent jet ignition (TJI) by the use of a separate, small (compared to the main combustion chamber) volume known as prechamber. The mixture is ignited by the spark plug in the prechamber and the flame propagates through the volume increasing the pressure and force hot products and active radicals to flow into the main combustion chamber via overflow bores. Thus, the ignition of the mixture in the main combustion chamber is induced at multiple locations due to the penetrating turbulent jets coming from the prechamber. However, the design of a prechamber and the choice of the appropriate operating conditions to achieve a highly efficient TJI system in combination with the main combustion chamber is not straightforward due to the large number of parameters and thus degrees of freedom. Therefore, experimental investigations as shown by Roethlisberger and Favrat [RF03] or Novella et al. [NPGS$^+$20] are complex as well as cost intense and thus limit the number of design variants that can be tested. To decrease the experimental effort, the state-of-the-art procedure is to apply numerical simulation, i.e. computational fluid dynamics (CFD), to increase the number of design and operation parameter variants by virtual testing and to achieve a comprehensive view on the physical effects which measurements are not able to provide.

Various studies regarding the application of CFD to evaluate prechamber efficiencies and related values can be found in literature. Benajes et al. [BNGS$^+$20] investigated the scavenging and turbulence distributions of several prechamber designs under ultra-lean conditions. The results of the study indicate that the reduction of the laminar flame speed due to the ultra-lean conditions significantly decreases the quality of the turbulent jets exiting the prechamber. Silva et al. [SSH$^+$20] carried out CFD combustion simulations to investigate the influence of several design parameters on the engine combustion characteristics. The authors stated that the diameter of the overflow bores impacts the peak pressure and the residence time of the main charge in the prechamber. Furthermore, the results show a large influence of the neck diameter on the pressure increase in the prechamber as well es the aforementioned residual time. Related works can be found in Feng et al. [FZQ$^+$18], Winter et al. [WSP$^+$20], or Zhang et al. [ZYE$^+$20]. In addition to common CFD simulation, data-driven approaches and machine learning (ML) concepts are increasingly finding their way into the design optimization of ICE components to facilitate the discovery of appropriate prechamber configurations. Ge et al. [GBY$^+$21] compared different ML models to develop a surrogate model which represents combustion CFD simulations. The trained model is further used in a genetic algorithm to find an optimum prechamber design. To decrease the number of required CFD simulations, a Bayesian updating optimization method was applied. In a similar study, Silva et al. [SMB$^+$22] compared design of experiments (DOE) optimization with different ML surrogate models and combinations of optimization techniques via a so-called automated super-learner framework. The results indicate that the proposed method performance is superior compared to the classical DOE approach. Although these studies use ML surrogate models for the optimization process, the training of the models requires a certain number of combustion CFD simulations which are computationally expensive. To overcome this issue, Posch et al. [PWZ$^+$21] presented a method to use condensed CFD simulations to train ML models, which are able to predict trends in the prechamber performance, thus helping to reduce the computational effort. Following the aforementioned studies of Ge et al. [GBY$^+$21] and Silva et al. [SMB$^+$22], the present study deals with the optimization of a prechamber design by the use of CFD and appropriate optimization techniques. A Bayesian optimization (BO) algorithm based on Gaussian processes is applied which is initialized by a DOE-based CFD data set. To reduce the computational effort, the CFD simulation only covers the range from bottom dead center until ignition timing (IT) rather than the entire cycle.

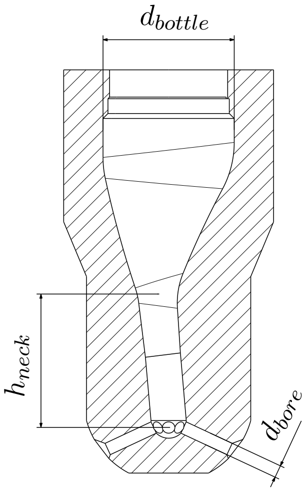


Figure 1: Base prechamber design and the three input parameters

This simplification can be justified as follows: (i) The present study focuses solely on the prechamber behaviour and thus gas exchange through the valves in the prechamber is neglected. (ii) Since the flow field around the spark plug at IT has a significant influence on the flame kernel development [GKK$^+$19, GG21, PGO$^+$22] and furthermore on the combustion in the pre- and main chamber, the optimization process focuses on the turbulent kinetic energy and velocity magnitude around the spark plug at IT. The results show that the proposed method is able to find a prechamber design that covers three design parameters and that fulfills the target values in terms of maximum turbulent kinetic energy by considering a certain maximum velocity magnitude around the spark plug. Figure 1 shows the base prechamber design with the three input parameters for the optimization process.

## 2 Methodology

### 2.1 Engine specification

The engine considered here was adapted from an existing engine. The main chamber was simplified to a perfect cylinder, and the prechamber had variable geometrical sizes as described later. The simplified engine had a bore of 190 mm, a stroke of 220 mm, a displaced volume of about 6.2 L and a speed of 1500 rpm. No valves or gas exchange were modelled. Thus, the time of interest for this study covered the engine cycle ranging from bottom dead centre at 540 CAD until ignition timing at 695 CAD.

### 2.2 CFD setup

The Reynolds-averaged Navier-Stokes (RANS) CFD simulations were performed with OpenFOAM (version from November 2022 [Opea] from The OpenFOAM Foundation [Opeb]). Meshing was done with snappyHexMesh. The grid size was 4 mm in the cylinder, 0.25 mm in the prechamber and 0.125 mm in the prechamber bores and near the spark plug, leading to approximately 1 mio cells. Turbulence was modelled by the RANS $k$-$\omega$ SST two-equation model. Initial temperature and pressure were set to 400 K and 3 bar, respectively. The time step was adapted to keep the Courant number below 2. The PISO algorithm with three correctors was used for pressure-velocity coupling. Additionally, two non-orthogonal correctors were used to improve convergence of the pressure prediction. Furthermore, second-order bounded discretization schemes were used for gradient and divergence, whereas a first-order implicit time discretization was used. Since the maximum observed speeds exceed a Mach number of 0.3, the transonic option was set for all simulations. Mesh movement was done with the standard OpenFOAM layered engine approach. Mass conservation was monitored by integration of the density over the volume, to check if the total mass of the closed volume remains constant, which indeed was the case for all simulations. A more detailed description of the simulation setup can be found in [GPP22]. For the evaluation of the quantities near the spark plug, all cells within a sphere with radius of 0.5 mm just outside of the spark plug gap were monitored. The relevant simulation output, namely the turbulence intensity measured by the volume-average value of $k$ and velocity magnitude $|\vec{v}|$ are both evaluated inside this sphere.

### 2.3 Optimization strategy

Three geometrical parameters of the prechamber were adjusted in this study to find the optimal shape: the diameter of the prechamber bottle ($d_{\text{bottle}}$), the diameter of the overflow bores ($d_{\text{bore}}$), and the height of the prechamber neck ($h_{\text{neck}}$). The optimization ranges were chosen as $d_{\text{bottle}} \in [8, 12]$, $d_{\text{bore}} \in [0.75, 1.15]$ and $h_{\text{neck}} \in [15, 20]$.

From the perspective of optimization, and summarizing the geometrical parameters of the prechamber in the tuple $\mathbf{x} = (d_{\text{bottle}}, d_{\text{bore}}, h_{\text{neck}})$, the aim is to find a set of solutions to

$$\max_{\mathbf{x}} k(\mathbf{x}) \quad \text{s.t.} \quad |\vec{v}(\mathbf{x})| \leq 25. \tag{1}$$

In BO [Fra18], both the objective $k$ and the constraint $|\vec{v}|$ are considered as black box functions that are expensive to evaluate, and that are thus approximated by surrogates. The most common surrogate models are Gaussian processes [RW06]. Given a dataset $\mathcal{D} = \{(\mathbf{x}_i, k(\mathbf{x}_i), |\vec{v}(\mathbf{x}_i)|)\}$ of different simulated settings, the Gaussian process models yield (Gaussian) distributions $\hat{k}$ and $|\hat{\vec{v}}|$ for the target $k$ and the constraint $|\vec{v}|$ for every possible parameterization $\mathbf{x}$. If $\mathbf{x}$ is close to a point $\mathbf{x}_i$ in the dataset $\mathcal{D}$, then the distribution of, e.g., $\hat{k}(\mathbf{x})$ has a mean close to $k(\mathbf{x}_i)$ and a small variance, whereas if $\mathbf{x}$ is far from every point in $\mathcal{D}$, the variance of $\hat{k}(\mathbf{x})$ will be large — there is high epistemic uncertainty. During optimization, BO relies on these probabilistic models $\hat{k}(\mathbf{x})$ and $|\hat{\vec{v}}(\mathbf{x})|$ to determine (using acquisition functions, see [Fra18]) candidate geometries $\mathbf{x}^\bullet$ that improve upon all geometries in $\mathcal{D}$ in the sense of (1). The thus proposed candidate geometries are then evaluated using CFD simulations, and consequently yield a new tuple $(\mathbf{x}^\bullet, k(\mathbf{x}^\bullet), |\vec{v}(\mathbf{x}^\bullet)|)$. This tuple is included in $\mathcal{D}$, and the probabilistic models $\hat{k}(\mathbf{x})$ and $|\hat{\vec{v}}(\mathbf{x})|$ are updated. This optimization loop is iterated until either a predetermined number of iterations is exceeded or until the utility of the solution is satisfactory.

To address the optimization problem (1), we select $\mathbf{x}^\bullet$ via the expected constrained improvement [GKX$^+$14]. Specifically, if $\mathbf{x}_j$ is the geometry in $\mathcal{D}$ that maximizes $\hat{k}(\mathbf{x}) = k(\mathbf{x})$, then $\mathbf{x}^\bullet$ is chosen as an optimizer of

$$\begin{aligned}\max_{\mathbf{x}} \mathbb{E}\left(\mathbf{1}(|\hat{\vec{v}}(\mathbf{x})| \leq 25) \cdot \max\{0, \hat{k}(\mathbf{x}) - k(\mathbf{x}_j)\}\right)\\ = \max_{\mathbf{x}} PF(\mathbf{x}) \cdot EI(\mathbf{x})\end{aligned} \tag{2}$$

where first factor is the probability that the candidate $\mathbf{x}$ is feasible (i.e., that $|\hat{\vec{v}}(\mathbf{x})| \leq 25$) and where the second factor is the standard expected improvement (w.r.t. $k$). Furthermore, to exploit the fact that CFD simulations can be run in parallel, batch (or parallel) BO was used, which responds with a set of $q$ candidate solutions $\{\mathbf{x}_1^\bullet, \dots, \mathbf{x}_q^\bullet\}$, cf. [Fra18, eq. (14)]

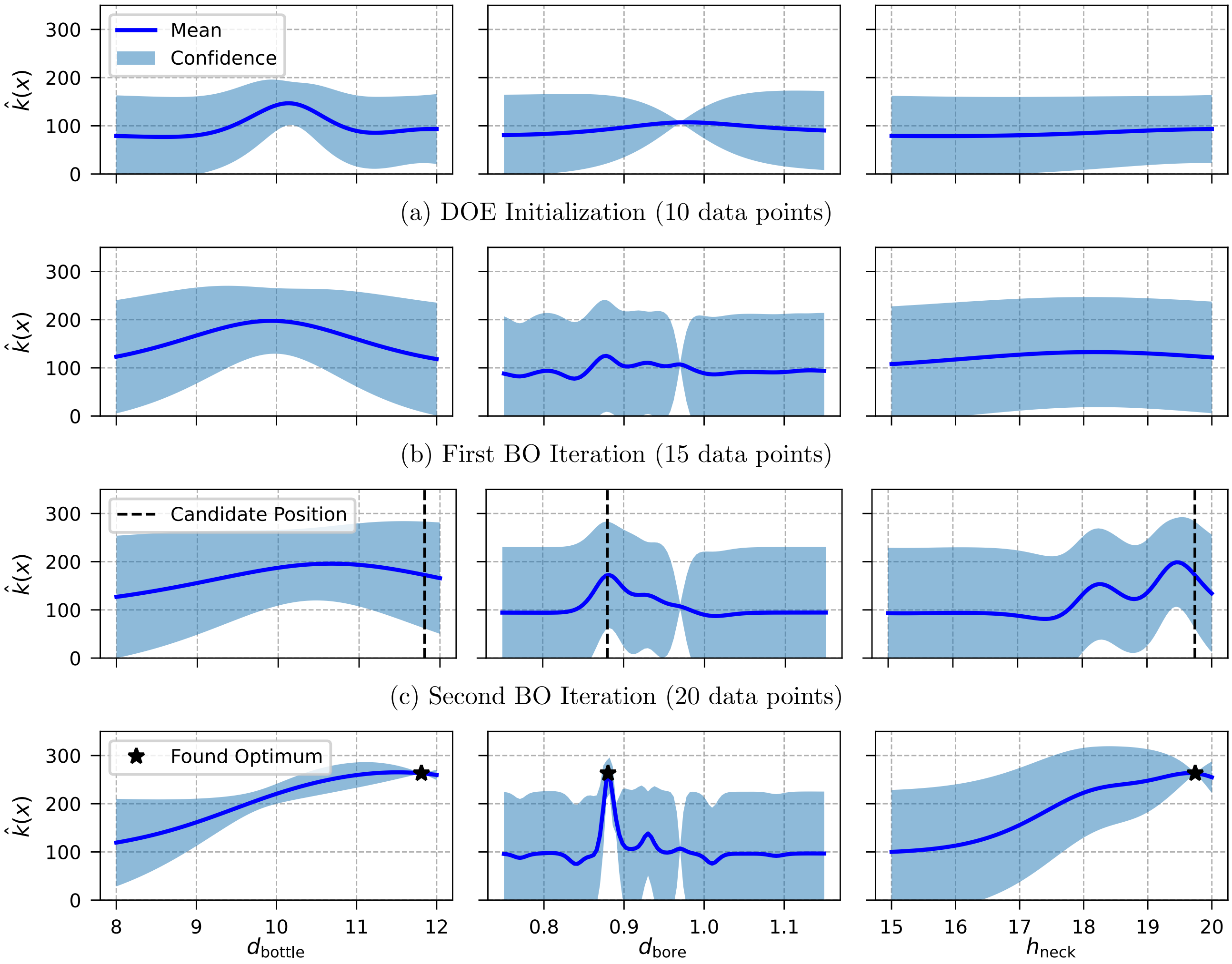
In this work, BO was implemented using BoTorch [BKJ$^+$20], a Python library that contains functionality for constrained, multi-objective, and batch BO. Specifically, both $\hat{k}$ and $|\hat{\vec{v}}|$ are modeled as Gaussian processes with a fixed noise standard deviation of 0.005 and a Matern kernel with optimized hyperparameters. For optimal functionality of the BO, the three geometry parameters $\mathbf{x}$ are scaled to the closed unit interval $[0, 1]$ and the objectives are scaled to unit variance ($k$ and $|\vec{v}|$) and zero mean (only $k$). The expected constrained improvement was evaluated using Monte Carlo sampling, with 1024 samples. The batch size was set to five.

A DOE study was performed with ten samples to create the initial data for three consecutive BO iterations based on latin hypercube formulation. The optimum value found by the DOE study was $k = 160.38$ (see Table 1).

## 3 Results

The results for the optimum values found by the DOE initialization and three consecutive BO iterations can be found in Table 1. While the first BO iteration shows considerable improvement ($k = 246.46$) compared to the DOE initialization, the second BO iteration was not able to further increase the objective within the following five candidate positions ($k = 244.56$). The third and final BO iteration again yields a new optimum value ($k = 263.16$) while still fulfilling the constraint ($|\vec{v}| = 22.53$).

Figure 2 illustrates the progress of BO over the three iterations. Given that after the third iteration the optimal value in $\mathcal{D}$ is $\mathbf{x}^\star = (d^\star_{\text{bottle}}, d^\star_{\text{bore}}, h^\star_{\text{neck}})$, the three columns show the mean function of $\hat{k}(\mathbf{x})$ and its standard deviation for two of the dimensions of $\mathbf{x}$ fixed to $\mathbf{x}^\star$, respectively. One can see that by including more training data, the surrogate $\hat{k}$ becomes more and more structured and that, especially w.r.t. $d_{\text{bottle}}$, the epistemic uncertainty reduces. The working of BO is best illustrated in the third and fourth rows of the figure: Figure 2c depicts the surrogate model $\hat{k}$ trained on 20 data points. The vertical dashed lines indicate the candidate position $\mathbf{x}^\bullet$ that maximizes the acquisition function (2). It can be seen that the candidate position lies at a local optimum of the mean of $\hat{k}$ for $d_{\text{bore}}$, with a value close to 180. Running CFD simulations with the thus suggested geometry parameters $\mathbf{x}^\bullet$ yields a value of $k(\mathbf{x}^\bullet)$ exceeding 250. Figure 2d then shows the surrogate $\hat{k}$ after including $\mathbf{x}^\bullet$ (and the remaining four geometries of the batch)



(a) DOE Initialization (10 data points)

(b) First BO Iteration (15 data points)

(c) Second BO Iteration (20 data points)

(d) Third BO Iteration (25 data points)

Figure 2: Progress of three consecutive BO iterations. The three columns show the mean function of $\hat{k}(\mathbf{x})$ and its standard deviation for two of the dimensions of $\mathbf{x}$ fixed to $\mathbf{x}^\star$, respectively

Table 1: Optimum values found by the DOE initialization (10 data points) and three consecutive BO iterations (5 data points per iteration).

| | $k$ | $\|\vec{v}\|$ | $d_{\text{bottle}}$ | $d_{\text{bore}}$ | $h_{\text{neck}}$ |
|---|---|---|---|---|---|
| DoE Init. | 160.38 | 17.93 | 10.20 | 0.89 | 18.75 |
| 1$^{\text{st}}$ BO It. | 246.46 | 22.70 | 10.02 | 0.88 | 18.26 |
| 2$^{\text{nd}}$ BO It. | 244.56 | 22.11 | 9.90 | 0.88 | 18.18 |
| **3$^{\text{rd}}$ BO It.** | **263.16** | **22.53** | **11.81** | **0.88** | **19.74** |

in the training data $\mathcal{D}$. It can be seen that training was successful in the sense that the $\mathbf{x}^\bullet = \mathbf{x}^\star$ now lies on the mean curve of $\hat{k}$ and that the epistemic uncertainty at this position is small.

Despite the fact that BO is a global optimization method, we have observed in our experiments that most candidate geometries were suggested in the vicinity of optima of the surrogate function. In other words, BO exploited knowledge about the currently best geometry rather than exploring geometries that have not been simulated so far. Upon inspection of Figure 2, one can see that the target $k(\mathbf{x})$ varies substantially with the bore diameter; with only few training data, this variation cannot be captured in the surrogate $\hat{k}(\mathbf{x})$. (Indeed, the surrogate $\hat{k}$ in Figure 2c underestimated the value of the $k$ at the candidate position indicated with the dashed lines, as can be seen by the star in Figure 2d.) This is aggravated by the fact that the kernel parameters are optimized based on $\mathcal{D}$ and, thus, may not be optimal for the black box functions $k$ and $|\vec{v}|$. Indeed, as the center image in Figure 2c suggests, the automatic hyperparameter optimization may have terminated with a kernel width that is too large to capture the variation of $k$ with the bore diameter (especially when compared with the corresponding images in Figure 2b and 2d, respectively). A manual setting of these hyperparameters, however, is not possible since the functions $k$ and $|\vec{v}|$ are unknown a priori.

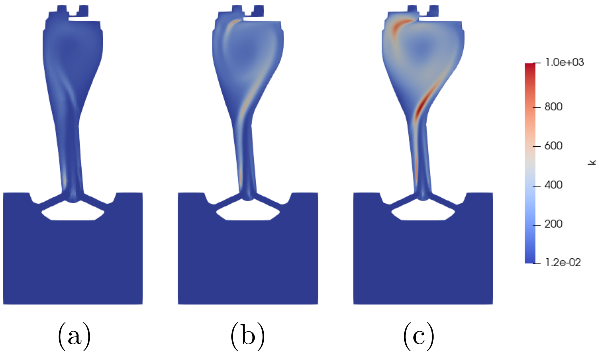


Figure 3: Turbulence intensity $k$-field in the prechamber cross section of one unfavourable design combination from the initial set (a), the best design combination of the initial set (b) and the final BO predicted design

Figure 3 shows the determined $k$-field in the cross section of the prechamber which allows a detailed analysis of the BO design parameter optimization results. While Figure 3a and Figure 3b represent an unfavourable and a favourable solution of the initial DOE set, respectively, Figure 3c shows the final resulting parameter combination found by the BO. It can be clearly observed, that the formation of a vortex in the prechamber is promoted by the appropriate combination of the chosen design parameters. Since the mass flowing from the main chamber into the prechamber is mainly influenced by the piston movement, a smaller value of $d_{\text{bore}}$ results in higher overflow velocity and consequently higher turbulence velocities in the prechamber for the predefined parameter range. It should be mentioned at this point that this effect is not unlimited since the reduction of the overflow area can lead to choking effects. The combination a small $d_{\text{bore}}$ value and high values of $d_{\text{bottle}}$ as well as $h_{\text{neck}}$ leads to a well-developed vortex with maximum values of $k$ in the vicinity of the spark plug.

## 4 Summary and Outlook

In the present study, a combination of CFD and BO was applied to optimize the design of a large bore gas engine prechamber. Three design parameter namely the diameter of the prechamber bottle, the diameter of the overflow bores and the height of the prechamber neck were chosen as input parameters. The target of the optimization process was maximizing the turbulence intensity

$k$ under a velocity magnitude constraint in the vicinity of the spark plug. After the initialization of the system via ten DOE-based samples, three BO steps were carried out. The analysis of the BO indicate that both the strong variability of $k$ with the bore diameter and the effects of automatic hyperparameter optimization appear to slow down BO. One may therefore assume that substantially more iterations are required for BO to enter the exploration phase and, consequently, suggest geometry parameters distinct from the current optima. An option to speed up BO is to chose acquisition functions that permit influencing the trade-off between exploration and exploitation, such as the upper confidence bound. Future work shall be devoted to investigate these alternatives. Furthermore, additional investigations regarding the complexity of the physical system and its sensitivity to design parameter changes will be performed in order to increase the input and output parameter space.

## Acknowledgments

The authors acknowledge the financial support of the Austrian COMET — Competence Centers for Excellent Technologies — Programme of the Austrian Federal Ministry for Climate Action, Environment, Energy, Mobility, Innovation and Technology, the Austrian Federal Ministry for Digital and Economic Affairs, and the States of Styria, Upper Austria, Tyrol, and Vienna for the COMET Centers LEC EvoLET and Know-Center, respectively. The COMET Programme is managed by the Austrian Research Promotion Agency (FFG).

## References

[BKJ+20] Maximilian Balandat, Brian Karrer, Daniel R. Jiang, Samuel Daulton, Benjamin Letham, Andrew Gordon Wilson, and Eytan Bakshy. BoTorch: A Framework for Efficient Monte-Carlo Bayesian Optimization. In Advances in Neural Information Processing Systems 33, 2020.

[BNGS+20] J. Benajes, R. Novella, J. Gomez-Soriano, I. Barbery, C. Libert, F. Rampanarivo, and M. Dabiri. Computational assessment towards understanding the energy conversion and combustion process of lean mixtures in passive pre-chamber ignited engines. Applied Thermal Engineering, 178:115501, 2020.

[Fra18] Peter I. Frazier. A Tutorial on Bayesian Optimization, 2018. `arXiv:1807.02811`.

[FZQ+18] Liyan Feng, Jun Zhai, Chuang Qu, Bo Li, Jiangping Tian, Lei Chen, Weiyao Wang, Wuqiang Long, and Bin Tang. The influence of the enrichment injection angle on the performance of a pre-chamber spark ignition natural-gas engine. Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering, 232(5):679–694, 2018.

[GBV+19] Elisa Guelpa, Aldo Bischi, Vittorio Verda, Michael Chertkov, and Henrik Lund. Towards future infrastructures for sustainable multi-energy systems: A review. Energy, 184:2–21, 2019. Shaping research in gas-, heat- and electric- energy infrastructures.

[GBY+21] Haiwen Ge, Ahmad Hadi Bakir, Suren Yadav, Yunseon Kang, Siva Parameswaran, and Peng Zhao. Cfd optimization of the pre-chamber geometry for a gasoline spark ignition engine. Frontiers in Mechanical Engineering, 6:599752, 2021.

[GG21] Clemens Gößnitzer and Shawn Givler. A new method to determine the impact of individual field quantities on cycle-to-cycle variations in a spark-ignited gas engine. Energies, 14(14):4136, 2021.

[GKK$^+$19] M. Ghaderi Masouleh, K. Keskinen, O. Kaario, H. Kahila, S. Karimkashi, and V. Vuorinen. Modeling cycle-to-cycle variations in spark ignited combustion engines by scale-resolving simulations for different engine speeds. Applied Energy, 250:801–820, 2019.

[GKX$^+$14] Jacob R Gardner, Matt J Kusner, Zhixiang Eddie Xu, Kilian Q Weinberger, and John P Cunningham. Bayesian optimization with inequality constraints. In Proc. Int. Conf. on Machine Learning (ICML), pages 937–945, 2014.

[GPP22] Clemens Gößnitzer, Stefan Posch, and Gerhard Pirker. Development of a framework for internal combustion engine simulations in openfoam. In 8th European Congress on Computational Methods in Applied Sciences and Engineering: ECCOMAS CONGRESS 2022, 2022.

[NPGS$^+$20] Ricardo Novella, Jose Pastor, Josep Gomez-Soriano, Ibrahim Barbery, Cedric Libert, Fano Rampanarivo, Chistou Panagiotis, and Maziar Dabiri. Experimental and numerical analysis of passive pre-chamber ignition with egr and air dilution for future generation passenger car engines. In WCX SAE World Congress Experience. SAE International, apr 2020.

[Opea] OpenFOAM Development Repository. Github commit c95d964 from 20th November 2022, `https://github.com/OpenFOAM/OpenFOAM-dev`. Accessed: 2023-01-03.

[Opeb] The OpenFOAM Foundation. `https://openfoam.org/`. Accessed: 2023-01-03.

[PGO$^+$22] Stefan Posch, Clemens Gößnitzer, Andreas B Ofner, Gerhard Pirker, and Andreas Wimmer. Modeling cycle-to-cycle variations of a spark-ignited gas engine using artificial flow fields generated by a variational autoencoder. Energies, 15(7):2325, 2022.

[PWZ$^+$21] S. Posch, H. Winter, J. Zelenka, G. Pirker, and A. Wimmer. Development of a tool for the preliminary design of large engine prechambers using machine learning approaches. Applied Thermal Engineering, 191:116774, 2021.

[RF03] R.P. Roethlisberger and D. Favrat. Investigation of the prechamber geometrical configuration of a natural gas spark ignition engine for cogeneration: part ii. experimentation. International Journal of Thermal Sciences, 42(3):239–253, 2003.

[RW06] Carl Edward Rasmussen and Christopher K.I. Williams. Gaussian Processes for Machine Learning. The MIT Press, 2006.

[SMB$^+$22] Mickael Silva, Balaji Mohan, Jihad Badra, Anqi Zhang, Ponnya Hlaing, Emre Cenker, Abdullah S. AlRamadan, and Hong G Im. Doe-ml guided optimization of an active pre-chamber geometry using cfd. International Journal of Engine Research, 0(0):14680874221135278, 2022.

[SSH$^+$20] Mickael Silva, Sangeeth Sanal, Ponnya Hlaing, Bengt Johansson, Hong G. Im, and Emre Cenker. Effects of geometry on passive prechamber combustion characteristics. In WCX SAE World Congress

Experience. SAE International, apr 2020.

[WSP+20] Hubert Winter, Eduard Schneßl, Gerhard Pirker, Jan Zelenka, and Andreas Wimmer. Application of CFD simulation to optimize combustion in prechamber gas engines with port injection. In Proceedings of the 6th European Conference on Computational Mechanics: Solids, Structures and Coupled Problems, ECCM 2018 and 7th European Conference on Computational Fluid Dynamics, ECFD 2018, pages 894–905, 2020.

[ZALY22] Sipeng Zhu, Sam Akehurst, Andrew Lewis, and Hao Yuan. A review of the pre-chamber ignition system applied on future low-carbon spark ignition engines. Renewable and Sustainable Energy Reviews, 154:111872, 2022.

[ZYE+20] Anqi Zhang, Xin Yu, Nayan Engineer, Yu Zhang, and Yuanjiang Pei. Numerical investigation of pre-chamber jet combustion in a light-duty gasoline engine. In Internal Combustion Engine Division Fall Technical Conference, volume 84034, page V001T06A013. American Society of Mechanical Engineers, 2020.

# First development steps towards autonomous driving

Marlena V. Hofer and Johannes Pieber

FH JOANNEUM – University of Applied Sciences, Institute of Automotive Engineering, Alte Poststraße 149, 8020 Graz, Austria

April 12th, 2023

## Abstract

In the framework of two consecutive Bachelor's theses [EH21], [DF20], an autonomously driving model car based on a *Smart Video Car Kit* for the *Raspberry Pi* was developed, which is able to use a camera and artificial intelligence based on neural networks to keep track on a road predetermined with adhesive strips, and to detect and react to objects on it.

In a subsequent Bachelor's thesis [HP22], partial results of this previous work were adapted to the application in real road traffic, and supplemented by additional system capabilities. For this purpose, a stereo camera with an integrated inertial measurement unit and a global navigation satellite system (GNSS) receiver were deployed. For the training of the artificial neural network, large amounts of images were collected with the stereo camera during real driving emission tests on the road, which served as data set for an object detection and recognition model. This data set was continuously optimized and complemented by publicly accessable databases provided by universities, original equipment manufacturers, and road safety authorities. In the end, a model was developed within the *YOLOv5* deep learning environment, which is able to detect traffic signs, traffic lights, vehicles, people and obstacles. Due to the enormous computational effort for the creation of the model, additional computing power of *Google Colab PRO* was utilized.

Furthermore, a lane detection model is under development. Lane line detection is currently the most fundamental step in lane line deviation warning systems, and higher levels of vehicle autonomy. For this purpose, a series of image preprocessing techniques and the sliding window algorithm are utilized to identify and track the lane line. For safety reasons, an emergency braking assistant was implemented that makes use of both stereoscopic distance detection and object detection.

The software also comprises a navigation system, based on a GNSS receiver and the *Google Maps* interface, which determines the position of the vehicle relative to the database of roads and displays the current position of the vehicle. In addition, a destination relative to the database of roads can be selected and the navigation system guides the vehicle to the selected destination.

In this paper, the theoretical background, the approach to the problem and the current state of this work in progress are presented and discussed.

## 1 Introduction

The first self-sufficient and truly autonomous cars appeared in the 1980s with Carnegie Mellon University's Autonomous Land Vehicle project [W$^+$85] and the European Eureka Prometheus Project (for an overview see, e.g., [Dic02]). Due to the enormous complexity of this task, almost three decades passed before this technology be-gan to flow into widespread application. Nowadays, autonomous vehicles are expected to revolutionize the future of transportation and travelling and are becoming increasingly important in vehicle development. The rapid increase in driver assistance systems, up to the development of fully autonomous vehicles does not only require a reorientation in industry (the automotive industry alone would not be able to achieve autonomous driving without support and computing power of Big Tech companies), but also necessary adaptations of the curricula in automotive engineering degree programs. In order to get started with this topic on undergraduate level education, it was decided to develop autonomous driving from scratch. The aim of this project is to build up knowledge in the field of autonomous driving in order to subsequently anchor this topic in education. For this purpose, an attempt was made to break down the complex task into sub-packages and to process them within the framework of Bachelor's theses. It started with the development of a lane detection and object recognition software, based on machine learning, for a model car [DF20], [DF21]. In a further development step, distance and position sensors were integrated and a navigation system, based on a GNSS receiver, was implemented [EH21], [EH22]. The results of this work are currently being incorporated into two further Bachelor's theses, in which the lane and object recognition software is being extended so that a real road vehicle can be controlled and operated with it.

This paper presents the self-programmed software for lane and object recognition, as well as driver assistance systems for an autonomously driving road vehicle, based on image analysis using artificial intelligence [HP22]. In the following sections, the essential modules of this software and the underlying mathematical models are described.

# 2 Object and Lane Detection

## 2.1 Object Detection

Neural networks are used in the majority of autonomous vehicles to detect and classify objects and pass them on to subsystems. All detected objects belong with a certain probability to one or more of the previously defined classes. In this work there are 63 of such classes. Most of the time, detected objects have certain probabilities in several classes. Depending on the selected threshold, one object, several objects or no object is considered to be detected. This threshold is in many cases in the range of 70% to 90%, but must be considered critically depending on the application. Especially in road traffic, for example, a higher threshold is assumed. In this work, an existing neural network developed by *Ultralytics*, i.e., *YOLO* in version 5, was used [A$^+$16]. Furthermore, a new model was trained using a custom data set. This neural network can be used in different sizes, from *Nano* to *XL*, which find different applications, such as *Nano* on microcomputers and *XL* in applications without time limit. In this work, the model ran in *Medium* on a *Nvidia Geforce RTX* graphics card, reaching between 6 to 12 frames per second. The dataset used was built in several iterations. In the first step, 1 500 photos were taken during a real driving situation in an urban environment.

To train a neural network, the data set, in this case the captured images, is first divided into training data and validation data. The training data represents 80% of the data, the validation data 20%. After each epoch, the weights and biases of the neural network layers are adjusted based on the results of the training data set. The mean-average-precision ($mAP$) of the neural network is then calculated from the validation data. The $mAP$ provides information on whether and how accurately the model identifies and classifies an object. For more information about the training procedures, see [HP22].

After a the first model was trained, it became clear that a much larger amount of data had to be used to generate a working network. However, with this knowledge, photos were collected in further test runs, leaving 4 500 photos for training after sorting. All photos that were not suitable for training, such as blurry or overexposed photos, were sorted out. In Fig. 1 a sample of the data set can be seen.

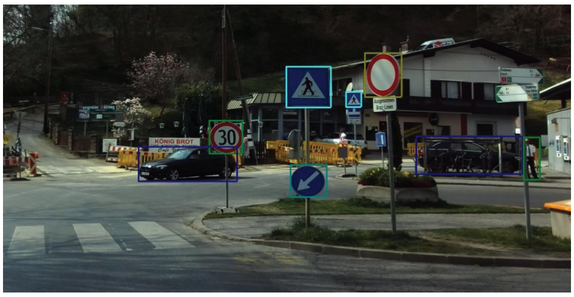


Figure 1: Sample photo of the dataset

A conspicuous behavior was detected when objects in the dataset were too small. Here, either unmarked objects were detected or marked objects were not detected. For this purpose, a program was utilized to erase the information of all labeled objects below a critical size and to paint over this area with a gray color. An example can be seen in Fig. 2.

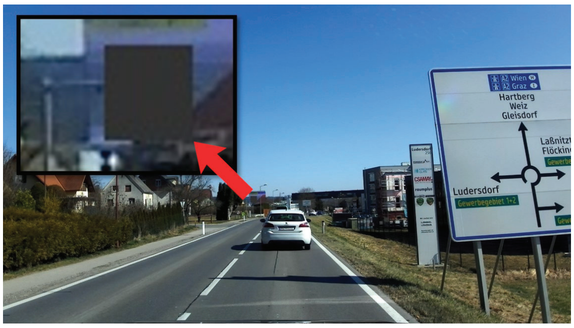


Figure 2: Deleted traffic sign

To further improve the results, the next step was to start manipulating these 4 500 photos to provide as much information as possible with just a few photos. For this purpose, the real photos were used as base photos and in the first step all existing marked traffic signs as well as traffic lights were cut out of these photos and classified. After at least 10 different objects were found for each class, they were cropped and augmented. During augmentation, the photo is randomly rotated, cropped, contrast and brightness are changed, and artifacts are added. Since all the photos were taken on bright days at midday, this step increases the variation of the images. This is important so that the neural network learns with as many situations as possible. An example of an augmented object can be seen in Fig. 3.

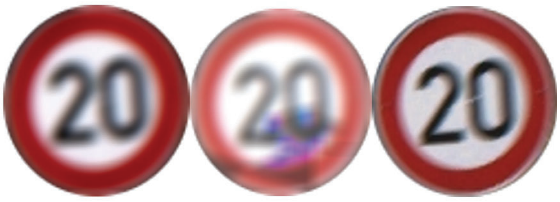


Figure 3: Augmented traffic sign

To put these objects back into the photos, a program was written which runs as follows: The base photos are duplicated with the help of the augmentation. In the next step one of these photos is selected as well as a cropped object. This new object is loaded into the photo and randomly positioned. Afterwards it is checked if the new object obscures an existing object in the photo. If this is not the case, this new object is added to the object label file of the photo. This process runs in a loop until 1 300 objects of the classes Traffic Signs and Traffic Lights are contained in the data set, respectively. Additionally, to extend the classes with more obstacles, vehicles and people, freely available datasets were added. An example of this new dataset can be seen in Fig. 4. Fig. 5 shows a diagram representing the set of objects in the dataset. The resulting dataset contains 51 classes for traffic signs and traffic lights, and 12 classes for people, obstacles, and vehicles. It consists of 15 000 photos with a total of 170 000 labeled objects. A new detection model was trained on this dataset for 150 epochs. During this phase of data collection, models were trained in parallel to find training parameters for more effective model training.

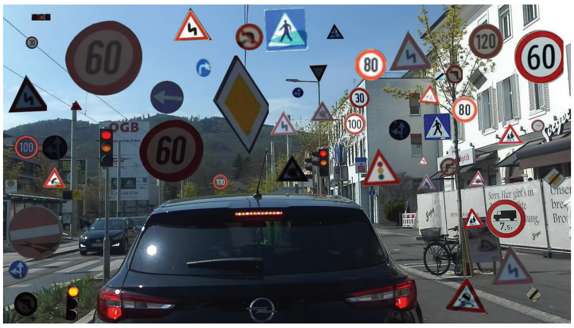


Figure 4: Example of an extended dataset

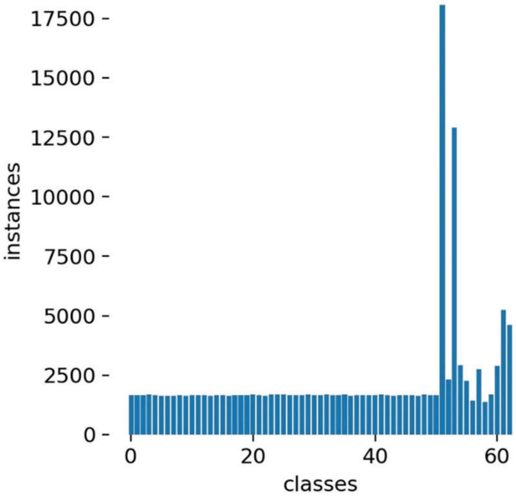


Figure 5: Amount of labeled objects in the dataset

A total of 40 models were trained and after approximately 1 000 hours of model training, the best training parameters were selected to train the model with the final dataset. This dataset was tested with a new test set and reached an *mAP-0.5* of over 90%, as depicted in Fig. 6. This result corresponds to the previously mentioned necessary safety in the automotive field. However, the result was only achieved with images that were recorded during the day in sunshine or slightly cloudy weather. Unfortunately, no validation of the result was performed in real driving situations or non-ideal conditions, such as rain or fog. Finally, the output data of the object detection is transferred to the distance calculation.

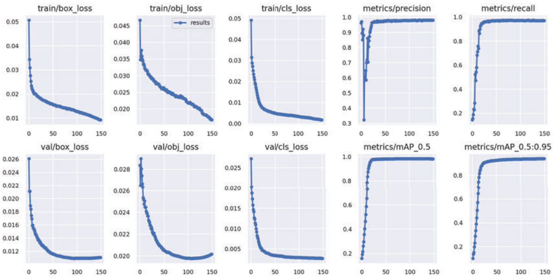


Figure 6: Result of trained detection model

This result corresponds to the previously mentioned necessary safety in the automotive field. However, the result was only achieved with images that were recorded during the day in sunshine or slightly cloudy weather. Unfortunately, no validation of the result was performed in real driving situations or non-ideal conditions, such as rain or fog. Finally, the output data of the object detection is transferred to the distance calculation.

## 2.2 Lane Detection

In order for a vehicle to stay in lane, a lane detection system is indispensable. The lane detection software of this project is based on an open source code, provided by the Australian roboticist Addison Sears-Collins [SC]. The software can detect white and yellow lane markings, and calculates the radii of their curvature. This method is carried out in a continuous loop and always works according to the same pattern. The process starts as soon as an image is passed on to the lane detection module. Then, various filters and functions are used to isolate the possible lanes in the image from interfering factors. For example, a function checks whether the value of the image channel of a pixel is above a certain threshold. If the value of this pixel is above the threshold, the color code *white* is assigned to it. Pixels whose image channel is below the threshold are declared as *black*. Afterwards, the two largest accumulations of *white* pixels in the image are searched for. For each accumulation, a polynomial function, which represents the detected lane, is calculated and displayed on the image. Finally, from these polynomials, i.e. the lanes, the radius of curvature is calculated. The driveable area, hence, the area between the detected lanes, is then displayed in the graphical user interface - see Fig. 7.

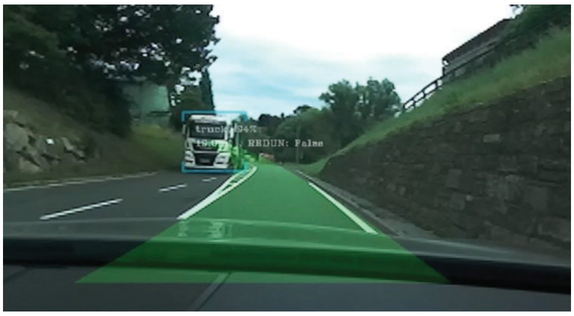


Figure 7: Lane detection: detected lanes are displayed in green

## 3 Distance to Objects Estimation

Although distance estimation is less important as object detection, it is useful in some cases, especially in emergency situations. Often, the distance to objects is detected with time-of-flight sensors, lidar or ultrasound. In our work, the stereo camera for the object detection is also used for distance estimation. The *ZED 2i* stereo camera is equipped with two lenses and associated sensors with a base distance of 12 cm. According to the manufacturer it is possible to measure distances up to 40 m with these cameras [Ste22]. Furthermore, this camera has a pre-trained neural network that can recognize and segment people. This data can be used to generate skeleton tracking. For the application in this work, these features were deactivated and the neural network specified for object recognition was used. To measure distance using the stereo camera the following three redundant systems are used: A stereoscopic measurement with object detection, a stereoscopic measurement without object detection, and a trigonometric measurement with object detection. Stereoscopic measurements have been carried out using the depth map. A depth map is generated with the base distance of the two camera sensors. This map consists of many points in a three-dimensional space, each of which is located at a calculated distance from the camera. This function is provided by the *ZED* manufacturer. An example depth map with 3D data representation is depicted in Fig. 8.

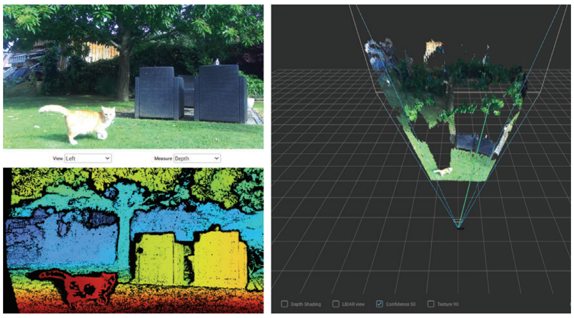


Figure 8: Depth map for stereoscopic measurements

To further validate the accuracy of the measurement, a defined test procedure was used to check the deviation. The test procedure ran as follows: An object was positioned at a defined distance and compared with the calculated distance from the system. Both resolutions which were also used in the object detection, VGA and 720p, were examined.

From these test measurements, the percentage deviation relative to the real distance was calculated. The diagram for the range from 1 m to 40 m can be seen in Fig. 9. The stereoscopic distance calculation without object detection runs only via the depth map and detects an undefined object which is at a critical distance. If such an object is detected, a warning is issued in the GUI and passed to the assistant systems. Finally, the third system uses the knowledge about an object, for example the standardized size of a traffic sign, and the knowledge about the camera sensor and the lens to calculate a distance to the object using the lens equation. This was also validated with the test procedure. The result can be seen in Fig. 10. In the range of 4 m to 40 m a deviation below 20% was achieved. In the next step, the two systems which work with object detection were compared. An 18% deviation in the range of 2 m to 30 m is depicted in Fig. 11. Based on this finding, the critical distance for a warning, or emergency braking, is set to below 2 m in the software. This warning is also transferred to the emergency braking assistant. In the range below 10 m, the stereoscopic calculation with object detection is in the GUI, for larger distances the results of the trigonometric calculations are provided. Both systems run in parallel, are compared in the background and in case of a deviation of more than 20% a warning is displayed in the GUI.

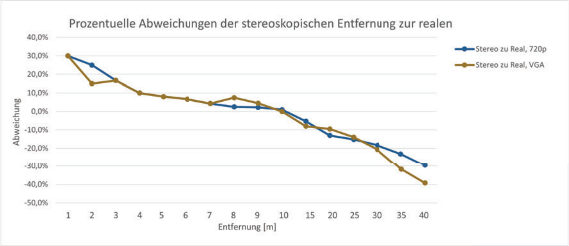


Figure 9: Percentage deviations of the calculated trigonometric distance (depth map)

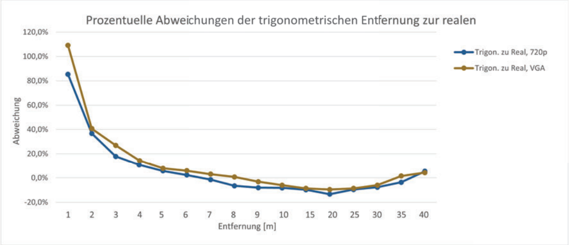


Figure 10: Percentage deviations of the trigonometric distance (object detection)

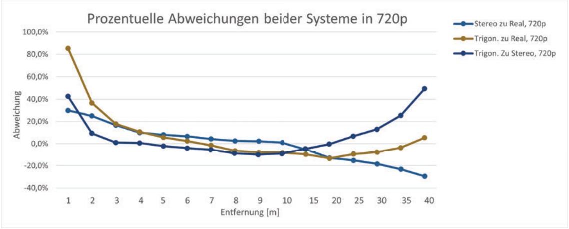


Figure 11: Percentage deviations of trigonometric to stereoscopic to real distance at 720p

# 4 Assistance Systems

## 4.1 Autonomous Emergency Braking System

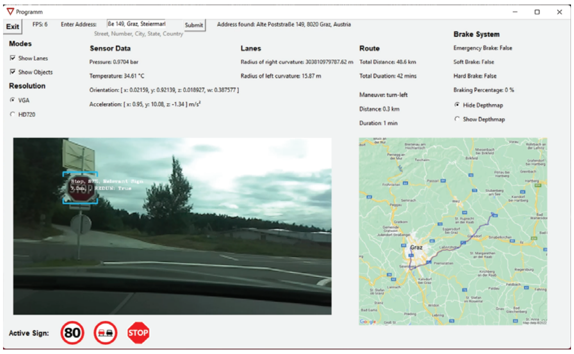
Many safety systems are needed in self-driving applications, such as a braking assistant. For this project, an emergency braking assistant and a system for foresighted braking were programmed. The dynamic data for the system comes partly from the object detection as well as from the distance detection. Static information about the vehicle is entered beforehand. With the help of distance detection, a relative speed can be calculated. The acceleration can be requested from the stereo camera's internal sensors. A velocity can be calculated via the GNSS. From the static data a maximum possible deceleration is calculated. Firstly, the system compares the relative speed and the theoretical time until contact with the object with the maximum possible deceleration. From this, a proportional braking command can be calculated to bring the vehicle to the same speed as the object with a safe distance between them. The data is shown in the GUI. If the distance measurement gives a warning for emergency braking, the braking command is overwritten and set to 100%. At this stage of the development the braking assistance system cannot distinguish between its own lane and the oncoming lane. Furthermore, the system has not been checked for accuracy and errors.

## 4.2 Traffic Signal Assistance System

The traffic sign and traffic light detection systems are also part of the assistance systems. When one of the 51 different traffic objects is detected, it is assigned to either the traffic sign or traffic light system. The first processes the traffic signs as defined by the law. For example, a vehicle must be stopped at a stop sign or an end of speed sign can override a speed sign. Additionally, the rotation of the sign is checked to see if it is also facing the camera. All active signs are displayed in the GUI as pictograms. An example can be seen in Fig. 12. The second subsystem deals with the traffic lights and determines if the vehicle has to be brought to a stop before it reaches the light and which light is relevant for the own vehicle. While the color of the traffic light is easy to distinguish, there are difficulties with the position of the traffic light. The problem lies here in the situations with several traffic lights as well as different kinds of traffic lights. With regard to horizontal and/or vertical traffic lights, these can be easily processed. But if the situation is extended to sidewalks and overhead traffic lights, there is a problem due to the distance to the center of the lane. Here, a correction to the center of the lane must be calculated using the lens equation. Due to the known distance to the traffic light and the defined focal length of the wide-angle lens, a correction to the center of the road strip can be calculated by means of trigonometric laws.

### 4.3 Navigation System

For the navigation system a GNSS receiver obtains the current location via satellites. The *Navilock NL- 8002U GNSS mouse* was selected for this purpose [Nav14]. With this device, the coordinates of the current location are obtained and then transmitted via a serial interface to the end device, in this case a laptop. For the navigation the APIs of the *Google Maps Platform* were used [Pla]. Thus, functions and services of Google Maps, for instance the route planning, can be integrated into the program. The process behind the satellite navigation is as follows: First, a destination has to be entered in the graphical user interface. After this, the *Google Maps Places API* searches for the coordinates of the entered address by using the *Autocomplete* function. Then, the current location, given in latitudes and longitudes, is queried via the GNSS mouse and passed on to an API. With this data, the API creates a route between the current location and the desired address. Additionally, navigational information including the route descriptions, the duration, the maneuvers and the distance of the individual route segments as well as the distance and the duration of the entire trip are output. The current location of the vehicle is then constantly compared with the coordinates of the route segments. If the coordinates of the vehicle's location match the coordinates of the end point of a route section, a maneuver - for example, *turn right* - is initiated. Then the cycle repeats until the end address is reached. The route description in the static map of the route, see Fig. 12, are simultaneously displayed in the GUI.



FPS: 6

Enter Address: ße 149, Graz, Steiermarl

Address found: Alte Poststraße 149, 8020 Graz, Austria

☑ Show Lanes

☑ Show Objects

● VGA

○ HD720

Pressure: 0.9704 bar

Temperature: 34.61 °C

Orientation: [ x: 0.02159, y: 0.92139, z: 0.018927, w: 0.387577 ]

Acceleration: [ x: 0.95, y: 10.08, z: -1.34 ] m/s²

Radius of right curvature: 303810979787.62 m

Radius of left curvature: 15.87 m

Total Distance: 48.6 km

Total Duration: 42 mins

Maneuver: turn-left

Distance: 0.3 km

Duration: 1 min

Emergency Brake: False

Soft Brake: False

Hard Brake: False

Braking Percentage: 0 %

● Hide Depthmap

○ Show Depthmap

Figure 12: GUI of the autonomous driving software

## 5 Summary

For the autonomous driving software, a data set was created to construct an object detection model. This data set was continuously optimized and updated. In the end, a model was developed that is capable of detecting traffic signs, traffic lights, vehicles, people and obstacles. Only traffic signs relevant to the vehicle – i.e., signs that are not too inclined or signs that are in their own lane – are taken into account. This model was integrated into the subsequently programmed software, so that the aforementioned objects can be detected by means of the stereo camera and displayed. In addition to the visual output, the camera generates sensor data – like acceleration, position, orientation, temperature, and pressure – which can be read out and displayed. Furthermore, a lane detection software was programmed to detect and filter lanes relevant for the vehicle. To ensure a redundant distance detection, the values of two depth detection calculations were compared and read out. In addition, an emergency braking assistant was programmed to simulate braking if the vehicle approaches an object too quickly. The functionality has been tested under simulated as well as real-world conditions in different environments. The software also contains a navigation system, which determines the directions from the current location to the selectable destination and displays them on a map. Finally, a graphical user interface was created, which provides and displays the detected objects and driving lanes, the sensor data, the navigation system and the data of the brake assistant. The results of this work serve as the basis for two further Bachelor's theses, which are currently being carried out, in which the track and object recognition is to be applied in a real road vehicle.

## Acknowledgements

The authors would like to thank Günter Bischof, Annette Casey, Florian Diernberger, Markus Eggbauer, Patrick A. Fritz, and Oliver Hofer for their support.

## References

[A$^+$16] M. Abadi et al. *TensorFlow: A system for large-scale machine learning.* 12$^{th}$ USENIX Symposium on Operating Systems Design and Implementation (OSDI 16), USENIX Association, pp. 265-283, 2016.

[DF20] Florian Diernberger and Patrick A. Fritz. *Autonom fahrendes Modellfahrzeug : Realisierung einer Spurhalte- und Objekterkennungssoftware mittels maschinellem Lernen.* Bachelor's thesis. Graz : FH JOANNEUM, Institute of Automotive Engineering, 2020.

[DF21] Florian Diernberger and Patrick A. Fritz. *Autonomously Driving Model Vehicle, Trends Challenges in Automotive Engineering.* 2$^{nd}$ ASCENT Conference - Engineering the Future, 2021.

[Dic02] Ernst D. Dickmanns. *The development of machine vision for road vehicles in the last decade, Intelligent Vehicle Symposium, IEEE Vol.1.* pp. 268-281, 2002.

[EH21] Markus Eggbauer and Oliver Hofer. *Autonom fahrendes Modellfahrzeug mit erweiterter Sensorik : Realisierung einer satellitenunterstützten Navigation.* Bachelor's thesis. Graz : FH JOANNEUM, Institute of Automotive Engineering, 2021.

[EH22] Markus Eggbauer and Oliver Hofer. *Autonomously Driving Model Vehicle.* 2$^{nd}$ UNITED-SAIG International Conference, 2022.

[HP22] Marlena V. Hofer and Johannes Pieber. *Software zur Realisierung eines autonom fahrenden Fahrzeugs für den realen Fahrbetrieb.* Bachelor's thesis. Graz : FH JOANNEUM, Institute of Automotive Engineering, pp. 140-148, 2022.

[Nav14] Navilock. *NL-8002U u-blox 8 multi GNSS USB receiver 1.50 m Operation manual (62523).* 2014.

[Pla] Google Maps Platform. Build awesome apps with Google's knowledge of the real world, howpublished = Website: https://developers.google.com/maps?hl=en. last access: 07.04.2023.

[SC] Addison Sears-Collins. The Ultimate Guide to Real-Time Lane Detection Using OpenCV. Website: `https://automaticaddison.com/`, note = last access: 22.6.2022.

[Ste22] Stereolabs. *ZED 2i Camera and SDK Overview [Version Februar 2022] data sheet.* 2022.

[W$^+$85] Richard Wallace et al. *First results in robot road-following.* JCAI'85 Proceedings of the 9 th International Joint Conference on Artificial Intelligence, 1985.

# Using AI to transform Traffic Models into Mobility Models

Georg Jäger[1]

[1]University of Graz, Institute for Environmental Systems Sciences, Merangasse 18, 8010 Graz, Austria

## Abstract

An important application of traffic models is to understand the origin of emissions and optimise traffic systems to be more sustainable. However, many models focus on car traffic and require the amount of car traffic from an external source. This has a huge disadvantage: Choosing not to use a private car for a trip has the biggest potential to reduce emissions, yet mode choice is a blind spot of most models. In this paper, we explore the possibility of using trained decision trees to simulate mode choice in a way that is easy to include in existing models. We investigate the performance of this approach for different types of traffic and conclude that this method, while possible for all traffic models, works best for microscopic agent-based traffic models. The trained decision trees used for this study are openly available to be used by other researchers or other interested parties.

## 1 Introduction

In order to combat climate change, reducing global CO2 emissions is paramount. Currently there is no sector in which the emissions reduce fast enough in order to reach the climate goal of keeping global warming below 2 degrees and in may sectors, the emissions even increase [OP$^+$17, SAC$^+$21, LGX19]. In the transport sector, however, we even observe the fastest growth [WG19]. This makes evident, that this sector is in dire need of significant innovation and improvement and not just minor adjustments to technological solutions or transport systems [EYYG20, ZFDH18, Sol19].

An important tool to generate knowledge regarding transportation systems and resulting emissions is the use of traffic models. There are many different ways of designing a traffic model, ranging from microscopic car-following model like SUMO [LBBW$^+$18] or VISSIM [FV10], over mesoscopic solutions [HJF18], to macroscopic models [HHST01]. While these models are capable of predicting the routes each car takes and based on those trajectories can calculate resulting emissions, they are focused on the use of private cars. Recently, more and more emphasis is put on mobility models in order to find solutions to the problem of increasing traffic emissions. In contrast to traffic models, they also incorporate the choice of mode and the choice of whether to even make a trip or not. Especially for questions related to sustainability, this makes sense: While green light optimal speed advisory systems can save up to 11.5% of emissions [EHG13] or using a green wave might lead to 5% less emissions [BJ21], choosing active mobility or CO2-neutral public transport options effectively reduces the emissions to zero.

A significant downside of mobility models is that they are more difficult to implement. Traditional traffic models need to find shortest paths between two points and include technical details about cars, engines and the transport system. These challenges can become complicated, but in principle it is clear how to solve them, since methods from engineering or network science can be used. For mobility models, the situation is much more complex: Especially mode choice is difficult to model, since it depends on many different factors, that can be different for each person and there is no simple theory that can explain it.

This problem is not unique to mobility science: In any field where human decision making plays a role, predicting those decisions is paramount, yet traditional agent-based models that rely on a fixed set of rules, quickly reach their limits. A recent solution to this kind of problem is to replace the rules by neural networks [Jäg19, Jäg21, JR22] and let a machine learning algorithm take care of the decision making process. In particular for mode choice problems there are many examples where machine learning was used successfully [TXZ15, ZYZW16, GSDG$^{+}$21, LLP13]. In this regard, training a decision tree [NAPS11] has been established as the dominant method.

Thus, using machine learning, an (agent-based) traffic model could in principle be expanded to a mobility model, by adding a trained decision tree that handles the mode choice of the agents. However, depending on the used model, the availability of information, in machine learning terms the *features* of an agent or trip, varies drastically. While a microscopic model has information about the trip reason for each person, a macroscopic model might not. For the knowledge about the closest public transport hub a spatial model is required. In general, each model will have access to different features and it is not clear how well suited this set of features will be, for it to be useful for predicting mode choice.

This study will explore the idea of transforming a traffic model into a mobility model by use of machine learning. Furthermore it will investigate different sets of features (available to different types of models) and show how good the resulting predictions are. The trained decision trees for each set of features and a script that can calculate mode choice probabilities according to them is available as open source software and can be used to augment any traffic or mobility model. (See `https://github.com/georgjaeger/mcp`)

The paper is organised as follows: Section 2 will shortly introduce the idea of solving classification problems using machine learning. After that, the process of training the decision trees for mode choice is explained in more detail. In Section 3 we will use the trained decision trees to predict the mode choice of people and compare to the actual results. Section 4 concludes by discussing the general merit of this approach and how well it is suited to which kind of models.

## 2 Methods

The problem of mode choice is a typical example of a classification problem [BNN$^{+}$00, MS57]. The goal here is to classify a given trip into categories representing the mode of transport, e.g. bike or public transport. This decision is based on certain information about the trip, the so-called features, e.g. the length of the trip or the age of the person performing the trip. Various types of machine learning are capable of solving such classification problems, like hidden layer neural networks [QLL$^{+}$16, HCB00], support vector machines [DL10, SV99, LM01, XGW13, BP12], or nearest neighbors classification [CH67, WS09]. For the problem of mode choice, however, decision trees [SY15, BPKR14] are the most commonly used algorithm.

A decision tree is an optimized set of true/false questions relating to the features that enables us to find the most likely class for any set of information. There are many algorithms of finding such an optimal set of decision sets [PARS13] and in this study we use the implementation provided by the Python package SciKit Learn [PVG$^{+}$11].

Figure 1 shows part of an decision tree used to predict mode choice. Based on the answer of the question in the first layer (in this example: Is the length of the trip shorter than a certain threshold?) different follow up questions occur. With each answered question, the algorithm determines the probability of each category.

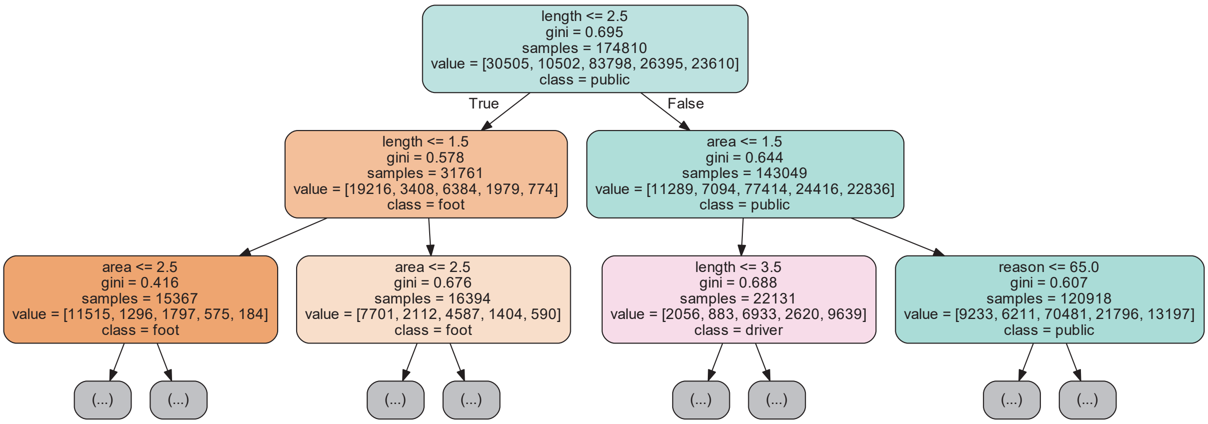


Figure 1: Part of a decision tree used to predict mode choice

For the problem at hand we will use the following categories:

- foot
- bike
- public transport
- passenger
- car driver

There are different feature sets we will investigate, but they all come from the same data source. We use data from an Austrian mobility survey called Österreich Unterwegs [THS+16]. In this survey, more than 18000 respondents described their mobility routines (distances, reasons, means of transport, . . . ) in a standardized way. This provides enough information to train a decision tree. However, not all reported information should be used, since we only want to use information that is available in the traffic model in question. This of course depends on the type of traffic model, so we will investigate three different sets of features, each tailored to be used in one type of traffic model.

***Feature set Mini*** is a minimalist feature set that only includes three features, namely the length of the trip, the area the trip is performed in and the reason of the trip. Thus it can be used in most traffic models.

***Feature set Macro*** is designed for macroscopic traffic models. The focus here is data that could be gathered from map data or inferred from other statistics about a certain region (e.g. employment rate etc.).

***Feature set Micro*** uses the most features and is intended for microscopic models that can resolve individual people. In addition to all features used by Macro it also includes details like car availability.

Table 1 shows all features and which feature set uses them.

Table 1: Available features and their utilisation in the three compared feature sets

| | Mini | Macro | Micro |
|---:|:---:|:---:|:---:|
| trip length | X | X | X |
| area | X | X | X |
| age | | X | X |
| economic situation | | X | X |
| job | | X | X |
| distance to public transport | | X | X |
| trip reason | X | | X |
| car availability | | | X |
| gender | | | X |

This is the meaning of each feature presented in Table 1:

**Trip length:** The length of the trip categorized into the bins using the following bin limits: 0.5 km, 1 km, 2.5 km, 5 km, 10 km, 20 km, 50 km.

**Area:** The area type, in which the trip takes place. We separate into large cities (> 1 million inhabitants), cities (> 100 000 inhabitants), central areas, and peripheral areas.

**Age:** The age of the person making the trip, using the following categories: 6-14 years, 15-19 year, 20-24 years, 25-34 years, 35-44 years, 45-54 years, 55-64 years and ≤ 65 years.

**Economic situation:** The economic situation of the household categorized as very bad, bad, average, good, or very good.

**Job:** Information about whether the person making the trip is currently employed or not. The used categories are pupil, employed, retired and unemployed/other.

**Distance to public transport:** The distance to the closest station of any kind of public transport, measured in minutes by foot. Categories are ≤ 5, 6-15, 16-30, 31-60, 61-120, and ≥ 121

**Trip reason:** The main reason for the trip. Used categories are: Commute to work, business, school/education, taking somebody somewhere, shopping, private activity, other leisure, private meeting and other reason.

**Car availability:** The availability of a car for that person. It can either be always, sometimes or never.

**Gender:** The gender of the person taking the trip.

All three feature sets are then used individually to train a decision tree for mode choice. The results of these predictions are shown in the following section.

## 3 Results

In order to test the accuracy of the predictions, we will use two test groups. The first test group is picked by selecting random entries from the database in a way that produces a sample that is representative of the whole data set. This is called the Representative Group. However, in this case a representative sample is not the best choice for evaluation. Even the most basic statistical method would perform well in predicting the mode choice of a representative group, since it would be sufficient to know the distribution of modes over the whole data set. For this reason, a second test group is needed, the so-called Homogeneous Group. Here, the data base entries are selected from a certain sub-group that has mobility behaviour that is different from the average. In this example, we use female inhabitants of a large city between 20 and 24 years, however, any subgroup could be used for this analysis. A well trained decision tree should be able to predict these decisions correctly.

The plots of this section show how well the predictions of the decision tree match the real mode choices. This information is presented in the following way: A bar plot shows the relative share of each mode: foot (green), bike (green-yellow), public transport (turquoise), passenger (yellow) and car driver (red). The left half of the bar (lighter shade) shows the real choices reported in the survey. The right half of the bar (darker shade) shows the predictions by the decision tree. This means that for a perfect fit, the colors of the left and right halves of the bar should change at the same time. In order to gain more insight, such bars were generated for each trip length class, since the mode shares are very dependent on this feature.

As a benchmark, Figure 2 shows the results of a decision tree that does not use a whole feature set, but rather only the feature *trip length* as sole basis for the choice. Thus, most of the predictions of individual trips will be wrong. However, as mentioned above, when making predictions about the Representative Group, these errors will average out and the overall shares of the trips will be more or less correct. This is visible on the left panel of Figure 2. The right panel, reporting on the results for the Homogeneous Group gives us more information: Here we

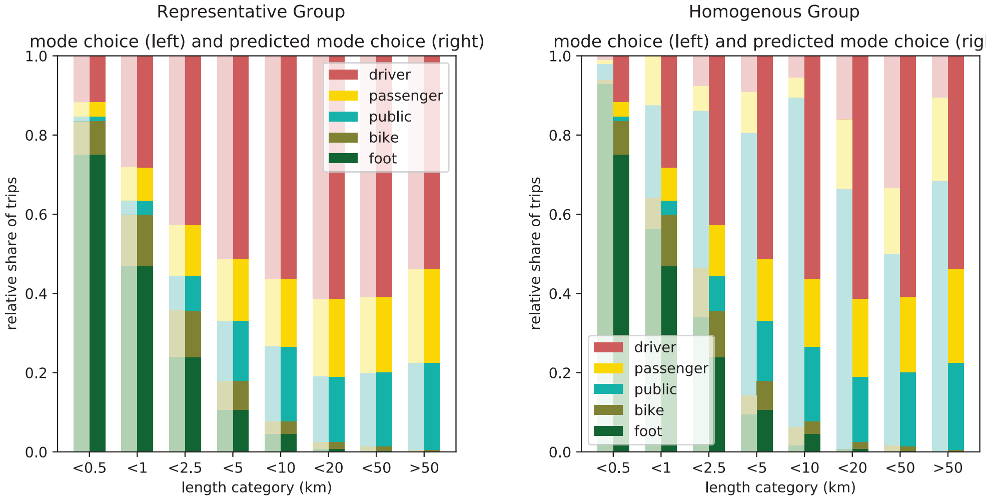


Figure 2: Just using the trip distance only works on average, but fails to predict the decisions of the Homogeneous Group

see that most of the predictions fail: Car use is overestimated, while public transport and active mobility is underestimated. Thus we conclude, that using just the trip length as a basis for mode choice gives an incomplete picture.

Next we will investigate the feature sets described above and see how the predictions improve. Results for the Representative Group are omitted, since they always show a perfect fit. The predictions for the Homogeneous Group using the feature set Mini, Macro and Micro are given in Figure 3.

The feature set Mini adds the area and the trip reason and thus significantly improves the prediction for the Homogeneous Group. It is still rather inaccurate, but public transport becomes more relevant, while car use decreases.

Even more information is provided in the feature set Macro. In the middle panel of Figure 3 we see that the predictions become significantly more accurate. For most distance classes the differences between prediction and real decision are below 5%. Especially the sum of driver and passenger (red and yellow, the top two segments), as well as the sum of foot and bike (green and green-yellow, the bottom two segments) fit well.

Using microscopic information further enhances this result, as shown in the right panel of Figure 3. Except for small deviations, even the Homogeneous Group is accurately depicted.

## 4 Discussion

In this study we investigated the possibility of using Machine Learning to transform traffic models into mobility models. We found that using a trained decision tree to perform the decision which mode of transport is suitable for a given trip is a simple way of including mode choice in already existing models.

However, the choice of which features to use for training and prediction is not a simple task. Mainly it depends on which information is available. We investigated three different sets of features: A minimalist approach, a set suitable for macroscopic models and a set that can be used by microscopic models. We found that the predictions for a representative group were correct in all examples if one aggregates on the whole group. Individual decisions, however, might be predicted incorrectly.

To gain more insight we also tested the trained decision trees on a more homogeneous group.

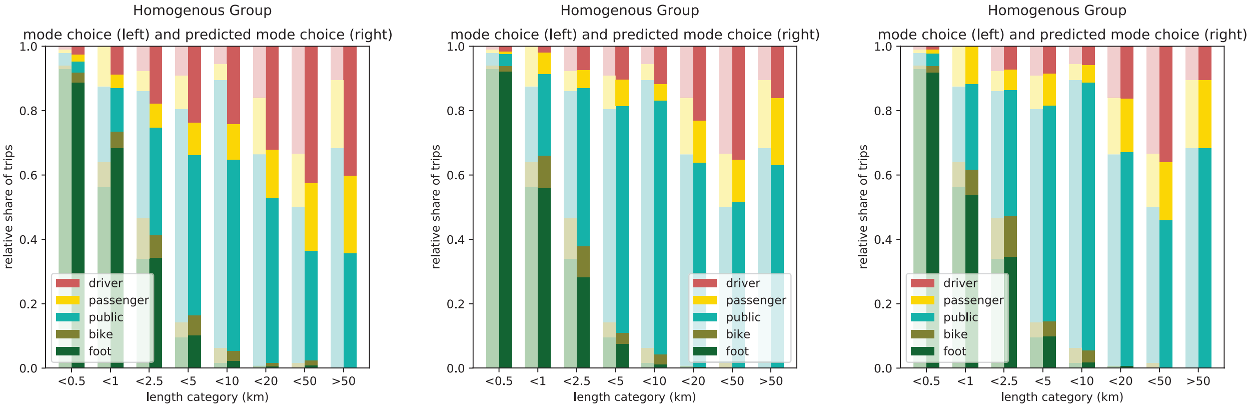


Figure 3: Prediction results of the feature set Mini (left), Macro (middle) and Micro (right)

This task is more difficult for a decision tree, since it cannot simply rely on global averages and probabilities, but must accurately judge each trip based on the available information. While the feature set Mini offered just minor improvements to the benchmark, in which we only used global probabilities based on the trip length, the new information provided by the feature set Macro drastically enhanced this result. The order of magnitude of the resulting error is in line with usual errors in most traffic models, which means that such an expansion could be used without drastically loosing accuracy. The best result was obtained for the feature set Micro. Here, the inaccuracies are minimal and show great potential to augment an existing traffic model in a simple way.

This brings us to the following conclusion: The suitability of trained decision trees to predict mode choice for traffic models heavily depends on the model that is used and its purpose. If one is mainly interested in the overall probabilities of each mode given its length, every feature set would work. However, for the set Mini it seems that individual trips are not predicted correctly and it offers little benefit over just using statistical probabilities. If one is interested in individual choices, the feature set Macro already provides satisfying results (see middle panel of Figure 3) and only uses information that is available on a macroscopic scale. However, if microscopic information about individual trips or people is available, the feature set Micro can be used, further enhancing the predictions (see right panel of Figure 3).

The trained decision trees used for this analysis (Mini, Macro and Micro) are available via `https://github.com/georgjaeger/mcp`, together with an example script that shows how they are used. They are licensed under an MIT license and thus can be used by anyone to augment existing models or perform other studies.

Including mode choice in traffic models is an important step towards a more holistic understanding of our transport systems, which is necessary in order to solve the major problems that are caused or amplified by transportation. This study provides a simple way to do so and can thus hopefully contribute to a more complete picture of our traffic systems and mobility behaviour.

## References

[BJ21] Elisabeth Bloder and Georg Jäger. Is the green wave really green? the risks of rebound effects when implementing "green" policies. *Sustainability*, 13(10):5411, 2021.

[BNN+00] Jan G Bazan, Hung Son Nguyen, Sinh Hoa Nguyen, Piotr Synak, and Jakub Wróblewski. Rough set algorithms in classification problem. In *Rough set methods and applications*, pages 49–88. Springer, 2000.

[BP12] Himani Bhavsar and Mahesh H Panchal. A review on support vector machine for data classification. *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)*, 1(10):185–189, 2012.

[BPKR14] Mr Brijain, R Patel, MR Kushik, and K Rana. A survey on decision tree algorithm for classification. 2014.

[CH67] Thomas Cover and Peter Hart. Nearest neighbor pattern classification. *IEEE transactions on information theory*, 13(1):21–27, 1967.

[DL10] K SRIVASTAVA Durgesh and B Lekha. Data classification using support vector machine. *Journal of theoretical and applied information technology*, 12(1):1–7, 2010.

[EHG13] David Eckhoff, Bastian Halmos, and Reinhard German. Potentials and limitations of green light optimal speed advisory systems. In *2013 IEEE Vehicular Networking Conference*, pages 103–110. IEEE, 2013.

[EYYG20] Seyfettin Erdoğan, Seda Yıldırım, Durmuş Çağrı Yıldırım, and Ayfer Gedikli. The effects of innovation on sectoral carbon emissions: evidence from g20 countries. *Journal of environmental management*, 267:110637, 2020.

[FV10] Martin Fellendorf and Peter Vortisch. Microscopic traffic flow simulator vissim. In *Fundamentals of traffic simulation*, pages 63–93. Springer, 2010.

[GSDG$^+$21] IHV Gue, J Soliman, M De Guzman, R Cabredo, A Fillone, NS Lopez, and JBM Biona. Decision tree analysis of commuter mode choice in baguio city, philippines. In *IOP Conference Series: Materials Science and Engineering*, volume 1109, page 012059. IOP Publishing, 2021.

[HCB00] Guang-Bin Huang, Yan-Qiu Chen, and Haroon A Babri. Classification ability of single hidden layer feedforward neural networks. *IEEE Transactions on Neural Networks*, 11(3):799–801, 2000.

[HHST01] Dirk Helbing, Ansgar Hennecke, Vladimir Shvetsov, and Martin Treiber. Master: macroscopic traffic simulation based on a gas-kinetic, non-local traffic model. *Transportation Research Part B: Methodological*, 35(2):183–211, 2001.

[HJF18] Christian Hofer, Georg Jäger, and Manfred Füllsack. Large scale simulation of CO2 emissions caused by urban car traffic: an agent-based network approach. *Journal of Cleaner Production*, 183:1–10, 2018.

[Jäg19] Georg Jäger. Replacing rules by neural networks a framework for agent-based modelling. *Big Data and Cognitive Computing*, 3(4):51, 2019.

[Jäg21] Georg Jäger. Using neural networks for a universal framework for agent-based models. *Mathematical and Computer Modelling of Dynamical Systems*, 27(1):162–178, 2021.

[JR22] Georg Jäger and Daniel Reisinger. Can we replicate real human behaviour using artificial neural networks? *Mathematical and Computer Modelling of Dynamical Systems*, 28(1):95–109, 2022.

[LBBW+18] Pablo Alvarez Lopez, Michael Behrisch, Laura Bieker-Walz, Jakob Erdmann, Yun-Pang Flötteröd, Robert Hilbrich, Leonhard Lücken, Johannes Rummel, Peter Wagner, and Evamarie Wießner. Microscopic traffic simulation using sumo. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 2575–2582. IEEE, 2018.

[LGX19] Dunnan Liu, Xiaodan Guo, and Bowen Xiao. What causes growth of global greenhouse gas emissions? evidence from 40 countries. *Science of The Total Environment*, 661:750–766, 2019.

[LLP13] Tingyang Li, Xin LUAN, and Zhenghong Peng. Application of traffic mode choice model based on decision tree algorithm. *Engineering Journal of Wuhan University*, 3, 2013.

[LM01] Yuh-Jye Lee and Olvi L Mangasarian. Ssvm: A smooth support vector machine for classification. *Computational optimization and Applications*, 20(1):5–22, 2001.

[MS57] Charles D Michener and Robert R Sokal. A quantitative approach to a problem in classification. *Evolution*, 11(2):130–162, 1957.

[NAPS11] Arundhati Navada, Aamir Nizam Ansari, Siddharth Patil, and Balwant A Sonkamble. Overview of use of decision tree algorithms in machine learning. In *2011 IEEE control and system graduate research colloquium*, pages 37–42. IEEE, 2011.

[OP+17] Jos GJ Olivier, Jeroen AHW Peters, et al. *Trends in global CO2 and total greenhouse gas emissions: 2017 report.* PBL Netherlands Environmental Assessment Agency The Hague, 2017.

[PARS13] Anuja Priyam, GR Abhijeeta, Anju Rathee, and Saurabh Srivastava. Comparative analysis of decision tree classification algorithms. *International Journal of current engineering and technology*, 3(2):334–337, 2013.

[PVG+11] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830, 2011.

[QLL+16] Bo-Yang Qu, BF Lang, Jing J Liang, A Kai Qin, and Oscar D Crisalle. Two-hidden-layer extreme learning machine for regression and classification. *Neurocomputing*, 175:826–834, 2016.

[SAC+21] Isak Stoddard, Kevin Anderson, Stuart Capstick, Wim Carton, Joanna Depledge, Keri Facer, Clair Gough, Frederic Hache, Claire Hoolohan, Martin Hultman, et al. Three decades of climate mitigation: Why haven't we bent the global emissions curve? *Annual Review of Environment and Resources*, 46, 2021.

[Sol19] Saeed Solaymani. Co2 emissions patterns in 7 top carbon emitter economies: the case of transport sector. *Energy*, 168:989–1001, 2019.

[SV99] Johan AK Suykens and Joos Vandewalle. Least squares support vector machine classifiers. *Neural processing letters*, 9(3):293–300, 1999.

[SY15] Yan-Yan Song and LU Ying. Decision tree methods: applications for classification and prediction. *Shanghai archives of psychiatry*, 27(2):130, 2015.

[THS+16] Rupert Tomschy, Max Herry, Gerd Sammer, Roman Klementschitz, Sebastian Riegler, Robert Follmer, Dana Gruschwitz, Felix Josef, Stefan Gensasz, Roman Kirnbauer, and others. Österreich unterwegs 2013/2014: Ergebnisbericht zur österreichweiten Mobilitätserhebung Österreich unterwegs 2013/2014. 2016.

[TXZ15] Liang Tang, Chenfeng Xiong, and Lei Zhang. Decision tree method for modeling travel mode switching in a dynamic behavioral process. *Transportation Planning and Technology*, 38(8):833–850, 2015.

[WG19] Shiying Wang and Mengpin Ge. Everything you need to know about the fastest-growing source of global emissions: Transport. 2019.

[WS09] Kilian Q Weinberger and Lawrence K Saul. Distance metric learning for large margin nearest neighbor classification. *Journal of machine learning research*, 10(2), 2009.

[XGW13] Yitian Xu, Rui Guo, and Laisheng Wang. A twin multi-class classification support vector machine. *Cognitive computation*, 5(4):580–588, 2013.

[ZFDH18] Runsen Zhang, Shinichiro Fujimori, Hancheng Dai, and Tatsuya Hanaoka. Contribution of the transport sector to climate change mitigation: insights from a global passenger transport model coupled with a computable general equilibrium model. *Applied energy*, 211:76–88, 2018.

[ZYZW16] Guangjun Zhan, Xuedong Yan, Shanjiang Zhu, and Yun Wang. Using hierarchical tree-based regression model to examine university student travel frequency and mode choice patterns in china. *Transport Policy*, 45:55–65, 2016.

PASO
SOLUTIONS
Business Analytics - Software Solutions - IT Consulting
EXPLORE
A NEW WAY
OF WORK

# Scientific Computing 2023

## Track: Wirtschaft

*Thomas Strametz, Fatima Jammoul, Joachim Schauer und Wilhelm Zugaj:* **Comparing different Sales Forecast approaches on real-world data using Machine Learning** ........ 190

*Wilhelm Zugaj, Fatima Jammoul, Joachim Schauer und Thomas Strametz:* **Image classification in the real world: Methodologies for hierarchical classification of customer provided images in e-commerce** ........ 198

*Raven Adam und Marie Kogler:* **Tracking the Evolution of Climate Protection Discourse in Austrian Newspapers: A Comparative Study of BERTopic and Dynamic Top Modeling** ........ 208

# Comparing different Sales Forecast approaches on real-world data using Machine Learning

Fatima Jammoul[1], Joachim Schauer[1], Thomas Strametz[1,⋆], and Wilhelm Zugaj[1]

[1]FH JOANNEUM – University of Applied Sciences, Software Design and Security, Werk-VI-Strasse 46, 8605 Kapfenberg, Austria
⋆corresponding author

## Abstract

Time series based regression forecasts arise in numerous practical applications and are frequently solved by using machine learning. Our concrete problem is situated in the brewing industry. The goal is to perform an accurate sales forecast for specific customers and products. For this reason, two prominent machine learning modeling approaches (i.e., Deep Learning and Gradient Boosting) are systematically compared in order to determine the better performing one. Additionally, the impact of individual predictors is investigated by testing different combinations of input features. The results show that Deep Learning outperforms Gradient Boosting on our real-world data set. Furthermore, the best performing predictions make use of similar feature combinations.

## 1 Introduction

Accurate sales forecasts significantly facilitate a variety of planning activities for companies [BSB$^+$19, PAA$^+$22], including, but certainly not limited to, logistics decisions, shift schedules and pricing strategies [HA18]. Despite the importance of this topic, conducted sales forecasts often prove to be inaccurate and error-prone due to lacking data quality, manual calculations or inappropriately chosen modeling techniques [LOE00]. Machine learning methods have shown to provide efficient and robust sales forecasts using a variety of different techniques [see, e.g., VPSOMS$^+$22, Pav19]. The goal of this work is to systematically compare two particular state of the art machine learning methods in the context of sales forecasting, namely Deep Learning [GBC18] with long short-term memory (LSTM) [HS97] or Gated Recurring Unit (GRU) [CGCB14] cells and Gradient Boosting using xgboost [CG16].

One particular challenge in dealing with real-world time series data is the fact that subsequent observations are by no means independent. An appropriately chosen machine learning method needs to be able to accurately represent and learn the underlying trend structure to perform forecasts. Deep Learning methods with recurrent layers, also referred to as recurrent neural networks (RNNs), are a prominent choice in this context, as the recurrent nature is designed to handle sequences of data [Smy20, ZL17]. Gradient Boosting is considered a prominent alternative to Deep Learning [see, e.g., Wan19, CGCdS20].

To demonstrate and compare model performance in terms of forecast accuracy, we apply these two methods to real-world data provided by an Austrian brewery. Besides comparing those different modeling techniques and the corresponding hyperparameters, multiple different feature combinations are used to find not only the optimal modeling approach, but also the ideal input parameters. The results of our research show, that RNNs generally outperform gradient-boosted trees with respect to forecast accuracy. The optimal predictors depend on the customer and also differ from product to product as each customer-product combination has dedicated models, but the ideal input parameters show similarities across all models.

## 2 Methodology

The conducted research was separated into three major stages. In the first step we performed an extensive data exploration in order to identify promising features at an early stage. Afterwards, we produced sales forecasts using different modeling techniques and varying input features. In the final stage, the conducted sales forecasts are compared in terms of forecasting accuracy.

Let for a given time series $(X_i)_{i=0}^{N}$, $\hat{X}_i$ denote the estimated 4-step forecasting value of observation $X_i$. Then the forecasting accuracy is defined as in Equation (1).

$$FCA = 1 - \frac{\sum\limits_{i=n+1}^{N} |\hat{X}_i - X_i|}{\sum\limits_{i=n+1}^{N} X_i} \tag{1}$$

To this end, the model is trained on the first $n$ observations. The final out-of-sample observations $X_{n+1}, \ldots, X_N$ are used to evaluate the forecasting performance. Typically, the out-of-sample observations account for about 20% of the available data.

### 2.1 Available features

In the following, we consider a real-world data set comprised of sales information for an Austrian brewery in a time frame of eight years. Given the data is available on a weekly basis, the time series contains approximately 420 observations for each customer and article. The goal is to provide an accurate weekly sales forecast (corresponding to the response variable *sales in hectolitre [hl]*) using the available features as described in Table 1. As is common in time series problems, using lagged values as features in forecasting models allows to account for a certain type of saisonality. In order to include these structures in our models, we infer the variables *weekly plan*, *previous sales* and *last year sales* using existing features. A potentially important feature includes information about sales promotions, coded into *special promotion* and *num of (ordinary) promotions.* It is important to emphasize that the existence of a special promotion is inherently independent of the number of ordinary promotions. One customer may offer several ordinary promotions that include a certain product, while a special promotion may be offered additionally. As, unfortunately, no information about the precise price reduction is available, the available features only indicate the existence and number of such promotions.

At this stage, we wish to further emphasize that the data is inherently structured as a time series on a weekly grid. This means that all available features correspond to a specific calendar week in a specific year and to a specific customer-stock keeping unit (sku). A stock keeping unit denotes a certain type of drink including its packaging format and the number of litres (e.g., 24 cans of a specific soda 500 ml). Thus, a 4-step forward forecast corresponds to a prediction for the sales of a given sku at a particular customer of the following month. This is in tune with real life forecasting demands in a professional setting.

### 2.2 Data exploration

The main purpose of the data exploration phase was to identify highly influential features in the context of improving the quality of the forecast. In the initial phase, explorative plots were generated to visually identify possible correlations between an input feature and the response, given as the *sales* value of a given calendar week. For the sake of brevity we omit the generated plots, but briefly summarize the most important findings.

Initial conjecture suggested that the weekly planned sales (*weekly plan*) was one of the most promising input features, as, in an ideal world, they would closely approximate the actually observed sales. Surprisingly, the weekly plan

| **id** | **feature name** | **explanation** |
|---|---|---|
| 0 | year | year of the observation |
| 1 | week | calendar week of the observation |
| 2 | sales [hl] | sales in hl of the observation |
| 3 | weekly plan [hl] | derived as the planned sales over a certain period of time divided by the number of calendar weeks in that time period |
| 4 | special promotion | an indicator 0 or 1 if there is a special promotion in this calendar week |
| 5 | num of promotions | an integer value specifying how many ordinary promotions are in this calendar week |
| 6 | previous sales [hl] | the average sales in hl based on all historical sales values in this certain calendar week |
| 7 | last year sales [hl] | the sales in hl value exactly one year ago |

Table 1: Summary of the available features

very often differs strongly from the actual sales and proved to be highly inaccurate for many customer-sku combinations. *Special promotion* on the other hand, despite being given as a binary variable only, tends to accurately indicate sales peaks. A promotion of this magnitude appears rather infrequently compared to ordinary promotions. Here, we could also observe a difference between smaller and bigger customers. Indeed, for big customers, ordinary promotions occur almost every week, which makes them a rather poor choice for a predictor as they do not provide much additional information. For smaller customers, however, ordinary promotions prove to be a very good indicator on whether there are any sales at all in a specific week.

*Last year's sales* and *previous sales* do not show strong correlation with *sales* for most customer-sku combinations. This suggests that most customers do not place orders based on historical experience or habit, but rather on current demand.

### 2.3 Modeling techniques

There are many modeling techniques available which are used for the purpose of forecasts [see, e.g., HA18, FGJW19]. Deep Learning and Gradient Boosting Trees are two popular modeling approaches for conducting time-series based forecasts [MMM23]. Recurrent neural networks such as LSTM and GRU in particular are commonly used in this context [LZGX21]. Therefore, we implemented and compared five different network architectures which mostly incorporate recurrent layers with different hyperparameters (e.g., unit size, activation function, etc.). As a point of comparison, we also consider networks with densely connected layers. Furthermore, 21 different feature combinations were constructed using the available features of Table 1. Given that there are 30 relevant customer-sku combinations, we trained a total of 3150 setups for the Deep Learning methods.

As a point of comparison, we also considered Gradient Boosting methods. Here we used two different booster called `gbtree` which is the default tree-booster in xgboost and `dart` [VGB15] which adds a dropout to boosted-trees. In order to provide an appropriate comparison, we trained the Gradient Boosting models on the same set of feature and relevant customer-sku combinations. Therefore, the number of total Gradient Boosting training runs is given by 1260.

After completing the runs for each setup and each competing method, a ranking revealing which modeling technique in combination with which input features leads to the most accurate forecast for a certain customer-sku combination can be generated. For a subset of this ranking, we refer to Table 2. Please note that in a productive environment, each customer-sku combination would use a different model with different input features in order to retrieve the most accurate results.

# 3 Results

Table 2 presents a subset of the obtained modeling results and their achieved forecast accuracy. Only the best results for selected customer-sku combinations are included. The first two columns specify the anonymised customer-sku combination. The third column denotes the general modeling approach (deep learning or gradient boosting) and the adjacent column *model id* specifies the exact model used. As can be seen in Table 2, the best results were achieved using Deep Learning based on only two specific network architectures (see Listing 1 and 2). The feature id column determines which input parameters were used for the forecast. Each digit of the feature id can be mapped to a certain input parameter using Table 1. For example, feature id 1234 indicates that the features 1 = week, 2 = sales, 3 = weekly plan and 4 = special promotion were used as input parameters. The final column *fca* determines the resulting forecast accuracy calculated using Equation (1).

```
def get_model_v3(input_shape):
    model = Sequential()
    model.add(
        LSTM(128,
        input_shape=input_shape,
        activation='relu')
    )
    model.add(
        Dense(64,
        activation='relu')
    )
    model.add(Dense(1))
    return model
```

Listing 1: Deep Learning model m3 using Keras

| customer | sku | modeling approach | model id | feature ids | fca |
|---|---|---|---|---|---|
| 25 | 803137 | deep learning | m4 | 1234 | 46.26 % |
| 31 | 803138 | deep learning | m3 | 123 | 46.51 % |
| 3 | 803138 | deep learning | m4 | 12345 | 47.33 % |
| 31 | 803205 | deep learning | m4 | 1234 | 49.68 % |
| 31 | 800001 | deep learning | m3 | 12345 | 51.04 % |
| 3 | 800017 | deep learning | m4 | 1234 | 51.10 % |
| 3 | 800001 | deep learning | m4 | 1234 | 51.93 % |
| 3 | 803305 | deep learning | m4 | 1234 | 57.83 % |
| 3 | 800002 | deep learning | m4 | 1234 | 58.40 % |

Table 2: Highest forecast accuracy of selected customer-sku combinations including relevant parameters sorted by fca

```
def get_model_v4(input_shape):
    model = Sequential()
    model.add(
        GRU(128,
        input_shape=input_shape,
        activation='relu')
    )
    model.add(
        Dense(64,
        activation='relu')
    )
    model.add(Dense(1))
    return model
```

Listing 2: Deep Learning model m4 using Keras

# 4 Conclusion and Outlook

The results in Table 2 suggest that Deep Learning methods using the two particular network architectures with recurrent layers *m3* and *m4* lead to the best forecast accuracy using the given real-world data set. The superior model performance of RNNs is in line with current research [see, e.g., SNTSN18, LDD18, Cho18]. Gradient Boosting could not compete with recurrent neural networks in this scenario.

However, there is still room for improvements. Even though the found network architectures achieved acceptable results, it is likely that additional tuning of hyperparameters would further improve the forecasting accuracy. Tuning hyperparameters manually is considered to be a cumbersome task, therefore tools like Keras Tuner or AutoML provide the capability of heuristically optimizing hyperparameters automatically. Incorporating similar tools could further improve the performance of predicting sales in the future.

Currently, only features provided by the brewery were included in the prediction process. Taking external features such as meteorological factors into account could further enhance the results [see, e.g., ADMB19].

## Acknowledgments

This research was partly funded by the Federal Ministry for Digital and Economic Affairs of the Republic of Austria through the COIN project FIT4BA.

## References

[ADMB19] Jeremiah Anderson, Vijay Daultani, Tariq Muman, and Mohamed Batran. The importance of weather for e-commerce orders forecasting. In *Proceedings of the 2019 International Conference on E-Business and E-commerce Engineering*, EBEE'19, page 15–19, New York, NY, USA, 2019. Association for Computing Machinery.

[BSB$^+$19] Kasun Bandara, Peibei Shi, Christoph Bergmeir, Hansika Hewamalage, Quoc Tran, and Brian Seaman. Sales demand forecast in e-commerce using a long short-term memory neural network methodology. In *Neural Information Processing*, page 462–474, Berlin, Heidelberg, 2019. Springer-Verlag.

[CG16] Tianqi Chen and Carlos Guestrin. Xgboost. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Aug 2016.

[CGCB14] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling, 2014.

[CGCdS20] T. Cinto, A. L. S. Gradvohl, G. P. Coelho, and A. E. A. da Silva. Solar flare forecasting using time series and extreme gradient boosting ensembles. *Solar Physics*, 295(7):93, Jul 2020.

[Cho18] François Chollet. *Deep Learning mit Python und Keras.* mitp Professional. MITP, Frechen, Germany, 2018 edition, May 2018.

[FGJW19] Christos Faloutsos, Jan Gasthaus, Tim Januschowski, and Yuyang Wang. Classical and contemporary approaches to big time series forecasting. In *Proceedings of the 2019 International Conference on Management of Data*, SIGMOD '19, page 2042–2047, New York, NY, USA, 2019. Association for Computing Machinery.

[GBC18] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep*

*Learning. Das umfassende Handbuch.* mitp Professional. MITP, Frechen, Germany, 2018 edition, October 2018.

[HA18] Rob J. Hyndman and George Athanasopoulos. *Forecasting: principles and practice.* OTexts, 2018.

[HS97] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, 1997.

[LDD18] Jesus Lago, Fjo De Ridder, and Bart De Schutter. Forecasting spot electricity prices: Deep learning approaches and empirical comparison of traditional algorithms. *Applied Energy*, 221:386–405, 2018.

[LOE00] Michael Lawrence, Marcus O'Connor, and Bob Edmundson. A field study of sales forecasting accuracy and processes. *European Journal of Operational Research*, 122(1):151–160, 2000.

[LZGX21] Zhenyu Liu, Zhengtong Zhu, Jing Gao, and Cheng Xu. Forecast methods for time series data: A survey. *IEEE Access*, 9:91896–91912, 2021.

[MMM23] Ricardo P. Masini, Marcelo C. Medeiros, and Eduardo F. Mendes. Machine learning advances for time series forecasting. *Journal of economic surveys*, 37(1):76–111, 2023.

[PAA$^+$22] Fotios Petropoulos, Daniele Apiletti, Vassilios Assimakopoulos, Mohamed Zied Babai, Devon K. Barrow, Souhaib Ben Taieb, Christoph Bergmeir, Ricardo J. Bessa, Jakub Bijak, John E. Boylan, et al. Forecasting: theory and practice. *International Journal of Forecasting*, 2022.

[Pav19] Bohdan M. Pavlyshenko. Machine-learning models for sales time series forecasting. *Data*, 4(1), 2019.

[Smy20] Slawek Smyl. A hybrid method of exponential smoothing and recurrent neural networks for time series forecasting. *International Journal of Forecasting*, 36(1):75–85, 2020. M4 Competition.

[SNTSN18] Sima Siami-Namini, Neda Tavakoli, and Akbar Siami Namin. A comparison of arima and lstm in forecasting time series. In *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 1394–1401, 2018.

[VGB15] Rashmi Korlakai Vinayak and Ran Gilad-Bachrach. Dart: Dropouts meet multiple additive regression trees. In *Artificial Intelligence and Statistics*, pages 489–497. PMLR, 2015.

[VPSOMS$^+$22] Iván Vallés-Pérez, Emilio Soria-Olivas, Marcelino Martínez-Sober, Antonio J. Serrano-López, Juan Gómez-Sanchís, and Fernando Mateo. Approaching sales forecasting using recurrent neural networks and transformers. *Expert Systems with Applications*, 201:116993, 2022.

[Wan19] Karan Wanchoo. Retail demand forecasting: a comparison between deep neural network and gradient boosting method for univariate time series. In *2019*

*IEEE 5th International Conference for Convergence in Technology (I2CT)*, pages 1–5, 2019.

[ZL17] Lingxue Zhu and Nikolay Laptev. Deep and confident prediction for time series at uber. *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*, Nov 2017.

# Image classification in the real world: Methodologies for hierarchical classification of customer provided images in e-commerce

Fatima Jammoul[1], Joachim Schauer[1], Thomas Strametz[1], and Wilhelm Zugaj[1,⋆]

[1]FH JOANNEUM – University of Applied Sciences, Software Design and Security, Werk-VI-Strasse 46, 8605
Kapfenberg, Austria
⋆corresponding author

## Abstract

Image classification is a classic machine learning problem, having found numerous practical applications in the past. Our problem is situated in the field of agricultural e-commerce. For a collection of customer created images of agricultural machinery, we want to detect category, brand, type and series. This enables an automatic deduction of technical product information.

The model we propose makes use of hierarchical classification. First, each uploaded image is pre-screened to determine the depicted product category. In case an image is identified as a tractor, the next model hierarchy level detects the specific brand. The last model in the hierarchy tries to deduce the type and series of a tractor.

The paper details the methodologies employed in this project, including ideas on the detection of unsatisfactory images that must be avoided for model training as well as suggestions on how to train and validate a hierarchical classification model. Finally, we argue that as a by-product, this approach can also be used for detecting fraudulent image submissions.

## 1 Introduction

In the realm of machine learning problems, image classification plays a central role in diverse fields ranging from the detection of anomalies in medical scans [SMR+19, HPHS19] to plant identification [PPW21]. The problem at hand usually constitutes classifying a large set of images into pre-defined categories. The employed methods range from classic principal component analysis [KSS13], linear discriminant analysis [LZO06], support vector machines [CGLRML20] to, arguably most popular, convolutional neural networks [LSZ22, TL19, RW17, LLY+18, SJM18, Gér19].

Hierarchical modelling approaches are often used in the context of highly complex data, where classification between different subgroups can be of varying difficulty. It is of course simpler to distinguish, e.g., a horse from a tractor, than it is to classify different types of tractors. To take this into account, it can be beneficial to identify a hierarchical structure in the input by separating the data into more general groups in a first layer, before more finely classifying each group. For example, one could split the data into animals and vehicles in a first step, before further identifying tractor types from the vehicle group. This underlying tree-like structure allows for an adjustment in the penalization of the model, as it is natural to penalize a failure to distinguish at a coarse level of the hierarchy more severely than a misclassification at a finer level.

One possible way to reflect the hierarchical structure in the training process is to adjust the loss function. An example of this can be seen in [WDD+18], which details the classification of images related to breast cancer. While the authors employ a classic “flat” convolutional neural network (CNN), they choose to use a hierarchical loss function instead of cross-entropy. The images are to be classified at two levels into invasive or *in situ* (in case of a carcinoma) and benign or normal (in case of a noncarcinoma). By using a hierarchical loss in the training stage, not recognizing a carcinoma is penalized more severely than not being able to distinguish between an invasive or in situ carcinoma. It can be seen that using this hierarchical loss function provides an improvement in model performance compared to non-hierarchical loss.

On the other hand, it seems reasonable to implement the hierarchy into the model architecture directly by training models for each layer. [YZP$^+$15] provide an overview of this topic using CNNs. They implement a hierarchical deep convolution neural network (HD-CNN) to take the hierarchical structure of the data into account. This hierarchy is divided into a coarse layer and a fine layer, where the former provides a classification into general groups and the latter a fine classification within each group. It is of particular interest that the estimated fine category prediction for each image is obtained by using the probabilities for the coarse layer groups obtained in the first step as weight. Furthermore, they describe methods to learn the category hierarchy from the training data in the first place and include the possibility of non-disjoint coarse categories.

This type of modelling approach is highly flexible and can be easily adjusted to the problem at hand, e.g., in certain problems only a subset of classes obtained in the coarse layer need to be classified further in subsequent layers. This can be seen in [KSE$^+$20], where hierarchical machine learning models are used to classify images of tissue biopsies of children in an effort to identify illnesses causing stunted growth. In the first layer, the samples are classified into Environmental Enteropathy, Celiac's Disease and a control group. In the next step, a classification model is trained to determine the severity of Celiac's Disease in 4 categories. The authors evoke tree imagery by referring to the initial classifier as *the parent level model* and the subsequent severity classifier as the *child level model*. The basic architecture makes use of CNNs by passing pre-processed images through three convolutional and subsequent pooling layers. The output layer provides a classification into three (parent layer) or four (child layer) classes. Using these techniques, the authors obtained either competitive or improved precision and recall scores compared to alternative methods.

A related application in e-commerce is presented in [SS19], where clothing items are classified in a three-layer hierarchical model. Here, the model architecture follows VGGNet as introduced by [SZ15], which consists of either 16 or 19 convolutional and fully connected layers separated into blocks and connected by filters. The proposed adaptation in [SS19] classifies through two "coarse" layers by splitting into clothing and goods in the first layer and into tops, bottoms, dresses, shoes and similar categories in the second layer. In this application, the second and third block include these coarse classification outputs. A fine classification is achieved in the final step of the hierarchy, where specific types of shoes (e.g., ankleboots and sandals) and clothes (e.g., T-shirts, shirts, sweaters) make up the output layer of the network. The overall loss is computed using different weights for the two coarse and one fine layer output, with a strong emphasis on the former layers in the first dozen epochs. Compared to base VGGNet models, the hierarchical method achieves lower loss and higher accuracy in both test and training data.

Our problem is situated in the field of e-commerce as well. In the following chapter, we will provide a concise problem statement and argue why the use of a hierarchical network architecture provides an intuitive approach to handle complex image classification. Our methods show similarity to the discussed work, but still provide an interesting contrast due to the depth of the proposed hierarchy and use of competitive pre-trained models that allow for an efficient and effective image classification.

# 2 Modelling

## 2.1 Problem Statement

The problem at hand concerns an application in the field of agricultural e-commerce. The website landwirt.com is an established marketplace, where numerous products linked to agriculture may be sold and bought by independent users. While the available commodities may range from fruit trees to actual farm animals, the lion's share of the purchases on the website may be attributed to pre-owned large agricultural appliances.

In the following, we focus on tractor sales in particular, since they are by far the most important product group of the platform and integral to numerous agricultural pursuits. In 2022 alone, a little over 4500 standard tractors were registered in Austria [CLA22]. As is common in user-based e-commerce, an advert for an item is created by the seller, who will assign a product category (from a list of categories), provide pictures (see, e.g, Figure 1 and 2), enter the technical data of their products (e.g., horsepower, dimensions, age, brand, series, model, etc.), describe the product and subsequently set the price for potential buyers to consider.

This kind of application has to deal with two challenges. First, sellers may make mistakes when entering the technical data of their products, leading to customer confusion and complaints. Second, some products are advertised in the wrong product category. Some sellers do this intentionally, since selling products in some categories is free, while for others a fee has to be paid. Thus, in order to mitigate the risk of human error or possible fraud in this process, an automatic classification of the provided images is needed.

Therefore, our goal is to implement a robust machine learning model that pre-screens the images to check whether they indeed depict standard tractors and, in that case, identifies the brand and, subsequently, series and model type. Integrating this model into the platform of landwirt.com enables a new process of advertising that checks whether the chosen product category is correct for a manually uploaded image and automatically fills out the technical data of the corresponding product from a database.

Due to the inherent hierarchical structure of the problem, we consider a hierarchical classification model. This allows for a clear division between more coarse levels (standard vs. non-standard tractor and tractor brands) and a fine-tuned classification for model type and series. As is common practice, we choose a CNN method to classify the provided images, in particular the EfficientNet-$B0$ and EfficientNet-$B4$ networks as introduced in [TL19]. These state of the art methods implement a compound scaling method by simultaneously scaling network depth (number of layers), network width (number of channels per layer) and image resolution. This scaling approach provides a general and flexible network that improves accuracy and efficiency compared to simply scaling one such attribute. The baseline model $B0$ already provides improvements in accuracy and number of parameters compared to competitive models such as ResNet when comparing performance on the ImageNet database [see Table 2 in TL19]. Further sequential scaling of the base network yields the models $B1 - B7$. For our purposes, we use the baseline model $B0$ for a proof of concept and a scaled model $B4$ for the final modelling in almost all layers because of better resulting accuracy while remaining within the limitations of our available hardware.

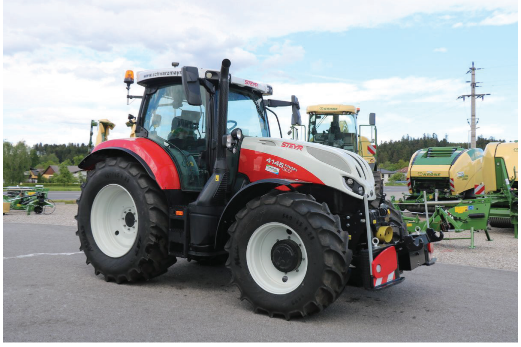


Figure 1: Example of an image used in validation, depicting a Steyr tractor 4145 Profi CVT.

## 2.2 Pre-Labelling: Adding additional regular images for training

The product hierarchy of the commercial platform consists of several categories at its top level. In the context of our work, we focus on the product type *tractor* and are interested in further sub classification of this category after the top level category has been determined.

The category *tractor* consists of several subcategories. For our application, the category *standard tractor* is the most important one and contains about 94 % of all tractors. Additional subcategories include *small tractors*, *special tractors*, *tractors for fruit growing* and several more. Finally, *standard tractor* is classified into several *brands* which are then further divided into *series and model.*

As in most machine learning applications, the quality and quantity of the training data available is crucial to the success of the method. Initially, there are 2882301 images available for training and evaluation of our models, which were extracted from user-made advertisements of landwirt.com. The images are labeled as primary or secondary image. A primary image represents the main image that is displayed for the product. These images typically show the product completely visible from the outside. Secondary images consist of additional images that might display arbitrary parts of the product. Furthermore, the images have been labelled to specify category, subcategory, brand and model of the product. We use lists provided by the industry partner that assign models to series to expand the model label to a *series and model* label.

There are several challenges associated with the image data, especially since it varies in several different ways. One particular variation of the training data concerns interior shots of the tractors. Indeed, many images in the data focus on the interior of the tractor (see Figure 2), which of course are not suitable to train a model for exterior shots. There is no available label or other information in the data to indicate that an image is an interior shot. Thus, we require some means to separate these images from the desired training data. A conservative approach would be to only use the primary image provided for each advertisement. This, however, would reduce the training-set significantly.

As an alternative, we perform an initial *pre-labelling step* in order to separate unwanted interior shots from exterior shots. This step is achieved by using a pretrained EfficientNet-$B0$ model as implemented in `EfficientNet-Pytorch` with its default configuration [Mel20, PGM$^{+}$19]. This highly performative CNN model has been pre-trained on the classic ImageNet dataset [DDS$^{+}$09], which contains over 1.2 million images separated into 1000 classes, one of which are *tractors*. The strategy is to perform a prediction for each given image and to extract the prediction score for the class *tractor*. Images with a prediction score higher than a certain threshold are subsequently included in the training data.

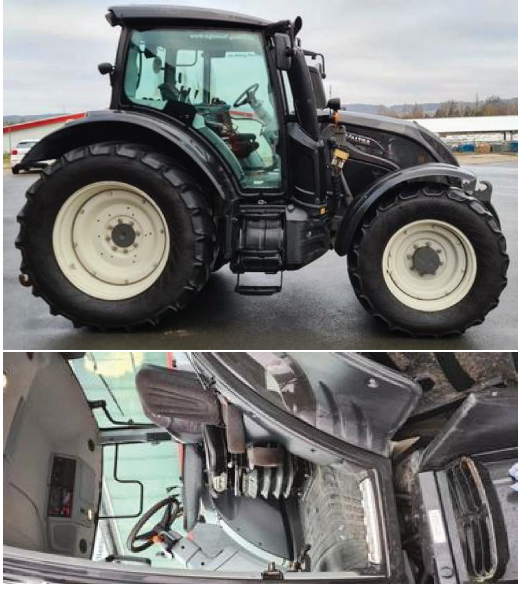


Figure 2: Exterior (top) and interior (bottom) images of a Valtra tractor. Pre-labelling scores for the class *tractor* are at about 85% (top) and 5% (bottom), i.e., only the top image is included in the training.

Another challenge is related to the pre-labelling step itself. Non-standard tractors like tractors for fruit growing look very different than standard tractors, thus pre-labelling does not recognize them as tractors. As a consequence, pre-labelling is applied to standard tractors only.

Even though we retain the use of some secondary images due to the pre-labelling step, a third challenge is found in the massively reduced number of images when the hierarchy reaches the bottom level of *series and model.* The reason for this reduction is threefold. First, there is a data quality problem. About one third of the data is missing model or series information. Second, pre-labelling reveals that less than half of all images are regular views from the exterior. And finally, the standard tractors break up into a lot of brands which in turn contain a lot of different *series and model* types leading to few representatives per leaf-node in the hierarchy. To retain a reasonable number of training data, we have to define a minimum threshold of 250 images for a *series and model* to be eligible for detection.

A few numbers give an impression. Of all available images, 915065 are labeled as *standard tractors* and only 305430 of those have a valid *series and model* and pass the pre-labelling threshold. Extracting all images belonging to a *series and model* category that contains at least 250 images, results in 9 brands with a total of 35168 images only. As a solution, hierarchical image detection stops at the level of brand detection for those brands that consist of *series and model* classes that do not contain a sufficient number of images.

## 2.3 Model Architecture

The root of our hierarchy is an EfficientNet-$B0$ trained to recognize the general category of all images. As the most important project goal is to identify the brand and further specifications for standard tractors, the next layer of the model distinguishes — for tractors only — specific tractor types (e.g. *standard tractor*, *small tractor*, ...) from each other.

To train this layer, we fine-tune a pre-trained EfficientNet-$B4$. We split between training data, validation data and test data in a $80-10-10$ ratio. The neural network is fine-tuned by training it to classify images into *standard tractor* or *non-standard tractor.* Since pre-labelling does not work for *non-standard tractors* (they are generally not detected as tractors by a pre-trained EfficientNet in the first place), pre-labels are omitted from this training step.

The two subsequent layers of the hierarchy are again implemented by fine-tuning a pre-trained EfficientNet-$B4$ model to recognize tractor *brands* and *series and model*, respectively. To that end, consider that putting the model and series of a tractor into a hierarchical order may collapse the need for classification in some cases. Indeed, in the given dataset multiple sets of series only have one model associated to it, i.e., many series-nodes would only have one model-leaf in the hierarchical tree. Thus, it makes sense to consider all series and model combinations as belonging to the same, final, hierarchical layer. We refer to Figure 3 for an illustration of the final hierarchical structure of the model. To train these two layers with a sufficient number of images, secondary images with valid pre-labels are permitted. On the other hand, all *series and model* containing less than 250 images have to be removed from the hierarchy to assure a minimum accuracy of 80%, as desired by the industry partner.

We use the same suitable hyperparameters to fine-tune the pre-trained EfficientNet-$B4$ for the bottom three hierarchy layers. Training is executed with a batch size of 32 images, and a learning rate of 0.001 for optimizer Adam [see, e.g., Gér19]. In a data augmentation stage, the dataset is initially resized to 224x224 pixel images. Furthermore, with a probability of 50%, images are horizontally flipped or have their brightness and contrast adjusted mildly. The determination of these settings is done heuristically.

## 2.4 Results

In this section we will briefly summarize the predictive accuracies for the hierarchical model as determined in the training phase. The classification in the layer separating *standard tractors*

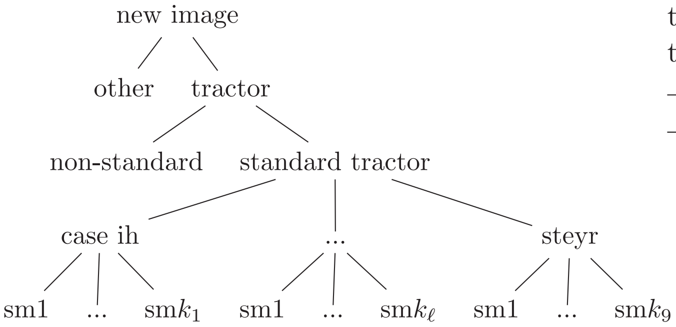


Figure 3: Schematic of the 4-layer hierarchical model pipeline used to classify an incoming image. The first layer separates tractors from non-tractors with an ImageNet-pre-trained model. The second layer splits images into standard and non-standard tractors, with the third layer identifying the tractor brand for standard tractors. The final layer identifies series ($s$) and model ($m$) of the tractor, with $k_\ell$ denoting the total number of such subclasses for the $\ell$-th tractor brand, $\ell = 1, \ldots, 9$.

from *non-standard tractors* as well as the classification by the subsequent layer providing the specific tractor *brands*, both achieve an accuracy of approximately 97%.

Table 1 provides an overview of the number of images used for training each brand for the classification of its *series and model*, and the accuracy resulting from testing that classification for each brand. As previously mentioned, the minimum accuracy specified by landwirt.com is 80 %, which means that the series and model of all brands but Lindner can be determined with sufficient accuracy. There is no significant difference in the number of images available for more accurately determined brands like Fendt compared to Lindner as can be seen in Table 2. We conjecture that the different models of Lindner are more similar to each other than those of Fendt, as there are several *series and model* combinations in Lindner that share the same *series* (602, 624). For Fendt, only 2 such combinations share the same series (205). Thus, a close distinction between models that share the same series may be more error-prone. A detailled investigation of this hypothesis is, however, beyond the scope of this article.

| *brand* | #images | accuracy |
|---|---:|---:|
| jcb | 355 | 1 |
| case-ih | 881 | 0,98 |
| valtra | 565 | 0,96 |
| fendt | 3317 | 0,88 |
| john-deere | 15253 | 0,86 |
| new-holland | 2611 | 0,86 |
| claas | 2768 | 0,84 |
| steyr | 6532 | 0,80 |
| lindner | 2886 | 0,78 |

Table 1: The number of images available for each brand to train, validate and test the prediction of *series and model* together with the accuracy achieved at testing

| *brand* | *series and model* | #images |
|---|---:|---:|
| fendt | 205-302 | 687 |
| fendt | 205-4771 | 356 |
| fendt | 33-3908 | 502 |
| fendt | 353-6212 | 276 |
| fendt | 363-2315 | 479 |
| fendt | 792-9516 | 263 |
| fendt | 798-9631 | 754 |
| | | |
| lindner | 602-7787 | 563 |
| lindner | 602-7910 | 334 |
| lindner | 602-8010 | 447 |
| lindner | 602-827 | 255 |
| lindner | 623-8098 | 302 |
| lindner | 624-2330 | 382 |
| lindner | 624-2918 | 344 |
| lindner | 624-5612 | 259 |

Table 2: The number of images of each *series and model* of Fendt and Lindner used for training.

## 3 Conclusion

In this work we demonstrate a real life application of hierarchical convolution neural networks in the context of image recognition for agricultural e-commerce. We develop a simple and modular network hierarchy of EfficientNet models to sequentially classify images of vehicles in order to identify brand name, model type and series for standard tractors. Using EfficientNet models pre-trained on the ImageNet database, we successfully pre-label the provided images to build a homogeneous training set of exterior shots of the tractors. With the resulting data, we post-train the EfficientNet models to allow for recognition of brand names and further specifications. The resulting empirical accuracy exemplifies the excellent model performance.

## Acknowledgments

This research was partly funded by the Federal Ministry for Digital and Economic Affairs of the Republic of Austria through the COIN project FIT4BA.

## References

[CGLRML20] J. Cervantes, F. Garcia-Lamont, L. Rodríguez-Mazahua, and A. Lopez. A comprehensive survey on support vector machine classification: Applications, challenges and trends. *Neurocomputing*, 408:189–215, 2020.

[CLA22] WKO Club Landtechnik Austria. Zulassungsstatistik Standardtraktoren österreich. `https://www.wko.at/service/netzwerke/standardtraktoren-dezember-2022.pdf`, 2022. Accessed: 2023-01-26.

[DDS+09] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *IEEE Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2009.

[Gér19] A. Géron. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems.* O'Reilly Media, Inc., 2nd edition, 2019.

[HPHS19] R. B. Hegde, K. Prasad, H. Hebbar, and B. Singh. Comparison of traditional image processing and deep learning approaches for classification of white blood cells in peripheral blood smear images. *Biocybernetics and Biomedical Engineering*, 39(2):382–392, 2019.

[KSE+20] K. Kowsari, R. Sali, L. Ehsan, W. Adorno, A. Ali, S. Moore, B. Amadi, P. Kelly, S. Syed, and D. Brown. Hmic: Hierarchical medical image classification, a deep learning approach. *Information*, 11(6), 2020.

[KSS13] H. B. Kekre, T. K. Sarode, and J. K. Save. Classification of image database using independent principal component analysis. *International Journal of Advanced Computer Science and Applications*, 4(7), 2013.

[LLY+18] W. Luo, J. Li, J. Yang, W. Xu, and J. Zhang. Convolutional sparse autoencoders for image classification. *IEEE Transactions on Neural Networks and Learning Systems*, 29(7):3289–3294, 2018.

[LSZ22] Z. Liu, L. Sun, and Q. Zhang. High similarity image recognition and classification algorithm based on convolutional neural network. *Comput. Intell. Neurosci.*, 2022:2836486, apr 2022.

[LZO06] T. Li, S. Zhu, and M. Ogihara. Using discriminant analysis for multi-class classification: an experimental investigation. *Knowledge and Information Systems*, 10(4):453–472, Nov 2006.

[Mel20] L. Melas. Efficientnet-pytorch. `https://github.com/lukemelas/EfficientNet-PyTorch`, March 2020. Version 0.7.1.

[PGM+19] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[PPW21] J. Pärtel, M. Pärtel, and J. Wäldchen. Plant image identification application demonstrates high accuracy in Northern Europe. *AoB PLANTS*, 13(4), 07 2021.

[RW17] W. Rawat and Z. Wang. Deep convolutional neural networks for image classification: A comprehensive review. *Neural Computation*, 29(9):2352–2449, 2017.

[SJM18] N. Sharma, V. Jain, and A. Mishra. An analysis of convolutional neural networks for image classification. *Procedia Computer Science*, 132:377–384, 2018. International Conference on Computational Intelligence and Data Science.

[SMR+19] L. Shen, L. R. Margolies, J. H. Rothstein, E. Fluder, R. McBride, and W. Sieh. Deep learning to improve breast cancer detection on screening mammography. *Scientific Reports*, 9(1):12495, 08 2019.

[SS19] Y. Seo and K. Shin. Hierarchical convolutional neural networks for fashion image classification. *Expert Systems with Applications*, 116:328–339, 2019.

[SZ15] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *3rd International Conference on Learning Representations (ICLR 2015)*, pages 1–14. Computational and Biological Learning Society, 2015.

[TL19] M. Tan and Q. Le. EfficientNet: Rethinking model scaling for convolutional neural networks. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 6105–6114. PMLR, 09–15 Jun 2019.

[WDD+18] Z. Wang, N. Dong, W. Dai, S. Rosario, and E. Xing. Classification of breast cancer histopathological images using convolutional neural networks with hierarchical loss and global pooling. In *Image Analysis and Recognition*, pages 745–753. Springer Cham, 06 2018.

[YZP+15] Z. Yan, H. Zhang, R. Piramuthu, V. Jagadeesh, D. DeCoste, W. Di, and Y. Yu. Hd-cnn: Hierarchical deep convolutional neural networks for large scale visual recognition. In *2015 IEEE International Conference on Computer*

*Vision (ICCV)*, pages 2740–2748, 2015.

# Tracking the Evolution of Climate Protection Discourse in Austrian Newspapers: A Comparative Study of BERTopic and Dynamic Topic Modeling

Raven Adam[1], Marie Kogler[1]

Institute of Environmental System Sciences, University of Graz, Merangasse 18/I 8010 Graz, Austria

10.04.2023

## Abstract

This study focuses on the media discourse surrounding climate protection in Austrian newspapers. Two topic modeling methods, Dynamic Topic Model (DTM) and BERTopic, are compared to investigate topic evolution, i.e. changes in the salient words of a topic over time. While both approaches are able to create evolving topics, their respective architectures and approaches are significantly different: BERTopic utilizes word embeddings to account for the context of words while DTM is based on latent word distributions where word occurrence order is irrelevant. Additionally, BERTopic assigns each document only one topic, while documents in DTM are assigned a mixture of topics. The study uses a self-collected corpus of 22,101 articles acquired from the database WISO, which were pre-processed and cleaned thoroughly. The results show that BERTopic and DTM produce comparable topics, but several differences need to be addressed. First, salient words in BERTopic are more specific and less general in regard to the overall association of topic, e.g. encompass particular climate actions and events, while DTM features very broad terminologies. Second, there are cases where no comparable topics are found since the content of a topic of one algorithm can be integrated and partialized in several topics of the other algorithm. We conclude that while BERTopics allow better insights into event-based topic variability, DTM shows a better resolution of topic formation. The comparative nature of this work highlights the ambiguity of conventional evaluation metrics and the need for complementary approaches incorporating human evaluation methods to automated content analysis.

## 1 Introduction

30 years ago, the *greenhouse summer of 88* marked the first time climate change was covered in a significant fashion by global media [1]. Since then, it has remained a constant part of news coverage. In recent years, an increasing amount of attention has been given to climate protection efforts made by and expected from different parties, namely politics, industry and the general public [2]. As mass media plays a major role in presenting information to the general public and shaping their views, the way it represents climate protection and its impacts can significantly influence public behavior [3]. Due to the entanglement of news and society, research on climate reporting reflects on the public climate debate as a whole, which is associated with climate protection and sustainability goals [4].

Topic modelling methods have been shown to be powerful tools when investigating large amounts of textual information. Statistical topic models have been well tested in this regard and become a mainstay in automated content analysis [5]. New advances have led to methods that allow focused investigation into the discourse behavior over time as well as methods that do not rely on statistics and encode contextual information [6].

Over the last years, topic modelling  has become a frequently used method to investigate the media discourse surrounding climate change [7], with LDA [8] being the most common technique  [5], [9]. Most of this research focuses on the US and the major European countries, whereas smaller countries and especially the southern hemisphere remain underrepresented [10]. While most studies investigate climate change coverage over extended periods of time, due to limitations of LDA they mostly focus on changes in topic relevance over time, which is simply the percentage of articles associated with each topic [11] – [13]. Due to the static nature of LDA topic word representations are mainly shaped by periods with high coverage and describe periods of low coverage poorly. For example, a topic about activism might mostly consist of words related to *Fridays4Future*, which would only make sense for articles published from 2018 onwards. To the best of our knowledge, more advanced methods like Dynamic Topic Models (DTM) [14] or BERTopic [15] which are capable of capturing changes within topics and remedy this problem have not been applied to the field of climate reporting. In this paper we will explore the three following points of interest based on a self-collected corpus on climate protection coverage in Austrian newspapers:

- Assessment of the ability of evolutionary topics to account for changing points of interest within a corpus.
- Evaluation of how feasible topic representations from contextual topic models are for human interpretation compared to commonly used statistical topic models.
- Whether common topic evaluation metrics are applicable for contextual topic models and comparing them to statistical models.

# 2 Methodology

To investigate changes over time within the discourse surrounding climate protection, two different methods are employed. DTM as a baseline approach to which the second method BERTopic, which is a more recent advancement in the field of topic modeling, will be compared.

## 2.1 Dataset

The articles are acquired from the database WISO[1] using the search term **klimaschutz*.** The collection resulted in 39,082 articles from nine major Austrian newspapers[2]. To ensure high data quality, duplicate articles were removed, and pre-processing steps including i) normalization of different spellings for the same word ii) removal of non-alphabetic signs iii) adding white spaces between punctuation and words and iv) sentence and word tokenization were performed. Articles containing more than 250 words were selected while shorter articles were dropped since DTM is not optimized to analyze a mixture of very short and long documents and short documents produce additional noise, lowering the quality of the results [16]. After these steps, 22,101 articles remained for analysis.

In preparation for the DTM analysis, part-of-speech (POS) tagging and lemmatization [17] were applied to filter nouns, adjectives, and adverbs. Preparations for BERTopic analysis encompass the aforementioned steps i) - iv), except for word tokenization. By only applying sentence tokenization the contextual information of the data can be retained and interpreted by BERTopic.

## 2.2 Dynamic Topic Model

DTM was introduced by [14] and builds on the framework of LDA. The main difference between the methods is that LDA treats a corpus as an unordered collection of documents, whereas DTM considers the order of documents within the corpus as important information, resulting in an evolving set of topics over time [18].

In DTM and LDA each document is generated from a set of latent topics. These topics are represented by a probability distribution over all the words in the corpus, with some words being more representative of the topic than others. By iteratively adjusting these probability distributions, topic representations consisting of the most salient words, i.e. words with highest probability in a topic, are obtained. These representations can be investigated by researchers and domain experts acquire a descriptive analysis of the examined corpus [9]

Regarding DTM, documents are additionally grouped together into specific time sequences. The word probability distribution is adjusted for each sequence to better represent documents contained in each time sequence. Investigation of the changes in the likelihood of salient topic words allows researchers to derive specific points of interest relevant to a certain time sequence such as political events or influential persons [19].

In order to use DTM, it is necessary to determine the number of topics beforehand. To find the best number of topics, several LDA models (chunk_size = 2000, iterations = 400, passes = 20) were trained on a range of two to fifty topics for four different time sequences as well

[1] https://www.wiso-net.de/dosearch

[2] Die Kronen Zeitung (n = 2033), der Standard (n = 3923), der Kurier (n = 4028), die Presse (n = 3387), Profil (n = 445), die Wiener Zeitung (n = 2497), die Salzburger Nachrichten (n = 3218), oberösterreichische Nachrichten (n = 2189) and der Falter (n = 389)

as the entire corpus. The models were evaluated using the $C_V$ [20] coherence measure. The basic idea is that the $C_V$ scores of a LDA model achieve local peaks for topic numbers that provide informative topic representations [21], [22]. The numbers of topics that result in shared or close peaks between the $C_V$ scores of each time sequence and the entire corpus are indicative of the number of topics for which the DTM will provide informative results [23]. For the respective climate mitigation corpus, the best results were achieved with 25, 35 and 47 number of topics. Using these local maxima three DTMs were trained. After thorough manual examination of quality and informativeness of the results, the variant with 35 topics was chosen.

### 2.3 BERTopic

BERTopic is a novel approach for topic modeling that leverages recent advances in embedding models. This approach stems from Bidirectional Encoder Representations from Transformers (BERT) [24]. Here, documents are represented as points in a high-dimensional vector space, where each coordinate contains contextual information about the document. This method generates a clustering of semantically similar documents. Dimensionality reduction [25] and clustering algorithms [26] are applied to find dense clusters of similar documents. From these clusters, the most salient terms are extracted through a class-based term frequency inverse document frequency measure (c-TFIDF), which can be interpreted as topic representations.

To track how topics evolve over time, BERTopic divides each cluster into an ordered set of time sequences and extracts the c-TFIDF representation for each time sequence in the cluster. This allows for an analysis of how topics evolve over time. Unlike LDA and DTM, BERTopic does not require a pre-determined fixed number of topics. Instead, it seeks to find the maximum number of topics and uses cosine similarity [27] to merge similar topics as needed.

In this study, the GBERT model [28] was employed to create document embeddings. Longer documents were split into segments with a maximum of 400 words to prevent loss of information due to truncation, as most BERT models have a maximum sequence length of 514 tokens. Using the HDBSCAN clustering algorithm, a grid search revealed that min_cluster_size = 66 and min_samples = 13 resulted in the lowest number of outliers and generated 65 topics. By considering the cosine similarity between topics as well as manual examination of these similarities, the final number of topics was reduced to 52 by merging manually selected topics. Finally, the topic word and phrase representations were extracted with [29].

### 2.4 Topic Model Evaluation

To evaluate the performance of the investigated topic models, three commonly used measures were employed. These include i) normalized pointwise mutual information (NPMI) [30], ii) $C_V$, and iii) topic diversity [31]. NPMI and $C_V$ are used to measure the co-occurrence of the most salient terms within a specified search window for each topic. This is commonly referred to as topic coherence [32]. No unified standard exists on what coherence values constitute a good quality topic model. Therefore, the respective scores are often used to just directly compare models and to select the model with the highest score.

Topic diversity ranks topics between 0 and 1 based on the amount of unique salient words. Higher scores indicate more meaningful and less redundant topics. These metrics are calculated using the OCTIS library with default settings [33]. In addition, a reference corpus is required to measure topic coherence. For this purpose, the preprocessed version of the climate change corpus is used as a reference for each model.

## 3 Results and Discussion

### 3.1 Evaluation Metrics

Table 1 compares the performance of the two models for three different evaluation metrics. The results show that DTM outperformed BERTopic in terms of $C_V$ and NPMI, while BERTopic scores higher in terms of topic diversity. However, it is difficult to determine which model performs better based solely on these metrics, as there is no clear consensus on the preference for "coherent" or "diverse" topics.

This ambiguity highlights an important issue that has recently arisen in the field of topic modeling, as pointed out in [34]. Evaluation methods were primarily developed to assist human evaluation, and should not be relied upon solely. Furthermore, coherence metrics were originally designed for statistical models, such as LDA and DTM, and have been shown to behave differently for embedding models such as BERTopic [35].

Several explanations can be provided for the significant difference in coherence scores between the two models. Firstly, coherence scores have been found to be impacted by document length, with shorter documents tending to decrease coherence [36]. BERTopic, which relies on BERT embeddings, is limited to a maximum sequence length of 512 tokens, while DTM can handle documents of any length.

Secondly, coherence is dependent on the reference corpus used, which is typically the pre-processed corpus the model was trained on [34]. Therefore, if different pre-processing steps have been used between models, their coherence will be calculated based on these different versions of the same corpus. As a result, the POS filtering applied to DTM in this study affects the final coherence scores by increasing the likelihood of co-occurrences within the specified window.

Finally, high coherence scores have been found to provide limited assistance in identifying high-quality topics, as more abstract or general topics frequently rank among the most coherent [37], [38]. The subsequent subsection of this paper shows that such topics are less common in BERTopic.

Table 1: Evaluation metric scores for DTM and BERTopic averaged over all time sequences. The top score for each metric is highlighted.

| Measure | DTM | BERTopic |
|---|---|---|
| $C_V$ | **0.531** | 0.444 |
| NPMI | **-0.012** | -0.096 |
| Diversity | 0.878 | **0.894** |

## 3.2 Human Evaluation

The previous subsection highlights the limitation of relying solely on automated evaluation metrics in evaluating the quality of a topic model. To determine the suitability of a model for answering research questions or providing a general understanding of a corpus, human evaluation is essential, despite its subjectivity and potential variability based on the specific use case [39].

A common method of human evaluation is to label each topic based on its most prominent words and representative documents. Table 2 provides an example topic with comparable context of salient words of both methods. DTM indicates that articles in this topic deal with housing related to climate protection and Austrian municipalities. A close reading of the most representative articles of the topic shows that these articles mainly discuss subsidies for housing renovations that reduce carbon emissions and climate regulations for new housing projects. Comparing the housing-related topic generated by BERTopic, the salient terms differ greatly. These already include heating solutions and make the topic more specific and interpretable. The manual analysis of associated articles further supports these observations.

One of the main findings from comparing the two methods is that DTM topics provide a broader overview, while topics from BERTopic are more specific. To be more precise, BERTopic identifies topics that focus on specific events, such as the introduction of a nationwide public transport ticket or discussion around a planned third runway for the Vienna airport. The difference in salient words from both approaches is not rooted in the lower number of topics from DTM (35), compared to BERTopic (52). Indeed, the DTM model with 47 results in very similar topics compared to the 35-topic run but with higher numbers of redundant or incomprehensible marginal topics. The underlying reason for the differences in salient words and topic representations result from the c-TFIDF extraction process of BERTopic. The extraction is done after topic clustering and evaluates the importance of words compared to the whole corpus according to their relevance to a topic. For DTM, in contrast, ranks terms according to how likely they are to occur in a topic. Therefore, topics in DTM tend to contain terms that are very common within their related documents, whereas BERTopic prioritizes words that are comparatively uncommon in unrelated documents.

Table 2: Example topic found in both models dealing with housing in the year 2019 and the ten most salient words translated to English.

| DTM |
|---|
| City, new, Project, Vienna, Municipality, Building, House, in Vienna, Climate Protection, big |
| **BERTopic** |
| Building, House, Heat Pump, Oil Heating, Real Estate, Housing, District Heating, Passive House, Schleicher, Property Developer |

Moreover, the difference of generic and specific topic representations of each model is observed regarding topic evolution. In DTM the changes from one time sequence to the next are more moderate and variations in the ranking of salient words happen slowly. New salient

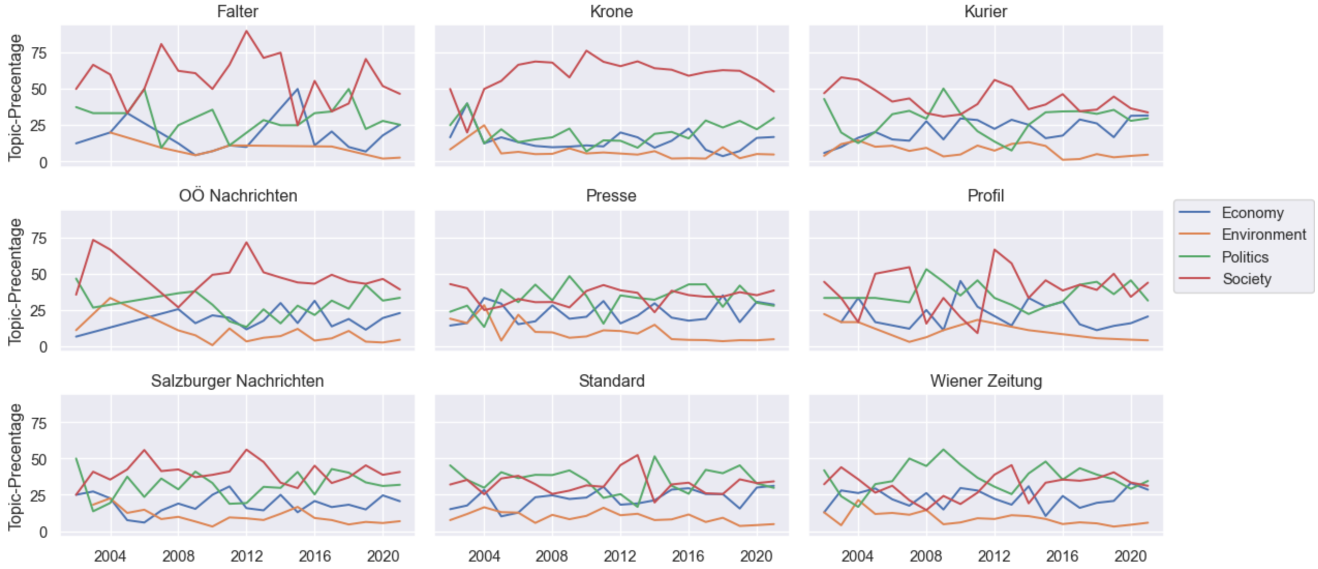


Figure 1: Evolution of each DTM category present in each newspaper

words are mostly introduced when they correspond to heavily covered events or to events that remain relevant over a long period of time, such as the Green Party becoming part of the government or the Green Deal proposed by the EU.

BERTopic evaluation showed more pronounced changes in topic representation between each time sequence. Here, events influence salient words of a topic stronger when they are only significantly mentioned within a single time sequence. Notable examples are the *Ibiza affair*, which led to the dissolution of the previous government and the appearance of the movie *An Inconvenient Truth* by Al Gore.

Another important result is obtained from examining a general overview of the discourse around climate protection in Austria by manually grouping all topics into one of four fundamental categories: **Economy**, **Environment**, **Politics**, and **Society** (see Figure 1 and Figure 2). Comparing category evolution of both models shows similar patterns over time. The two newspapers with a pronounced partisan orientation, *die Krone* (right wing) and *der Falter* (left wing), strongly focus on the category **Society** and feature many interviews and social commentary. The low number of articles published in *der Falter*, however, leads to high variability between the years. The other seven newspapers, particularly *der Standard* and *die Presse*, are more equally distributed between the categories **Society** and **Politics**. Additionally, the category **Environment** plays only a minor role in the climate protection coverage of all newspapers. Environmental aspects are rarely the sole or major issue articles focus on. Instead, they are mainly discussed as positive or negative consequences that arise from actions taken by the government, companies or private persons.

The most striking difference between DTM and BERTopic in terms of the share of each category in overall reporting is the higher percentage of articles assigned to the category **Economy** in the DTM results. The particular reason for this lies in a crucial difference in topic formation between both algorithms: In DTM, one of the largest topics is labeled **Economic Policies**, however, applying BERTopic did not result in a comparable topic. Instead, the articles from this DTM topic are contained in a number of smaller topics, that are split up between the categories **Politics** and **Economy**. Therefore, the overall representation of the category **Economy** is much lower in BERTopic than DTM results.

## 4 Summary and Outlook

In this paper we compared the performance of DTM and BERTopic for automated content analysis. With this approach, we contribute to the field of automated text analysis and topic modelling. We expand upon the ongoing discussion about the ambiguity of automatic evaluation metrics. Our results indicate that rather than using these metrics as the sole indicator of a model’ s performance, human evaluation must be included in a complimentary fashion. As our findings demonstrate, BERTopic generates topics that are more specific when

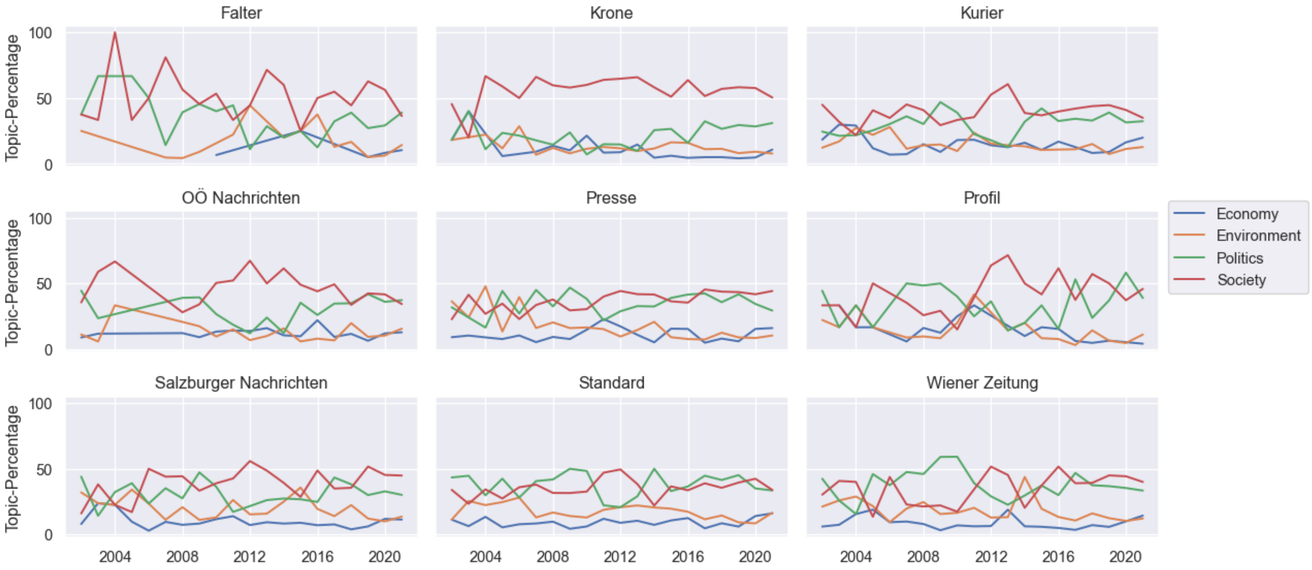


Figure 2: Evolution of each BERTopic category present in each newspaper.

compared to DTM but ranks lower in two widely used coherence metrics. This highlights the need for new automatic scoring metrics that can account for the differences between statistical and embedding-based models, as well as the continued importance of human evaluation.

Despite the aforementioned deliberations concerning evaluation metrics, they are often employed in research when comparing BERTopic to well established topic modeling techniques, mainly LDA. In a large number of cases BERTopic outperforms LDA in both coherence and diversity [15], [36], [40] – [44], but there is a notable number of studies, where the difference is either negligible or inconclusive as it is in our study [45] – [50]. Human evaluation, however, tends to favor BERTopic regardless of evaluation metric scores.

Besides the general limitations of topic modeling [37], the following limitations apply to our study. Only print articles are considered in our investigation. However, since 2021 online media has surpassed print media as the main source of news in Austria [51]. Most newspapers include additional articles on their respective online portals, which are not included in the presented work. If certain topics are covered more prominently in online articles, they will not be properly represented in our models. Future research could aim to investigate the differences between print and online media. Another important factor that affects BERTopic is the used embedding model. Since the model was not fine-tuned, some words that are highly specific to climate protection are included in the model vocabulary and treated as out-of-vocabulary words. As a result, the semantic clustering might not recognize the similarity between some articles and falsely assign them to different clusters.

Manual investigation regarding the respective corpus on climate mitigation articles, shows several critical differences between DTM and BERTopic results. Since BERTopic considers contextual information, articles can be clustered into topics with very specific salient words that represent the topics. As a result, the evolution of salient words of a topic contains information about events that are limited to a single time sequence. Therefore, BERTopic facilitates secondary manual analysis of researches and domain experts by great detail in temporally resolved quantitative analysis. Nonetheless, DTM also provides a valuable tool for time-dependent topic analysis. We find that topic generation of BERTopic and DTM does not produce comparable topics on every subject. Instead, topics generated by one of the algorithms may not show up as a separate topic in another because their respective content has been integrated and partitioned into other topics. This bears the risk that important content is overlooked when only applying one of the two algorithms, as we highlighted in the case of economic policies. We conclude that the dual and comparative deployment of both methods, accompanied by a human evaluation, leads to the best results in respect to content analysis on climate mitigation coverage.

## Acknowledgements

We thank the reviewers for their valuable comments.

## References

[1] S. Ungar, „The Rise and (Relative) Decline of Global Warming As a Social Problem", *Sociol. Q.*, Bd. 33, Nr. 4, S. 483–501, 1992, doi: 10.1111/j.1533-8525.1992.tb00139.x.

[2] Z. W. Kundzewicz, P. Matczak, I. M. Otto, und P. E. Otto, „From "atmosfear" to climate action", *Environ. Sci. Policy*, Bd. 105, Nr. December 2019, S. 75–83, 2020, doi: 10.1016/j.envsci.2019.12.012.

[3] M. McCombs, „The Agenda-Setting Role of the Mass Media in the Shaping of Public Opinion", 2011, [Online]. Verfügbar unter: https://www.researchgate.net/publication/237394610_The_Agenda-Setting_Role_of_the_Mass_Media_in_the_Shaping_of_Public_Opinion.

[4] R. Willis, „How Members of Parliament understand and respond to climate changett", *Sociol. Rev.*, Bd. 66, Nr. 3, S. 475–491, 2018, doi: 10.1177/0038026117731658.

[5] M. Stede und R. Patz, „The Climate Change Debate and Natural Language Processing", in *Proceedings ofthe 1st Workshop on NLP for Positive Impact*, 2021, S. 8–18, doi: 10.18653/v1/2021.nlp4posimpact-1.2.

[6] A. B. Dieng, F. J. R. Ruiz, und D. M. Blei, „The dynamic embedded topic model", *arXiv*, S. 1–17, 2019.

[7] M. S. Schäfer und V. Hase, „Computational methods for the analysis of climate change communication: Towards an integrative and reflexive approach", *Wiley Interdiscip. Rev. Clim. Chang.*, Nr. April 2022, S. 1–10, 2022, doi: 10.1002/wcc.806.

[8] D. M. Blei, A. Y. Ng, und M. I. Jordan, „Latent Dirichlet allocation", *J. Mach. Learn. Res.*, Bd. 3, Nr. 4–5, S. 993–1022, 2003.

[9] R. Grundmann, „Using large text news archives for the analysis of climate change discourse: some methodological observations", *J. Risk Res.*, Bd. 0, Nr. 0, S. 1–13, 2021, doi: 10.1080/13669877.2021.1894471.

[10] M. S. Schäfer und I. Schlichting, „Media representations of climate change: A meta-analysis of the research field", *Environ. Commun.*, Bd. 8, Nr. 2, S. 142–160, 2014, doi: 10.1080/17524032.2014.914050.

[11] J. Bohr, „"Reporting on climate change: A computational analysis of U.S. newspapers and sources of bias, 1997–2017"", *Glob. Environ. Chang.*, Bd. 61, Nr. February, S. 102038, 2020, doi: 10.1016/j.gloenvcha.2020.102038.

[12] V. Hase, D. Mahl, M. S. Schäfer, und T. R. Keller, „Climate change in news media across the globe: An automated analysis of issue attention and themes in climate change coverage in 10 countries (2006–2018)", *Glob. Environ. Chang.*, Bd. 70, 2021, doi: 10.1016/j.gloenvcha.2021.102353.

[13] T. R. Keller, V. Hase, J. Thaker, D. Mahl, und M. S. Schäfer, „News Media Coverage of Climate Change in India 1997–2016: Using Automated Content Analysis to Assess Themes and Topics", *Environ. Commun.*, Bd. 14, Nr. 2, S. 219–235, 2020, doi: 10.1080/17524032.2019.1643383.

[14] D. M. Blei und J. D. Lafferty, „Dynamic topic models", *ACM Int. Conf. Proceeding Ser.*, Bd. 148, S. 113–120, 2006, doi: 10.1145/1143844.1143859.

[15] M. Grootendorst, „BERTopic: Neural topic modeling with a class-based TF-IDF procedure", 2022, [Online]. Verfügbar unter: http://arxiv.org/abs/2203.05794.

[16] C. Ha, V. D. Tran, L. Ngo Van, und K. Than, „Eliminating overfitting of probabilistic topic models on short and noisy text: The role of dropout", *Int. J. Approx. Reason.*, Bd. 112, S.

85 – 104, 2019, doi: 10.1016/j.ijar.2019.05.010.

[17] C. Wartena, „A probabilistic morphology model for German lemmatization ", in *Proceedings of the 15th Conference on Natural Language Processing, KONVENS 2019*, 2019, Nr. Konvens, S. 40 – 49.

[18] I. Vayansky und S. A. P. Kumar, „A review of topic modeling methods ", *Inf. Syst.*, Bd. 94, S. 101582, 2020, doi: 10.1016/j.is.2020.101582.

[19] J. Marjanen, E. Zosa, S. Hengchen, L. Pivovarova, und M. Tolonen, „Topic modelling discourse dynamics in historical newspapers ", in *CEUR Workshop Proceedings*, 2020, Bd. 2865, S. 63 – 77.

[20] M. Röder, A. Both, und A. Hinneburg, „Exploring the space of topic coherence measures ", in *WSDM 2015 - Proceedings of the 8th ACM International Conference on Web Search and Data Mining*, 2015, S. 399 – 408, doi: 10.1145/2684822.2685324.

[21] S. Mosallaie, M. Rad, A. Schiffauerova, und A. Ebadi, „Discovering the evolution of artificial intelligence in cancer research using dynamic topic modeling ", *COLLNET J. Sci. Inf. Manag.*, Bd. 15, Nr. 2, S. 225 – 240, 2021, doi: 10.1080/09737766.2021.1958659.

[22] H. Yin, S. Yang, und J. Li, „Detecting Topic and Sentiment Dynamics Due to COVID-19 Pandemic Using Social Media ", *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, Bd. 12447 LNAI, Nr. June, S. 610 – 623, 2020, doi: 10.1007/978-3-030-65390-3_46.

[23] R. Řehůřek und P. Sojka, „Software framework for topic modelling with large corpora ", in *Proceedings of LREC 2010 workshop New Challenges for NLP Frameworks*, 2010, S. 46 – 50, doi: 10.13140/2.1.2393.1847.

[24] J. Devlin, M. W. Chang, K. Lee, und K. Toutanova, „BERT: Pre-training of deep bidirectional transformers for language understanding ", *NAACL HLT 2019 - 2019 Conf. North Am. Chapter Assoc. Comput. Linguist. Hum. Lang. Technol. - Proc. Conf.*, Bd. 1, Nr. Mlm, S. 4171 – 4186, 2019.

[25] L. McInnes, J. Healy, und J. Melville, „UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction ", *J. Open Source Softw.*, Bd. 3, Nr. 29, 2018, [Online]. Verfügbar unter: http://arxiv.org/abs/1802.03426.

[26] L. McInnes und J. Healy, „Accelerated Hierarchical Density Based Clustering ", in *IEEE International Conference on Data Mining Workshops, ICDMW*, 2017, S. 33 – 42, doi: 10.1109/ICDMW.2017.12.

[27] A. Singhal, „Modern Information Retrieval: A Brief Overview ", *Bull. Tech. Comm. Data Eng.*, Bd. 24, Nr. 4, S. 35 – 43, 2001.

[28] B. Chan, S. Schweter, und T. Möller, „German' s Next Language Model ", in *Proceedings of the 28th International Conference on Computational Linguistics*, 2021, S. 6788 – 6796, doi: 10.18653/v1/2020.coling-main.598.

[29] T. Schopf, S. Klimek, und F. Matthes, „PatternRank: Leveraging Pretrained Language Models and Part of Speech for Unsupervised Keyphrase Extraction ", in *Proceedings of the 14th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management - KDIR*, 2022, S. 243 – 248, doi: 10.5220/0011546600003335.

[30] G. Bouma, „Normalized ( Pointwise ) Mutual Information in Collocation Extraction ", in *Proceedings of German Society for Computational Linguistics (GSCL 2009)*, 2009, S. 31 – 40.

[31] A. B. Dieng, F. J. R. Ruiz, und D. M. Blei, „Topic modeling in embedding spaces ", *arXiv*, Bd. 8, S. 439 – 453, 2019, doi: 10.1162/tacl_a_00325.

[32] S. Syed und M. Spruit, „Full-Text or abstract? Examining topic coherence scores using latent dirichlet allocation ", in *Proceedings - 2017*

*International Conference on Data Science and Advanced Analytics, DSAA 2017*, 2017, S. 165–174, doi: 10.1109/DSAA.2017.61.

[33] S. Terragni, E. Fersini, B. Galuzzi, P. Tropeano, und A. Candelieri, „OCTIS: Comparing and optimizing topic models is simple! “, in *EACL 2021 - 16th Conference of the European Chapter of the Association for Computational Linguistics, Proceedings of the System Demonstrations*, 2021, S. 263–270, doi: 10.18653/v1/2021.eacl-demos.31.

[34] A. Hoyle, P. Goel, D. Peskov, A. Hian-Cheong, J. Boyd-Graber, und P. Resnik, „Is Automated Topic Model Evaluation Broken?: The Incoherence of Coherence “, *Adv. Neural Inf. Process. Syst.*, Bd. 3, Nr. NeurIPS, S. 2018–2033, 2021.

[35] C. Doogan und W. Buntine, „Topic Model or Topic Twaddle? Re-evaluating Semantic Interpretability Measures “, in *NAACL-HLT 2021 - 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference*, 2021, S. 3824–3848, doi: 10.18653/v1/2021.naacl-main.300.

[36] M. de Groot, M. Aliannejadi, und M. R. Haas, „Experiments on Generalizability of BERTopic on Multi-Domain Short Text “, S. 1–3, 2022, [Online]. Verfügbar unter: http://arxiv.org/abs/2212.08459.

[37] D. Korenčić, S. Ristov, und J. Šnajder, „Document-based topic coherence measures for news media text “, *Expert Syst. Appl.*, Bd. 114, S. 357–373, 2018, doi: 10.1016/j.eswa.2018.07.063.

[38] S. I. Nikolenko, S. Koltcov, und O. Koltsova, „Topic modelling for qualitative studies “, *J. Inf. Sci.*, Bd. 43, Nr. 1, S. 88–102, 2017, doi: 10.1177/0165551515617393.

[39] I. Harrando, P. Lisena, und R. Troncy, „Apples to Apples: A Systematic Evaluation of Topic Models “, in *International Conference Recent Advances in Natural Language Processing, RANLP*, 2021, Nr. January, S. 483–493, doi: 10.26615/978-954-452-072-4_055.

[40] R. Egger und J. Yu, „A Topic Modeling Comparison Between LDA, NMF, Top2Vec, and BERTopic to Demystify Twitter Posts “, *Front. Sociol.*, Bd. 7, Nr. May, S. 1–16, 2022, doi: 10.3389/fsoc.2022.886498.

[41] T. Schmiedel, O. Müller, und J. vom Brocke, „Topic Modeling as a Strategy of Inquiry in Organizational Research: A Tutorial With an Application Example on Organizational Culture “, *Organ. Res. Methods*, Bd. 22, Nr. 4, S. 941–968, 2019, doi: 10.1177/1094428118773858.

[42] Z. Zhang, M. Fang, L. Chen, und M. R. Namazi-Rad, „Is Neural Topic Modelling Better than Clustering? An Empirical Study on Clustering with Contextual Embeddings for Topics “, in *NAACL 2022 - 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference*, 2022, S. 3886–3893, doi: 10.18653/v1/2022.naacl-main.285.

[43] B. Ogunleye, T. Maswera, L. Hirsch, J. Gaudoin, und T. Brunsdon, „Comparison of Topic Modelling Approaches in the Banking Context “, *Appl. Sci.*, Bd. 13, Nr. 2, 2023, doi: 10.3390/app13020797.

[44] A. Abuzayed und H. Al-Khalifa, „BERT for Arabic Topic Modeling: An Experimental Study on BERTopic Technique “, *Procedia CIRP*, Bd. 189, S. 191–194, 2021, doi: 10.1016/j.procs.2021.05.096.

[45] X. Xu, G. Stulp, und A. Gauthier, „Understanding Narratives from Demographic Survey Data: a Comparative Study with Multiple Neural Topic Models “, in *Proceedings ofthe Fifth Workshop on Natural Language Processing and Computational Social Science (NLP+CSS)*, 2022, Nr. 2018, S. 33–38.

[46] H. Zankadi, A. Idrissi, N. Daoudi, und I. Hilal, „Identifying learners' topical interests from social media content to enrich their course preferences in MOOCs using topic modeling and NLP techniques", *Educ. Inf. Technol.*, 2022, doi: 10.1007/s10639-022-11373-1.

[47] K. N. Keya, Y. P. Healx, und J. R. Foulds, „Neural Embedding Allocation: Distributed Representations of Topic Models", *Comput. Linguist.*, Bd. 48, Nr. 4, S. 1021–1052, 2022, doi: 10.1162/coli_a_00457.

[48] E. Zeng, M. Wei, T. Gregersen, T. Kohno, und F. Roesner, *Polls, clickbait, and commemorative $2 bills*, Bd. 1, Nr. 1. Association for Computing Machinery, 2021.

[49] Y. Meng, Y. Zhang, J. Huang, Y. Zhang, und J. Han, „Topic Discovery via Latent Space Clustering of Pretrained Language Model Representations", in *WWW 2022 - Proceedings of the ACM Web Conference 2022*, 2022, S. 3143–3152, doi: 10.1145/3485447.3512034.

[50] S. Vasudeva Raju, B. Kumar Bolla, D. K. Nayak, und K. H. Jyothsna, „Topic Modelling on Consumer Financial Protection Bureau Data: An Approach Using BERT Based Embeddings", 2022, doi: 10.1109/I2CT54291.2022.9824873.

[51] S. Gadringer, R. Holzinger, S. Sparviero, J. Trappel, und C. Schwarz, „Digital News Report Austria 2021. Detailergebnisse für Österreich", 2021. doi: 10.5281/ZENODO.4775408.

# Scientific Computing 2023

## Poster

*Petra Fenzel:* **A Pony Farm's Combinatorial Use Case: Finding a Suitable Assignment by Using a SAT Solver** ........ 222

*David Koch:* **Energiespeicher im privaten Sektor – Eine nachhaltige Energielösung für Österreich? ........** 223

*Manuel Karner:* **Extended Virtual Acrophobia Treatment** ........ 224

*Florian Gürtler, Fabian Anderle, Martin Schweighofer und Marie-Therese Uhl:* **Generating Satellite Deepfakes with AI: Using Handmade Sandscapes as Input and Projection Mapping for Output** ........ 225

*Luis Trojer, Raphael Vierhauser und Gerhard Pirker:* **Hydrofoil Boat Design** ... 226

*Laura Gregorc:* **In-Cabin Gaze Detection using Neural Networks on Mobile Devices** ........ 227

Sandra Schadenbauer, Alexander Nischelwitzer: **Interactive Sandscape – explore the interactivity of height modeling by shaping your own landscape with kinetic sand** ........ 229

Daniel Kavic, Michael Melcher, Christian Bernhard und Roman Rössler: **Investigation of Process Influences on the Steel Composition and Inclusion Modification in Secondary Metallurgy using Diverse Statistical Learning and Data Mining Methods**........ 230

Florian Auer-Welsbach, Andreas Windisch, Giacomo Nodjoumi: **Landform Detection on Mars Using a U-Net Architecture** ........ 231

Lisa Göberndorfer: **Modeling Rebound Effects from a Shift to Electric Vehicles in Private Mobility in Styria** ........ 232

Florian Leber: **Vorhersage von Radsatz-Reprofilierungen durch künstliche Intelligenz**........ 233

FH CAMPUS WIEN
UNIVERSITY OF APPLIED SCIENCES

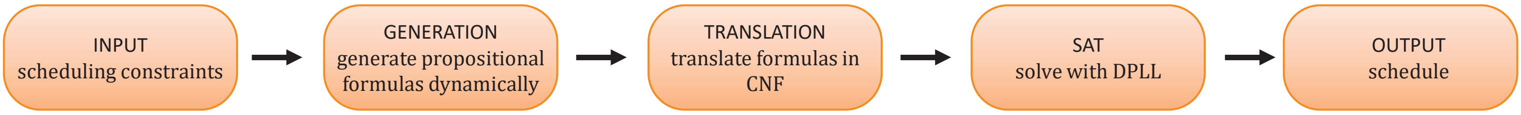
Computer Science and Digital Communications

Author: Petra Fenzel

$(x_{p,i,1,1,1,1} \wedge x_{p,i,1,2,1,1} \wedge x_{p,i,1,3,1,1} \wedge x_{p,i,1,4,1,1}) \Rightarrow$
$\neg (x_{p,i,1,5,1,1} \vee x_{p,i,1,6,1,1} \vee x_{p,i,1,7,1,1})$

# A Pony Farm's Combinatorial Use Case: Finding a Suitable Assignment by Using a SAT Solver [1]

## Concept



INPUT
scheduling constraints

GENERATION
generate propositional formulas dynamically

TRANSLATION
translate formulas in CNF

SAT
solve with DPLL

OUTPUT
schedule

Why: current digital solutions do not consider/assign all resources and their dependencies

6 resources are defined and form the indexes of the variable which is:

$$x_{p,i,a,r,l,t}$$

in which "p" stands for pony, "i" for (riding) instructor, "a" for (riding) arena,"r" for rider, "l" for lesson and "t" for timeslot.

The resources have different constraints and dependencies to each other. Some examples are mentioned below:

**Riding Arenas**

| size | suitable for | max. simultaneous occupancy |
|---|---|---|
| small | beginner, intermediate lessons | 2 x beginner lessons or<br>1 x beginner and 1 x intermediate lessons |
| big | beginner, intermediate, well-advanced lessons | 2 x beginner lessons or<br>1 x beginner and 1 x intermediate lessons or<br>1 x well-advanced |

**Riders**

| skill-level | suitable for | number of morning lessons | number of afternoon lessons |
|---|---|---|---|
| beginner | all ponies | 1 | 1 |
| intermediate | all ponies | 1 | 1 |
| intermediate | big ponies up | 1 | 1 |
| well-advanced | big ponies up | 1 | 1 |

The dynamically entered resources and their constraints are then brought into propositional formulas and are converted to CNFs.

Consider an input of:

- 3 beginners & 8 intermediates
- constraints: max. 2 beginners & max. 6 intermediates per lesson
- for timeslot no. 1 and a small riding arena

Propositional formula:

$(x_{p,i,1,1,1,1} \wedge x_{p,i,1,2,1,1} \wedge x_{p,i,1,4,2,1} \wedge x_{p,i,1,5,2,1} \wedge x_{p,i,1, 6,2,1} \wedge x_{p,i,1,7,2,1} \wedge x_{p,i,1,8,2,1} \wedge x_{p,i,1,8,2,1}) \Rightarrow \neg (x_{p,i,1,3,1,1} \vee x_{p,i,1,9,2,1} \vee x_{p,i,1,10,2,1})$

Each possible combination of the variable x has to be saved in a unique integer value dependent on the user input to save the clauses then in a DIMACS file. This is done in a Pandas DataFrame of which the unique index is taken. Example for 25 ponies, 5 riders, 4 timeslots, 3 instructors, 3 kind of lessons, 1 riding arena:

| | |
|---|---|
| 0 | 0,0,0,0,0,0 |
| 1 | 0,0,0,0,0,1 |
| 2 | 0,0,0,0,1,0 |
| 3 | 0,0,0,0,1,1 |
| 4 | 0,0,0,0,2,0 |
| ... | ... |
| 24955 | 25,5,4,3,1,1 |
| 24956 | 25,5,4,3,2,0 |
| 24957 | 25,5,4,3,2,1 |
| 24958 | 25,5,4,3,3,0 |
| 24959 | 25,5,4,3,3,1 |

The example input of 3 beginners, 8 intermediates and each 2 x 30 minutes morning/afternoon timeslots results already in 252 clauses of which each one is saved in one line in the DIMACS file, e.g.:

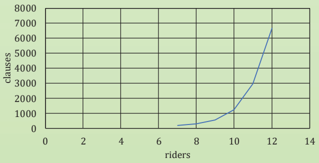
```
p cnf 44 252
-195 – 355 -515 -677 -837 -997 -1157 -1317 -1477 0
-195 – 355 -515 -677 -837 -997 -1157 -1317 -1637 0
-195 – 355 -515 -677 -837 -997 -1157 -1317 -1797 0
-195 – 355 -515 -677 -837 -997 -1157 -1477 -1797 0
………
195 227 0
-195 -227 0
```

Chosen algorithm: „pure“ DPLL: Davis, Putnam, Logemann, Loveland with

- unit clause propagation
- pure literal elimination
- chronological backtracking

Impact of constraints on number of clauses and run time:

7 - 12 intermediates / 8 timeslots / 3 constraints



Although PoC with small number of resources/constraints:

- number of clauses grows quickly
- erratic run time as backtracking is purely chronological

Samples of 11 intermediates, 8 timeslots, → 2948 clauses

positive/negative polarity: how often a variable was chosen during the back-tracking process

| sample no. | run time in sec. | positive polarity | negative polarity |
|---|---|---|---|
| 1 | 3480.275 828 | 5770 | 5744 |
| 2 | 20.557 193 | 187 | 161 |
| 3 | 894.990 265 | 6531 | 6501 |
| 4 | 11.198 000 | 111 | 81 |
| 5 | 3.701 843 | 28 | 2 |
| 6 | 41.438 544 | 258 | 230 |
| 7 | 156.692 385 | 1222 | 1190 |
| 8 | 15.814 933 | 142 | 109 |
| 9 | 8055.212 759 | 37 589 | 37 559 |
| 10 | 4.892 999 | 40 | 11 |

Conclusion:

- dynamic generation of clauses is complex but feasible
- definition of constraints: soft/hard: where does the flexibility end?
- more elaborated algorithm is necessary in regards of run time e.g.:
  - branching heuristic
  - conflict-driven clause learning
- Is it all about SAT solvers? What about e.g. genetic algorithms?

References:
[1] Petra Fenzel, Bachelor Thesis „A Pony Farm's Combinatorial Use Case: Finding a Suitable Assignment by Using a SAT Solver“, 13.08.2022

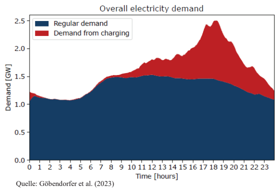
Many thanks to my supervisor Dr. Christian Steineder for his dedicated support, feedback and ongoing encouragement throughout this bachelor project as well as to my husband Jürgen and my children Max and Lea for their patience and tolerance.

# Energiespeicher im privaten Sektor – Eine nachhaltige Energielösung für Österreich?

Autor: David Koch | Betreuung: Georg Jäger

## Einführung | Hintergrund

- ➢ Energiewende in Österreich
- ➢ Zunehmender Strombedarf durch E-Mobilität als Herausforderung
- ➢ Missmatch: Solarenergieproduktion tagsüber – Strombedarf abends
- ➢ Gefahr von Lock-in Effekt in eine Technologie
- ➢ Innovation öffnet Diskrepanz zwischen Haushalten und Staat



Quelle: Göbendorfer et al. (2023)

**Pfadabhängigkeit**

Die Wahl eines bestimmten Weges oder Pfads kann die Optionen und Möglichkeiten in der Zukunft einschränken oder erweitern. Ein Pfad, der einmal eingeschlagen wurde, kann schwer zu ändern sein und den weiteren Verlauf des Systems oder Prozesses beeinflussen.

**Lock-in-Effekt**

Der Lock-in-Effekt ist ein spezielles Beispiel für Pfadabhängigkeit, bei dem eine Entscheidung oder ein Ereignis das System oder den Prozess auf einen bestimmten Pfad bringt, der schwer zu verlassen ist. In diesem Fall kann eine anfängliche Investition oder Entscheidung dazu führen, dass die zukünftigen Optionen eingeschränkt werden oder sogar ausgeschlossen werden. Der Lock-in-Effekt kann zu einer Art von "Institutionalisierung" führen, bei der es schwierig oder unmöglich wird, zu einer früheren oder alternativen Entscheidung zurückzukehren.

Forschungsaspekte

Sind Batterriespeicher ein möglicher Lösungsansatz im privaten Sektor zur Entlastung des Stromnetzes?

Was sind nötige Anrzeie für den privaten Sektor?

Wie gut ist es, dieses Ziel zu erreichen?

## Methodik – Agenten-basiertes Modell

ABM-Tool: NetLogo: Multiagentenprogrammiersprache mit integrierter Modellierungsumgebung

Agenten: relevante Haushalte in Österreich

Modellierungszeitraum: 15 Jahre

Investitionsentscheidung zwischen PV und PV& Batterie abhängig von

- ➢ Anschaffungskosten
- ➢ Strompreis
- ➢ Einspeisevergütung
- ➢ Subventionierung

Investitionsentscheidungsformel eines Haushalts

$$\frac{Anschaffungskosten - Subvention}{Eigennutzen * Strompreis * p + e * (1 - Eigennutzen) * p} \leq Amortisationszeit$$

p – jährliche Stromproduktion (kWh)
e – Einspeisevergütung (€/kWh)

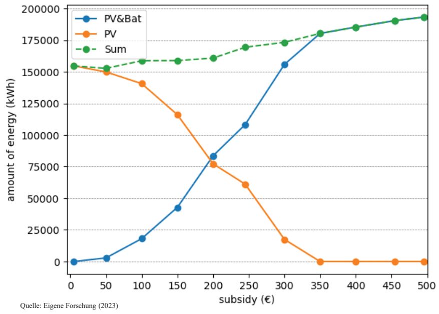
Quelle: Eigene Forschung (2023)

Der Modellkern besteht darin, dass Agenten (Haushalte) eine Investitionsentscheidung treffen, die für sie in Anbetracht ihrer Situation profitabel ist.

Durch eine Szenarienanalyse im Rahmen einer One-factor-at-a-time-Analyse können Auswirkungen von Änderungen der Variablen auf Kaufentscheidung einzelner Agenten ausgewertet warden. Das in NetLogo integrierte Tool BehaviorSpace ermöglicht es diese Analysen durchzuführen.

## Ergebnisse

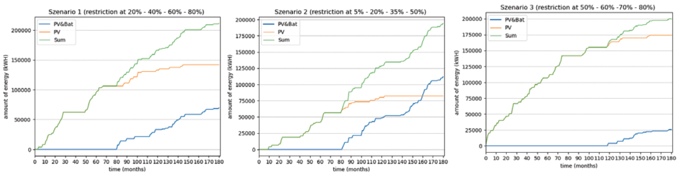
### Untersuchung von Effekten staatlicher Förderung



Quelle: Eigene Forschung (2023)

Die Grafik zeigt die im Modell erzeugte Energiemengen (kWh) der Haushalte durch Solaranlagen und Solaranlagen mit Batterie bei unterschiedlichen Förderungszuschüssen (subsidy (€)).

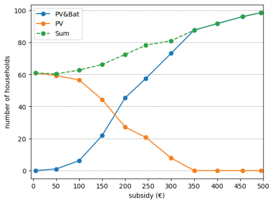
### Untersuchung von Effekten verschiedener Einspeisevergütungsstrategien



Quelle: Eigene Forschung (2023)

Die Untersuchung von verschiedenen Einspeisevergütungsszenarien zeigt die Effekte von Senkung der Einspeisevergütung in 25% Schritten ab einem gewissen Prozentsatz an Solaranlagen im System. Die kumulierten Summen über die Zeit aufgetragen machen die Strategien auf im System erzeugte Gesamtenergiemenge vergleichbar.

### Effekte von Batterien im Stromsystem



Quelle: Eigene Forschung (2023)

Eine höhere Subventionierung generiert eine höhere Investitionsbereitschaft für PV-Anlagen mit Batterie. Um Rückschlüsse von der Makro- auf die Mikroebene zu ziehen, wird über den Prozentsatz an Batterien im Modell Auswirkung auf ganz Österreich berechnet. Hierbei wird die durch Batterien im Haushaltssektor erzeugte Energiemenge mit dem E-Mobilität-Energiebedarf auf nationaler Ebene gegenübergestellt.

| Anteil PV&Bat | 20% | 40% | 60% | 80% | 100% |
|---|---|---|---|---|---|
| Deckungsgrad Östereich | 3,49% | 6,97% | 10,46% | 13,94% | 17,42% |

Quelle: Eigene Forschung (2023)

Die Berechnung zeigt, dass wenn 100% aller relevanten Haushalte in Österreich einen Batteriespeicher besitzen diese Kapazität ausreicht, um den täglichen E-Mobilitätsenergiebedarf aller in Österreich lebenden Menschen zu 17,42% zu decken.

## Conclusio

Agentenbasierte Modelle ermöglichen die Darstellung eines Systems, welches von vielen Parameter beeinflusst wird. Durch eine Analyse von kleinen Änderungen einzelner Einflussgrößen wird es hier möglich, Effekte von finanziellen Anreizen zu erörtern. Hierbei wird deutlich, dass bereits eine geringe Subventionierung ausreicht um die Kaufanreize für Solaranlagen mit Batteriespeicher zu erhöhen und einen erheblichen Beitrag dazu leisten, den zukünftig erhöhten Strombedarf zu decken.

## Acknowledgement

Das Projekt wird im Rahmen einer Masterarbeit im Studium Innovations- und Nachhaltigkeitsmanagement am Fachbereich für Umweltsystemwissenschaften an der Karl-Franzens-Univeristät Graz in Betreueung von Ass.-Prof. Dr.rer.nat. BSc MSc Georg Jäger durchgeführt.

## Quellen

Abar, S., Theodoropoulos, G. K., Lemarinier, P., & O'Hare, G. M. (2017). Agent Based Modelling and Simulation tools: A review of the state-of-art software. Computer Science Review, 24, 13–33. https://doi.org/10.1016/j.cosrev.2017.03.001

AnyLogic. (2023, February 27). https://www.anylogic.de/

Chen, S.-H. (2017). Agent-Based Computational Economics: How the idea originated and where it is going (First edition). Routledge Advances in Experimental and Computable Economics. Taylor and Francis. https://www.taylorfrancis.com/books/mono/10.4324/9781315734422/agent-based-computational-economics-shu-heng-chen https://doi.org/10.4324/9781315734422

Energieinstitut: SUSI - die Strom-Unabhängigkeits-Simulation. (2023, March 1). https://www.energieinstitut.at/tools/susi/

Filatova, T., Verburg, P. H., Parker, D. C., & Stannard, C. A. (2013). Spatial agent-based models for socio-ecological systems: Challenges and prospects. Environmental Modelling & Software, 45, 1–7. https://doi.org/10.1016/j.envsoft.2013.03.017

GAMA. (2023, February 27). https://gama-platform.org/

Gilbert, N. (2020). Agent Based Models. SAGE Publications Ltd. https://doi.org/10.4135/9781526421036836969

Göberndorfer et al. (2022). Charging Rush Hour.

Heraclitus. (n.d.). Change is the only constant in life.

Irshaid, J. (2022). Herausforderungen für die lokale Energiewende in Österreich: Innovation und Umsetzung in ländlichen und städtischen Gemeinden. https://pure.iiasa.ac.at/id/eprint/17983/1/irshaid%20et%20al_%20session%20i_herausforderungen%20f%c3%bcr%20die%20lokale%20energiewende%20in%20%c3%96sterreich.pdf

Marchi, S. de, & Page, S. E. (2014). Agent-Based Models. Annual Review of Political Science, 17(1), 1–20. https://doi.org/10.1146/annurev-polisci-080812-191558

MASON: Multiagent Simulation Toolkit. (2023, January 25). https://cs.gmu.edu/~eclab/projects/mason/

NetLogo. (2023, February 27). https://ccl.northwestern.edu/netlogo/

Repast Suite Documentation. (2023, February 15). https://repast.github.io/

Schmidt, O., Melchior, S., Hawkes, A., & Staffell, I. (2019). Projecting the Future Levelized Cost of Electricity Storage Technologies. Joule, 3(1), 81–100. https://doi.org/10.1016/j.joule.2018.12.008

Schöniger. (2020). Erneuerbare Energiewende in Österreich–große Ziele, weiter Weg. https://energiestiftung.ch/files/energiestiftung/fliesstextbilder/veranstaltungen/2020%20fachtagung/04_praesentation_franziskascho%cc%88niger_fachtagung2020.pdf

SEO Analyse. (2022, April 21). Logo: Programmiersprache Begriffserklärung & Definition. https://www.seo-analyse.com/seo-lexikon/l/logo-programmiersprache/

STATISTIK AUSTRIA. (2023a). Bevölkerung zu Jahres-/Quartalsanfang. https://www.statistik.at/statistiken/bevoelkerung-und-soziales/bevoelkerung/bevoelkerungsstand/bevoelkerung-zu-jahres-/-quartalsanfang

STATISTIK AUSTRIA. (2023b). Privathaushalte. https://www.statistik.at/statistiken/bevoelkerung-und-soziales/bevoelkerung/familien-haushalte-lebensformen/privathaushalte

Tomschy, R., Herry, M., Sammer, G., Klementschitz, R., Riegler, S., Follmer, R., Gruschwitz, D., Josef, F., Gensasz, S., Kirnbauer, R., & Spiegel, T. (2016). Oesterreich unterwegs 2013/2014 : Ergebnisbericht zur oesterreichweiten Mobilitaetserhebung „Oesterreich unterwegs 2013/2014". https://trid.trb.org/view/1471489

Wagner, B., Hauer, C., Schoder, A., & Habersack, H. (2015). A review of hydropower in Austria: Past, present and future development. Renewable and Sustainable Energy Reviews, 50, 304–314. https://doi.org/10.1016/j.rser.2015.04.169

Wilenksy. (2004). Netlogo: A simple environment for modeling complexity. https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=a65e2af0f4a1b03db4b05357c4cb3b8a6a4d7894

Wurster, S., & Hagemann, C. (2020). Expansion of Renewable Energy in Federal Settings: Austria, Belgium, and Germany in Comparison. The Journal of Environment & Development, 29(1), 147–168. https://doi.org/10.1177/1070496519887488

Institute Business Informatics & Data Science

# Extended Virtual Acrophobia Treatment

DI Manuel KARNER, BA

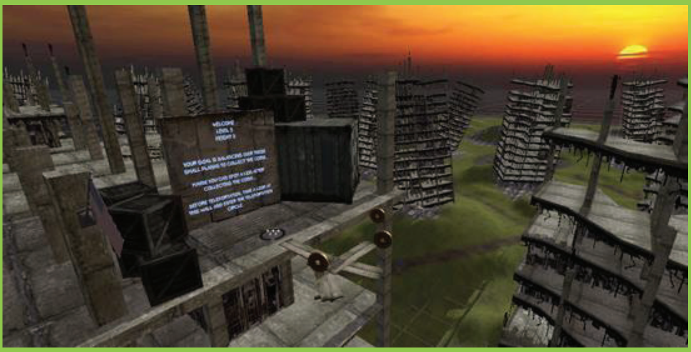


Fig. 1: **Final Application**

An example image of the final application showing the third and most difficult level.

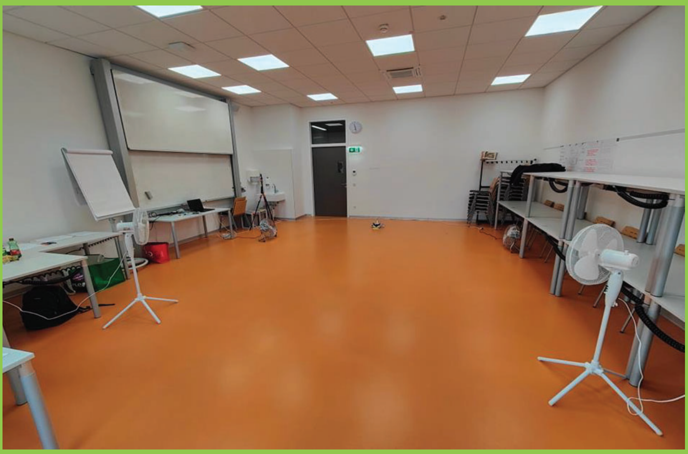


Fig. 2: **Test Setup**

The setup for the usability and user testing at FH Joanneum. The whole shown space was used by the application and wind was simulated via the fans.

## BACKGROUND / OBJECTIVE / INTRODUCTION

Virtual Reality is gaining significance, as the pandemic showed everyone, that access to certain health care institutions is and can be restricted. Therefore, using VR as an enabler and supporter in that sector is essential. Additionally, immersion is a big topic for VR and indispensable for a believable VR experience. [1]

## METHODOLOGY

The thesis is focused on the technical implementation of an extended virtual reality application via the use of additional senses and gamification. As a verification, several tests with people were conducted, which were followed by a questionnaire after the usage of the application.

## RESULTS

The results have shown that creating an application for exposure treatment in VR is no trivial task, as VR is not as mature as one would expect. Furthermore, the evaluation of the questionnaires showed that additional sensorial inputs play a big rule for the immersion of a user and gamification aspects are a valid approach to bring treatments closer to people.

## DISCUSSION

An important aspect of this research is, that the application was developed with the input of professionals, but it would take more feedback loops for a more practical implementation. Furthermore, the tests were conducted with a small sample size of only ten people. This is a direct implication that the results are not applicable for the general public.

### References

*[1] Berkman, M.I., Akan, E. (2019). Presence and Immersion in Virtual Reality. In: Lee, N. (eds) Encyclopedia of Computer Graphics and Games. Springer, Cham. https://doi.org/10.1007/978-3-319-08234-9_162-1*

# Generating Satellite Deepfakes with AI: Using Handmade Sandscapes as Input and Projection Mapping for Output

FH | JOANNEUM
Business Informatics

**Anderle** Fabian, **Bsc** **Schweighofer** Martin, **Bsc**

**Gürtler** Florian, **Bsc** **Uhl** Marie-Therese, **BA**

Supervisor: **Dipl.-Päd. Sprung** Gerhard, **MSc**

## Introduction

Our completed master's project delved into the use of generative adversarial networks (GANs) to produce deepfakes of landscapes. GANs are a type of artificial neural network that comprises two competing networks. One that generates realistic data (generative model) and the other that classifies that data as real or artificial (discriminator model). With repeated processing, GANs can create synthetic images that are nearly indistinguishable from real ones.

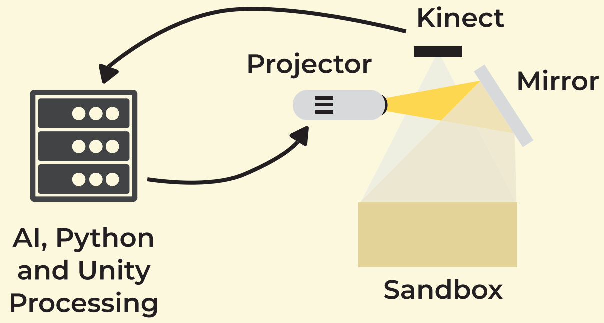
Source: A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta and A. A. Bharath, "Generative Adversarial Networks: An Overview," in IEEE Signal Processing Magazine, vol. 35, no. 1, pp. 53-65, Jan. 2018, doi: 10.1109/MSP.2017.2765202.

## Objective

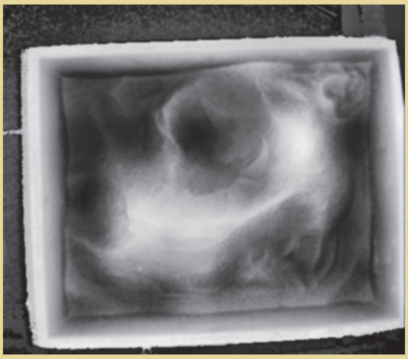
Our project aimed to create realistic deepfake landscapes by using GANs and integrate this feature into an existing sandbox project at FH JOANNEUM. Users can create unique landscapes by molding regular sand, which will serve as input for our GAN model. The GAN model will generate a fitting landscape. Ultimately, the goal of the project is to create an immersive experience by projecting the generated image back onto the sand.

## Methodology

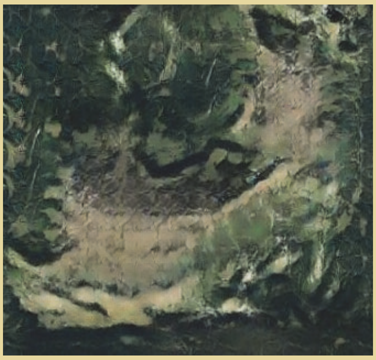
To achieve our objective, we utilized a Microsoft Kinect to capture the sand landscapes molded by users. The depth image acquired from the Kinect was then processed using Python and OpenCV to create a fitting input for our GAN model. The training data for this AI model was obtained with QGIS and after 300,000 iterations, we achieved the best deepfake result. With this input image, the GAN model is capable of generating a realistic satellite deepfake, which we projected back onto the sand using Unity and projection mapping. The combination of these techniques resulted in a captivating and immersive experience.



Kinect Depth Image



AI Generated Landscape



Result

# Hydrofoil Boat Design

Luis Trojer, Raphael Vierhauser and Gerhard Pirker

PEAKFOIL

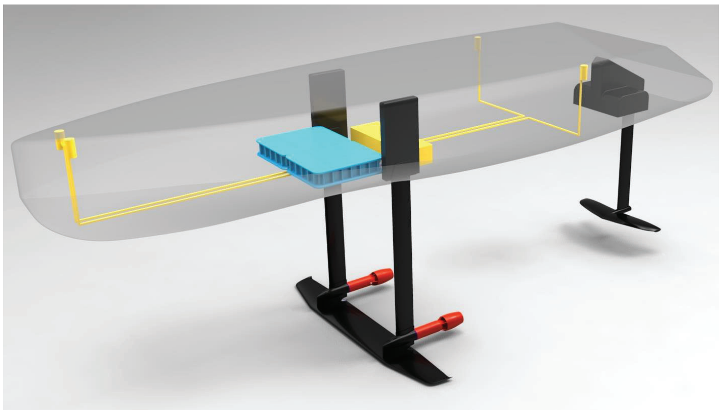
FH JOANNEUM University of Applied Sciences

FFG Forschung wirkt.

Flying has been a dream of humanity since ancient times, and boats are one of the oldest forms of transportation. Combining these two ideas into one small craft offers the potential for incredibly fast and efficient travel. In two separate Master's theses, the main wing of such a craft was designed using HPC methods, while the control system was set up and simulated in MATLAB.

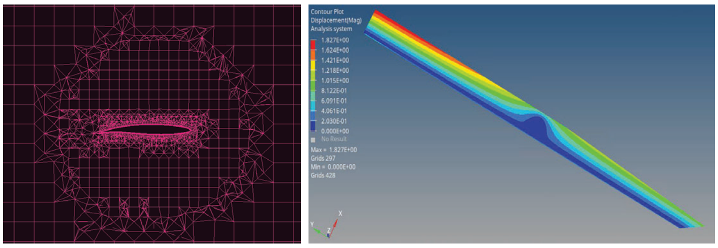
## ELECTRIFYING MARINE MOBILITY

Conducted at eFoiler GmbH in Vienna, the basis for two theses was the PEAKFOIL eFoil, an electrically-powered hydrofoil surfboard. Many of the eFoil components are planned to be incorporated into the boat project.



Rendering of the hydrofoil boat, showing the different components: flight control system (yellow), propulsion system (red), energy storage (blue) and foil system (black)

In the first study, the main wing of the boat was designed. A preliminary design process was conducted, basing a lot of the assumptions on data of the rear foil, which was already available [1]. Some other constraints were fixed during the concept phase of the boat. Dimensioning methods from aircraft design and model aircraft building were applied in this phase. The foil was then optimized in an iterative process, leading to its final shape. For said shape, a computerized model was generated using Computer Aided Design (CAD) tools [2], thus paving the way for analyses of different kinds.

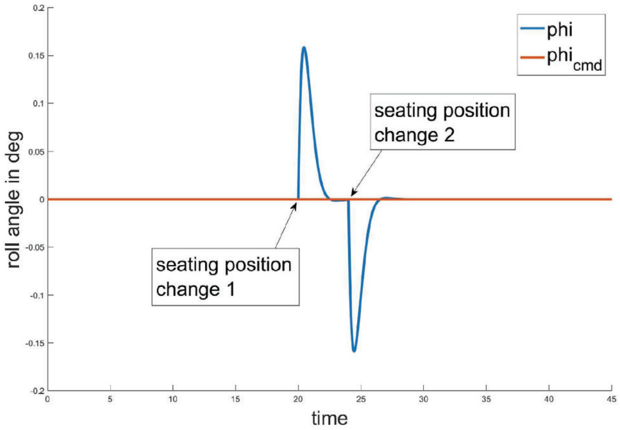


Cut through the mesh, showing the higher resolution around the foil, as well as the prism layers (left) and displacement plot of the aileron when extended (right)

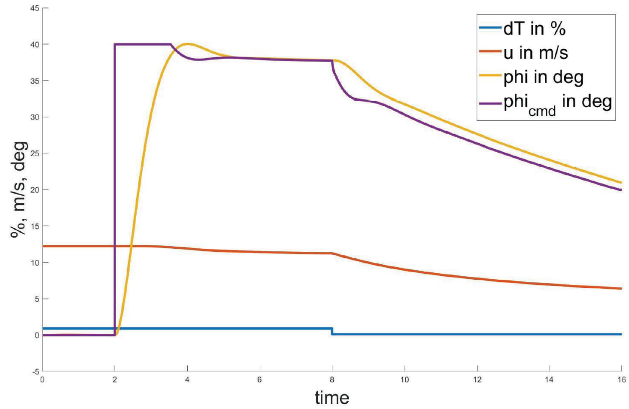
Different analyses were conducted using Computational Fluid Dynamics (CFD) [3], leading to a better understanding of the future performance and behavior of the foil in real-world conditions. Lastly, a Finite Element Analysis (FEA) [4] was set up to calculate the necessary composite layup of the finished product.

Table of References:
[1] Beaver, B., Zseleczky, J., "Full Scale Measurements on a Hydrofoil International Moth", U.S. Naval Academy Hydromechanics Lab, Annapolis, MD, March 2009.
[2] CATIA®, Software Package, Ver. 5-6 Release 2012, Dassault Systèmes, 2021.
[3] ANSYS Workbench, Software Package, Ver. 2021 R1, ANSYS Inc., 2021.
[4] HyperWorks®, Software Package, Ver. 2021, Altair Engineering Inc., 2021.
[5] MATLAB, Software Package, Ver. R2021b, The MathWorks Inc., 2021.

The fluiddynamic data from the first study was then fed into the development of the flight control system, which was the topic of the second study. As the center of gravity lies in between the two lifting surfaces, making it unstable, a sophisticated flight control algorithm is needed to stabilize the system. To simulate and develop the necessary control algorithm, the behavior of the vehicle was simulated in Matlab/Simulink [5]. The processes and techniques of flight mechanics were adapted, the optimal control theory was then used to develop an algorithm that stabilizes the unstable system. Sensors like an ultrasonic distance sensor, a static pressure sensor, an attitude and angular rate sensor are used to gather the states of the vehicle.



Comparison of the commanded and the actual roll angle during simulation with changes of the seating position by one of the drivers (20cm, 70kg)



Roll limit simulation, depending on the current aileron, flap and elevator positions the rollangle is decreased by the controller, regardless of the user input to ensure a safe flight envelope

Many thanks are due to DI (FH) Andreas Tramposch PhD, DI Dr. Bernd Messnarz and DI Gerhard Pirker for supervising the studies, as well as FH JOANNEUM, FFG and eFoiler GmbH for facilitating and financing the projects.

# IN-CABIN GAZE DETECTION USING NEURAL NETWORKS ON MOBILE DEVICES

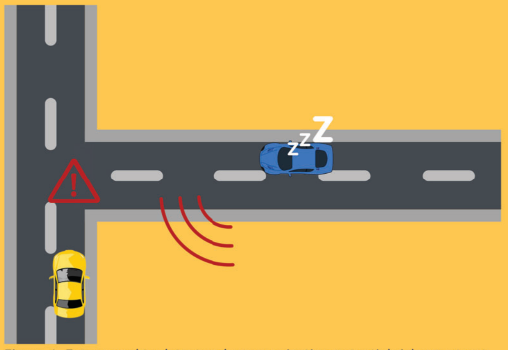


Figure 1. Future goal to detect and communication potential risks on street caused by distraction of the driver

LAURA GREGORC
Author

MAG. DR. ANDREAS WINDISCH
DR. ELEM GÜZEL KALAYCI
Supervisors

## 1 MOTIVATION

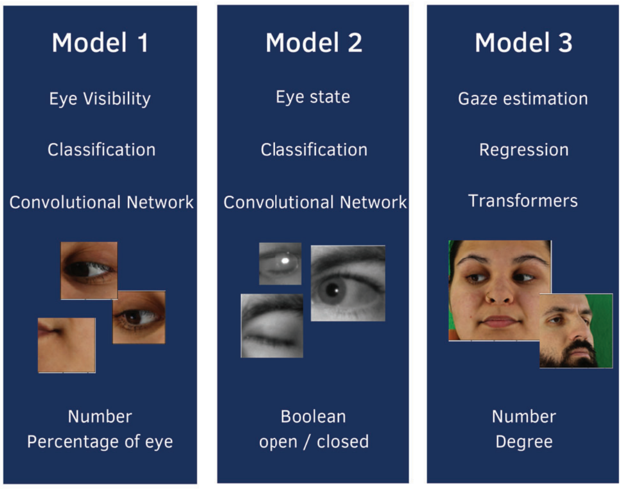
In the automotive industry the use of **assistive technologies** is becoming more popular. Not only in the multimedia field, but also to **improve driving** and **preventing accidents**. One future goal of the research center Virtual Vehicle is to detect whether a driver is distracted by analyzing the driver's eye.

By contributing in a former CORDIS project [1] to create a platform for artificial intelligence systems, a pipeline including a series of models was developed and trained on datasets. The goal of the thesis is to improve the models and make it **available on mobile devices in the driver's cabin** with real-time video data.

## 2 MODELS

From a technical point of view, a combination of two convolutional networks and one transformer is used. According to the current findings of the paper **Gaze estimation using transformer** [2], transformers are very useful to solve **regression problems** in the field of computer vision and object detection.

While driving, a video is recorded in real-time and segmented into frames. A bounding box is added to make sure the image includes a face. The **current state of the pupils** can be pictured quiet well with the three measurements of the output vector. It has to be admitted, that there is more data needed to detect distractions of the driver. Despite it, the gaze and area of interest brings valuable information [5].

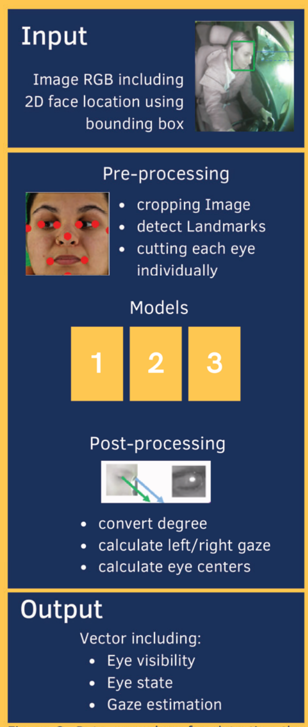


Model 1
Eye Visibility
Classification
Convolutional Network
Number
Percentage of eye

Model 2
Eye state
Classification
Convolutional Network
Boolean
open / closed

Model 3
Gaze estimation
Regression
Transformers
Number
Degree

Figure 2. Overview of the model architecture for detecting eye gaze



Input

Image RGB including 2D face location using bounding box

Pre-processing
- cropping Image
- detect Landmarks
- cutting each eye individually

Models

1 2 3

Post-processing
- convert degree
- calculate left/right gaze
- calculate eye centers

Output

Vector including:
- Eye visibility
- Eye state
- Gaze estimation

Figure 3. Data procedure for detecting the area of interest of a driver on a edge device

## 3 DATASETS

In the previous project, the models were trained on **publicly available** datasets and produced an accuracy around **99%**.

- ETH-XGaze [3]
  - over 1 mio. observations
  - 18 digital SLR cameras
- MRL Eye [4]
  - approx. 15k observations
  - Intel RealSense 300 sensors
  - IDS Imaging sensor

**Limitations** of the current datasets:
- missing continous head positions
- only preprocessed image data instead of video material

By Virtual Vehicle, a datasets of **frontal facing recordings** on car rides is provided. Each observation contains up to **two hours** material. When cutting this videos into segments, a useable dataset can be created.

## 4 PROTOTYPE

Technical requirements:
- platform independent
- Cross-platform framework
- Flutter and Tensorflow Lite [6]
- processing in real-time
- saved to the local storage of the device for further analysis

Interface:
- start button to press when the driver starts the test ride
- a textbox which shows recently calculated gazes
- stop button when the test scenario is finished

Scope of the app:
- in-house test environments
- no focus on useablity and design
- self-explanatory and easy to use

## 5 RESEARCH QUESTIONS

1. Can a **more realistic dataset** be used to overcome gaze detection errors and the problem of discrete head angles?
2. What are additional **performance metrics** for every single one of the three models?
3. How would a **prototype** developed with Tensorflow Lite and Flutter look like for such a combination of models?
4. How can the preprocessing step be implemented **on the mobile device**?
5. How **robust** are the resulting models, embeded in the Flutter prototype, with **real-time** recorded data?

## REFERENCES

[1] **Platform for Open Development of Systems of Artificial Intelligence** https://cordis.europa.eu/project/id/732204

[2] Cheng, Yihua, and Feng Lu. Gaze estimation using transformer. 2022 26th International Conference on Pattern Recognition (ICPR). IEEE, 2022

[3] Xucong Zhang, Seonwook Park, Thabo Beeler, Derek Bradley, Siyu Tang, and Otmar Hilliges. Eth-xgaze: A large scale dataset for gaze estimation under extreme head pose and gaze variation. In European Conference on Computer Vision, 2020

[4] Fusek, R. Pupil localization using geodesic distance. In Advances in Visual Computing: 13th International Symposium, ISVC 2018, Las Vegas, NV, USA, November 19–21, 2018, Proceedings 13 (pp. 433-444). Springer International Publishing, 2018

[5] Kapitaniak, B., Walczak, M., Kosobudzki, M., Jóźwiak, Z., & Bortkiewicz, A. Application of eye-tracking in drivers testing: A review of research. International journal of occupational medicine and environmental health, 28(6), 2015

[6] Singh, A., & Bhadani, R. Mobile Deep Learning with TensorFlow Lite, ML Kit and Flutter: Build scalable real-world projects to implement end-to-end neural networks on Android and iOS. Packt Publishing Ltd., 2020

## ACKNOWLEDGEMENT

The publication was written at Virtual Vehicle Research GmbH in Graz, Austria. The authors would like to acknowledge the financial support within the COMET K2 Competence Centers for Excellent Technologies from the Austrian Federal Ministry for Climate Action (BMK), the Austrian Federal Ministry for Labour and Economy (BMAW), the Province of Styria (Dept. 12) and the Styrian Business Promotion Agency (SFG). The Austrian Research Promotion Agency (FFG) has been authorised for the programme management.

FH JOANNEUM University of Applied Sciences
Data and Information Science

Institute Business Informatics & Data Science

FH | JOANNEUM
University of Applied Sciences

# Interactive Sandscape - explore the interactivity of height modeling by shaping your own landscape with kinetic sand

Sandra SCHADENBAUER*, Alexander NISCHELWITZER

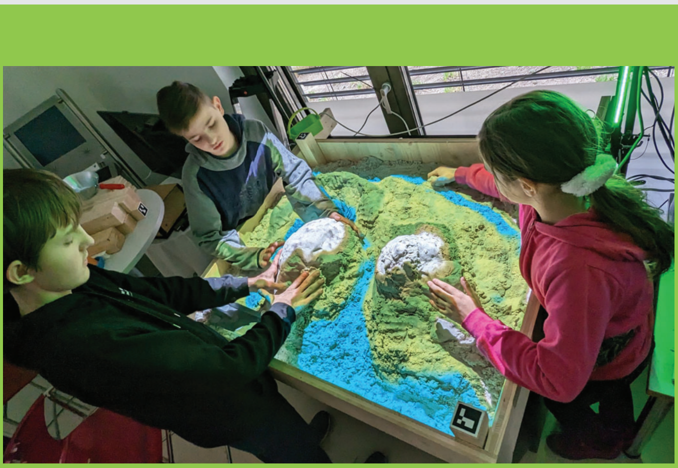


Fig. 1: **Sand table**
Children interacting with the sand table
Image source: Alexander Nischelwitzer

This study examines the potential to detect, calculate and show elevation models on kinetic sand in real time. Kinetic sand is known for its ability to hold its shape and be molded into different forms. The Microsoft Azure Kinect can capture detailed 3D images and with our software, the elevation model of the sand is calculated and five layers (from rivers and lakes filled with fishes to snow-covered mountains) are shown directly as a projection on the kinetic sand.

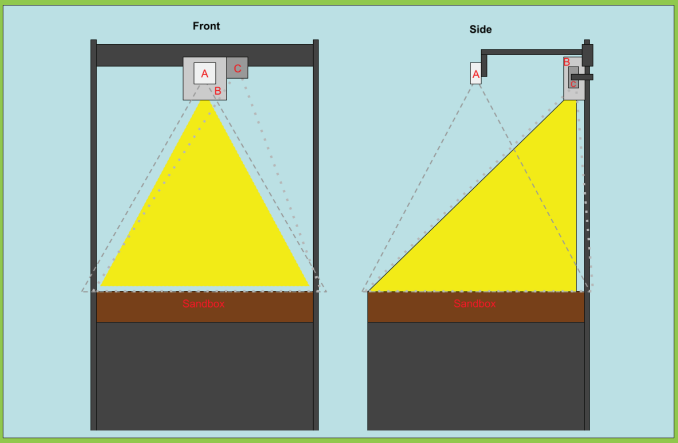


Fig. 2: **Sand table arrangement**
A: Azure Kinect, B: Beamer, C: Camera (for remote calibration)
Image source: Sandra Schadenbauer

Depending on the measured biodiversity of the molded terrain a specific quantity of birds is flying over the sand, as well as a light scale indicates the biodiversity from 0 to 100 percent.

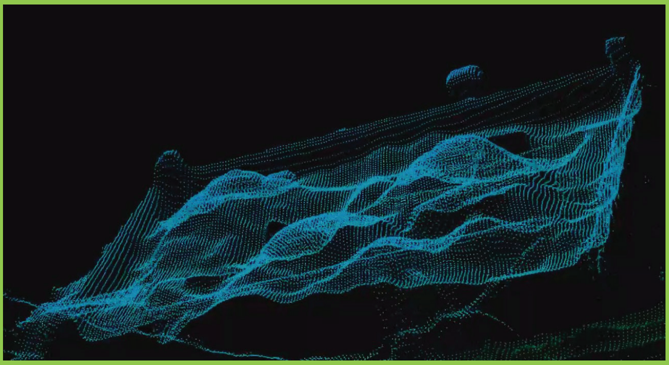


Fig. 3: **Digital elevation model**
Real time 3D model of the sand table processed by Azure Kinect
Image source: Screenshot of "Azure Kinect Viewer" facing the sand table

Another key feature of our development is the possibility to remote calibrate the challenging interplay between Microsoft Kinect Azure and the projection. Therefore, a calibration tool was developed, which allows fine adjustments of detection and projection with the help of a web cam and a remote access and remote-control computer software.

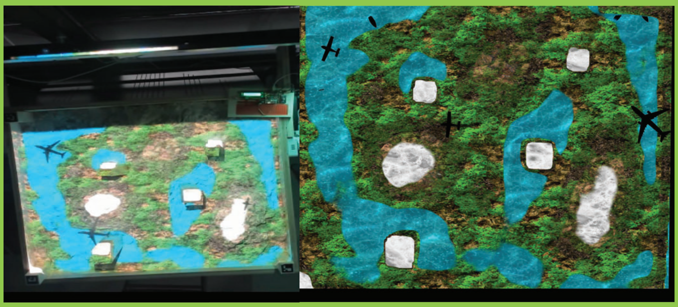


Fig. 4: **Remote calibration view**
Left: webcam view, right: program view. Wooden blocks are used to match the projection and the real sand landscape during the calibration process.
Image source: Sandra Schadenbauer

* Corresponding author: sandra.schadenbauer@fh-joanneum.at

# Investigation of Process Influences on the Steel Composition and Inclusion Modification in Secondary Metallurgy using Diverse Statistical Learning and Data Mining Methods

Daniel Kavic[1,2,3], Michael Melcher[1], Christian Bernhard[2] and Roman Rössler[4]

[1]Institute of Business Informatics and Data Science, FH Joanneum – University of Applied Sciences, Austria •
[2]Chair of Ferrous Metallurgy, University of Leoben, Austria • [3]K1-Met GmbH, Austria • [4]voestalpine Stahl Linz GmbH, Austria

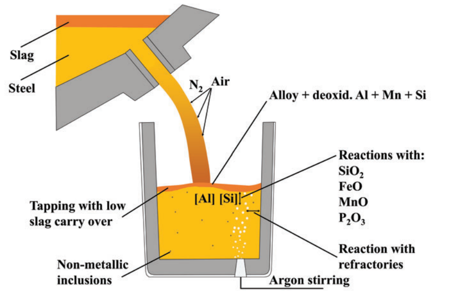
## Introduction

Secondary metallurgy (SecMet) significantly influences the final alloy composition and non-metallic inclusion modification of steel. **Steel composition affects castability** and is decisive for the final product's subsequent **mechanical properties** (e.g., tensile strength, ductility). In general, non-metallic **inclusions harm product quality** and lower the working life of a component under dynamic load or accelerate the material's corrosion. Therefore, liquid steel must satisfy precise compounding specifications before it is cast. The provided data originate from the following two sources: Firstly, the SecMet process steps, which start with the **tapping of the basic oxygen furnace (BOF)** and end with the **ladle furnace (LF) treatment**, are **continuously monitored** and recorded. Secondly, **steel samples** are taken from the melt at specific times during the process, and their **chemical composition** is **determined** through optical emission spectrometry. A detailed study of the process and steel composition data to gain new knowledge for future data-driven models are being performed during the ongoing research. A data evaluation process using **descriptive statistical** methods was carried out. Subsequently, first **statistical learning** and **data mining methods** (MLR [1], PLS [2]) have been applied to predict the composition of the liquid steel.

## Problem Statement

### Phenomena Occuring during the Tapping Process at the BOF

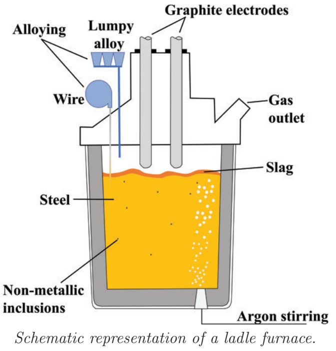
- Entrainment of (FeO)-rich slag must be kept at a minimum - Al, Si oxidation and P recovery can occur.
- Absorption of $N_2$ from the atmosphere.
- Alloying and addition of treatment slag for the SecMet.



*Schematic representation of the occurring events at the tapping process of the BOF after [3].*

### The Ladle Furnace (LF) – Typical Treament Options and Sequence for the Liquid Steel

- Temperature measurements [°C]
- Argon (Ar) stirring quantity and duration [$m^3/h$]
- Heating amount and duration [kWh]
- 32 different alloy additions [kg]



*Schematic representation of a ladle furnace.*

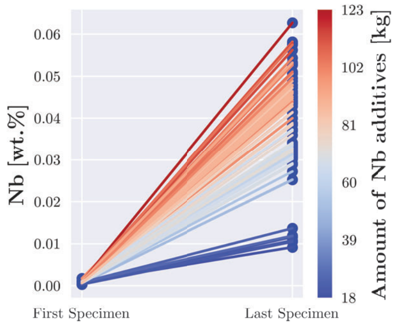
Approx. 9000 heats, which were treated exclusively by a ladle furnace, are available for investigation. The following steps describe the progress up to the present stage:

1. Comprehensive data preparation was performed to completely cleanse the data of artifacts and remove inconsistencies. In addition, a final data set was created that combines both process data and chemical analysis.
2. Descriptive statistic, in which the dispersion and location measures of all alloying elements and process related interventions were investigated.
3. Due to the high variability of the process control, a simplified data set was generated. The sequential order of the process influences is not taken into account in this new data set.
4. Development and evaluation of the first statistical learning and data mining models.

## Results - Descriptive Statistics

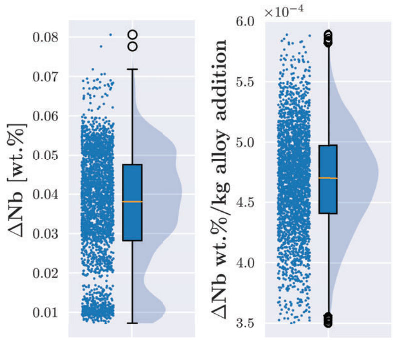
### Exemplary presentation of the results on the basis of niobium

2759 grades have been alloyed with niobium and show a very significant increase. FeNb serves as the main alloying agent.



*Initial niobium (Nb) content during tapping process (left) and after the LF treatment (right).*

The figures clearly show that the addition of FeNb is measurable. In addition, the increases vary greatly depending on the operating conditions and the composition of the steel.



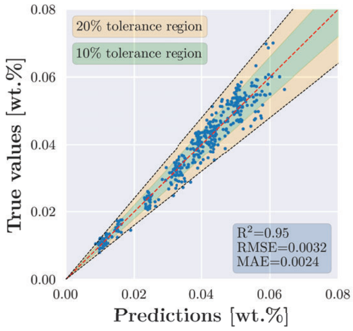
*Absolute change in niobium (Nb) content between the first and last specimen (left) and the relative change per kg FeNb addition (right).*

## Train- and Validation Data Set

It was important to ensure that all grades were represented in each data set. A 70 - 30% split was performed.

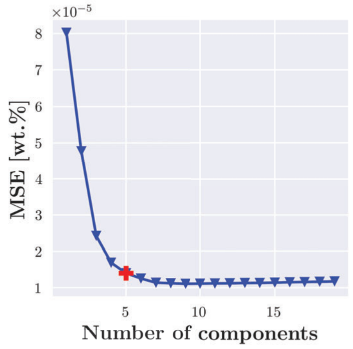
## Results - MLR and PLS

In the **Multiple Linear Regression (MLR)**, variable selection was performed based on a forward selection. Two independent variables were selected: Initial content of niobium and the alloy additive FeNb.

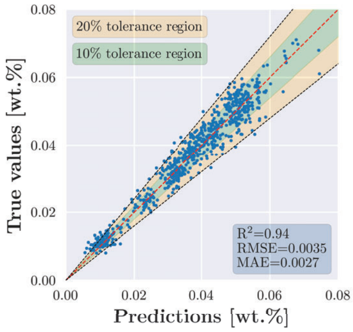


*True vs. predicted values for niobium (Nb) using MLR.*

**Partial Least Squared (PLS)** provided 5 components as a result of the component variation process.



*MSE against the number of components used in PLS.*



*True vs. predicted values for niobium (Nb) using PLS.*

Both methods show excellent predictive power. However, MLR is more suitable because the coefficients are easily extractable and thus can be used for the thermodynamic models.

## Summary and Outlook

The described procedure and the evaluation on the basis of a multiple linear or PLS model have already been carried out on other elements (see table on the right). Excellent results with $R^2$ values of 0.9 and more could be achieved for most of the studies. Future steps will be the use of further statistical learning and data mining methods (e.g., Random Forest [4], PCR [5], Ridge [6] and Lasso [7] Regression, ...) to predict the alloy composition. Finally, the results of the models will be compared. After successfully predicting the alloying elements, the focus will lie on the inclusion modification.

*Table: Summary of the results to date.*

| * in wt.-% | **MLR** | | **PLS** | |
|---|---|---|---|---|
| **Element** | $R^2$ | MAE* | $R^2$ | MAE* |
| C | 0.96 | 3e-3 | 0.96 | 4e-3 |
| Mn <1.25* | 0.99 | 0.013 | 0.98 | 0.0120 |
| Nb | 0.93 | 3e-3 | 0.94 | 0.028 |
| Ti | 0.89 | 2e-3 | 0.75 | 0.003 |
| Cr | 0.92 | 3e-4 | 0.65 | 0.032 |

At this point I would like to thank my supervisors for their contribution and the industrial partners for making this work possible.

## Contact

Daniel Kavic, BSc
Junior Researcher
Chair of Ferrous Metallurgy
University of Leoben
8700 - Leoben, Austria
Email: daniel.kavic@unileoben.ac.at
Phone: +43 680 232 242 8

## References

[1] J. D. Jobson, in *Multiple Linear Regression*, Springer, 1991, pp. 219–398.
[2] A. Höskuldsson, *Joual of Chemometrics*, 1988, **2**, 211–228.
[3] K. O., *Schulungsunterlagen VAI*, 2004.
[4] L. Breiman, *Machine Learning*, 2001, **45**, 5–32.
[5] I. T. Jolliffe, in *Principal Components in Regression Analysis*, Springer New York, New York, NY, 1st edn., 1986, pp. 129–155.
[6] A. E. Hoerl and R. W. Kennard, *Technometrics*, 1970, **12**, 55–67.
[7] R. Tibshirani, *Journal of the Royal Statistical Society. Series B (Methodological)*, 1996, **58**, 267–288.

## Support

This work was supported by the industrial partners voestalpine Stahl Linz GmbH and K1-Met GmbH

# Landform Detection on Mars Using a U-Net Architecture

**Florian Auer-Welsbach[1], Andreas Windisch[2,3,4], Giacomo Nodjoumi[5]**

[1]FH-Joanneum, Data Science and Artificial Intelligence, Department of Applied Computer Sciences, AT
(florian.auerwelsbach@edu.fh-joanneum.at)
[2]Know-Center GmbH, Inffeldgasse 13, 8010 Graz, AT
[3]Graz University of Technology, Institute of Interactive Systems and Data Science, Inffeldgasse13, 8020 Graz, AT
[4]Department of Physics, Washington University in St. Louis, MO 63130, USA
[5]Constructor University Bremen gGmbH, Bremen, DE

## 1. Overview

### 1.1 Abstract

The detection and classification of landforms on planetary surfaces is a time-consuming task which deeply relies on expert knowledge. Such a process can be partially automated and optimized in a resource-efficient way using image segmenting algorithms [1]. By classifying the surface into different landforms, such as volcanic craters, asteroid impacts, dunes, and more, several analyses can be performed, for instance the widely used crater counting age estimation method [3].

In this work, a multi-class image segmentation algorithm is presented, which is based on a U-Net convolutional neural network architecture. U-Nets classify each pixel of a given input image and can thus produce segmentation masks for various landforms. Given that enough labeled data is available, such a classifier can replace manual detection and classification, thereby saving resources by providing a fast method for landform detection.

### 1.2 Research questions

**Main questions:**
Considering first a binary segmentation and then multi-class object segmentation and classification, how well do these models perform?

**Sub questions:**
How can we use meta data, image background information, longitude/latitude information, ect. to improve the model´s performance?
What data augmentation strategies prove useful to enrich the data set?

## 2. Data

- 2375 total images
- 1914 pits (4 different types)
- 461 craters
- Data augmentation: image with different spatial resolution

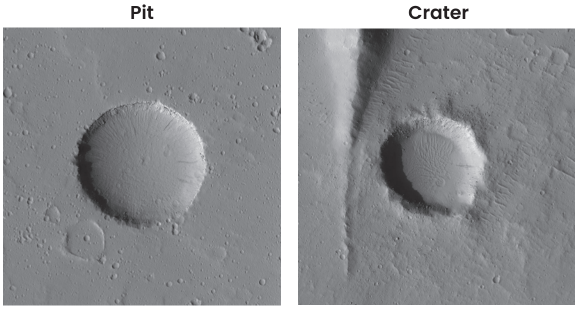


Figure 2: Comparison between a crater and a pit [2].

## 3. Model architecture

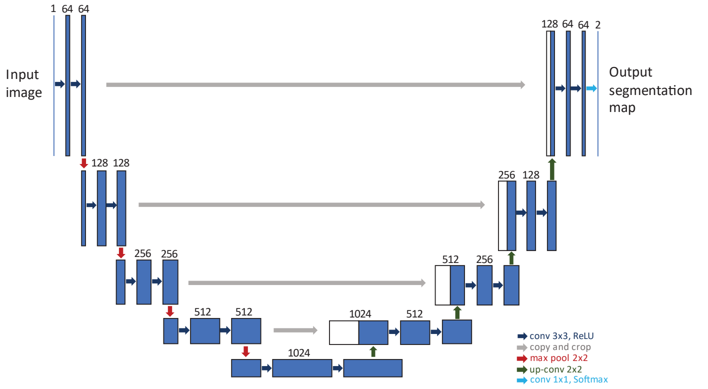
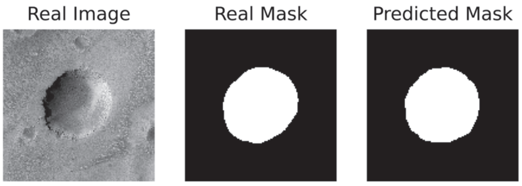


Figure 1: Proposed U-Net model visualized [4].

## 4. First results

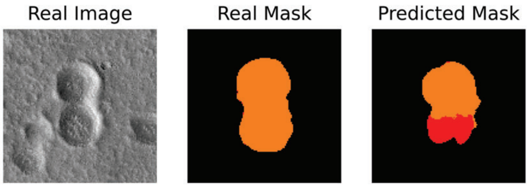
### 4.1 Binary segmentation



**F1-Score**
Background: 0.99
Object: 0.92

Figure 3: U-net trained on a binary segmentation problem. Comparison between the real and the predicted mask.

### 4.2 Crater/Pit segmentation



**F1-Score**
Background: 0.99
Pit: 0.88
Crater: 0.77

Figure 4: U-net trained on a multi-class segmentation problem. Comparison between the real and the predicted mask. (crater in orange and pit in red)

Post-processing: Majority voting when multiple objects occur in one object → Crater wins in Figure 4

## 5. Outlook

- Adding additional landform types
- Construct a segmentation pipleline for large images
  - Break down the image into smaller images
  - Use a sliding window and predict mask
  - Get the final prediction with IOU
- Split segmentation and classifcation
  - Use two distict networks
  - U-Net for segmentation
  - Vanilla CNN or YOLO for classification

## 6. References

[1] C. Lee, J. Hogan (2017). Automated crater detection with human level performance. Computers & Geosciences. Volume 147. https://doi.org/10.1016/j.cageo.2020.104645.
[2] G. Nodjoumi, R. Pozzobon, F. Sauro, A. P. Rossi (2022). DeepLandforms: A Deep Learning Computer Vision toolset applied to a prime use case for mapping planetary skylights. Earth and Space Science. Volume 10. Issue 1. https://doi.org/10.1029/2022EA002278.
[3] T. Platz, G. Michael, K. L. Tanaka, J. A. Skinner, C. M. Fortezzo (2013). Crater-based dating of geological units on Mars: Methods and application for the new global geological map. Icarus. Volume 225. Issue 1. https://doi.org/10.1016/j.icarus.2013.04.021.
[4] O. Ronneberger, P. Fischer, T. Brox (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. LNCS. Volume 9351. Pages 234-241. https://doi.org/10.48550/arXiv.1505.04597.

# Modeling Rebound Effects from a Shift to Electric Vehicles in Private Mobility in Styria

Lisa Göberndorfer BSc
Environmental System Sciences / Climate Change and Environmental Technology
Graz University of Technology and University of Graz

## Introduction

This master's thesis is part of the research project **Charging Rushhour** conducted between November 2021 and October 2022 at the Institute of Environmental Systems Sciences at the University of Graz. The project identified the impact of the charging demand of a fully **electric private vehicle fleet** on the electric grid in Styria. It employed an agent-based model written in Python[1]. Charging Rushhour was funded by the Government of Styria and the results will be published in two scientific papers.

The **first publication** covers the agent-based model as well as immediate effects of the charging demand of an entirely electric vehicle fleet on the grid[1]. This paper is currently in review, whereas the **second publication** is in the writing stage. It includes a scenario analysis of different policy measures to reduce the peak electricity demand present at the "charging rush hour"[2]. If combustion engine vehicles in private mobility are fully replaced by battery electric vehicles, there will be considerable electricity demand of up to 80% above what the grid currently needs to handle[1]. Different policy measures can lead to peak shift and demand distribution[2].

The aim of this **master's thesis** is to complement the conducted research and determine additional factors requiring consideration when enabling a sustainable mobility system on a regional scale. The agent-based model from the research project will be adapted and expanded to answer the research question: Which rebound effects arise from this drastic shift and how to mitigate them? The time span and physical area considered in this thesis are identical with Charging Rushhour, i.e., in the federal state of Styria until the year 2040. Nevertheless, the development of the mobility system until 2030[3,4] and **2035**[5], respectively, is highly relevant for this thesis as significant changes within the mobility system need to take place within the next years. This urgency is justified by the decision of the European Commission to stop the licensing of new combustion engine vehicles by the year 2035 with intermediate targets for 2030[5].

The main goal of this thesis is to model different possible **rebound effects** resulting from this drastic shift and to suggest appropriate **policy measures** to mitigate or prevent them. Affected areas beyond the electric grid such as traffic volume, congestion and shifts in the mode of transport will be considered. The impact of different demographic factors represented in the survey on the driving behavior and charging demand during the day will be investigated. Sources include the survey Österreich Unterwegs 2013/14[6], its result report[7] and the extension report on mobility behavior[8] available through the Federal Ministry for Climate Action, Environment, Energy, Mobility, Innovation and Technology.

## Methodology

The master's thesis will contain a thorough explanation of the agent-based model created for the research project as well as any amendments thereof. Furthermore, to answer the research question different **scenarios** will be implemented to investigate changes within the mobility system including rebound effects. The scenarios are then compared and analyzed regarding their robustness, relevance and interconnectedness. The research is concluded by a sensitivity analysis of the input parameters. Finally, possible mitigation and prevention approaches are suggested in the form of policy measures.

The agent-based model created for Charging Rushhour is based on real-world data compiled from about 38,000 participants of the survey **Österreich Unterwegs 2013/2014**. It therefore contains anonymized sociodemographic data including municipality codes, household sizes, travel modes and vehicle sizes, among others[6]. Additionally, Austrian electricity demand data accurate to the minute is obtained from operating statistics[9].

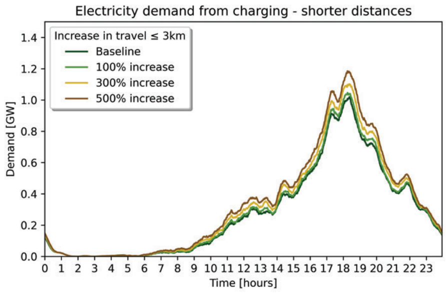
The **agent-based model** is written in Python and consists of agents that are representative of the population of Styria. The agents travel various numbers of paths per day, of which the frequency, duration, length, starting point and destination are derived from the survey data[6]. The agents then charge their electric vehicles at a predetermined time of day. This is assumed to be usually after the last travel of the day[1], however other scenarios are also possible[2]. Charging is considered to occur at a constant charging rate and the **charging process** is finished once the electric energy used for traveling is recovered[1,2]. The daily overall electricity demand accurate to the minute[9] is used as a basis to determine the effect of the charging demand on the grid. Styrian municipalities are grouped into urban, suburban and rural areas to compare their charging demand[1]. Additionally, some percentage of agents can exhibit a scenario-dependent individual behavior such as charging their vehicles at different times or working from home[2].

## Acknowledgements

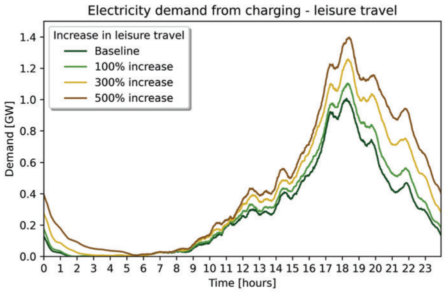
I would like to thank Ass.-Prof. Dr. Georg Jäger from the Institute of Environmental Systems Sciences for the supervision and guidance in the research project Charging Rushhour as well as in this master's thesis. As a student assistant in the research project, I appreciated the professional work atmosphere and collegiality very much. Therefore, I would also like to thank my colleague Milica Savanovic for the cooperation in the writing process of the scientific publications. Additionally, I would like to extend my gratitude towards Ass.-Prof. Dr. Georg Jäger and Christian Jin for feedback on this poster.

## Current status and preliminary results

The model currently yields results based on the effects of different **behavioral changes** on the charging demand. The results are displayed as an electricity demand curve throughout an entire day with the demand given in gigawatts. Charging times are slightly randomized while preserving the overall data structure to prevent artefacts. Model runs with the same parameters therefore do not produce identical results[1,2].



The **electricity demand peak** in the late afternoon and early evening hours is considerable if individual driving behavior and therefore assumed charging behavior remains constant[1,2]. The increase in travel of shorter distances of up to three kilometers only slightly increases the overall electricity demand as depicted in the top diagram. In the diagram below, the increase in leisure travel does significantly increase the charging demand. This leads to a considerably higher electricity demand in the evening and at nighttime if no counter-measures are taken. Both scenarios display possible behavioral changes associated with rebound effects such as the increased usage of battery electric vehicles on certain travels. This could be explained by reduced travel costs, wider accessibility or electric vehicles being perceived as more environmentally friendly.



## Research gap and outlook

The changes needed to restructure the mobility system and the urgency at which they must take place are considerable motivation for this thesis. There does not seem to be substantial research on rebound effects related to private electric mobility in Styria yet[3,4]. On the **regional and local scale**, rebound effects are not mentioned in mobility strategies[3,10] and are consequently presumed to be currently not under investigation. This presents a research gap which this thesis aims to fill through agent-based modeling with a focus on individual behavior, resulting effects and suggestions for policy measures.

This master's thesis is currently in its early stages, nevertheless it is clear from the data sources that there is a lack of current high-resolution behavioral data[3,6,9,10]. The dataset of the survey Österreich Unterwegs 2013/2014 thus remains integral to this agent-based modeling approach despite its data not reflecting the increased usage of electric vehicles within the last ten years.

## References

[1] Göberndorfer et al., Charging rush hour (submitted to SAGE Open, currently in review)
[2] Savanovic et al., Mitigating the Charging rush hour (manuscript in preparation)
[3] Amt der Steiermärkischen Landesregierung, Landesstrategie Elektromobilität Steiermark 2030, 2016.
[4] Mobilitätsmasterplan 2030 für Österreich, Bundesministerium für Klimaschutz, Umwelt, Energie, Mobilität, Innovation und Technologie, 2021.
[5] European Parliament, Fit for 55: zero CO2 emissions for new cars and vans in 2035, February 14th, 2023.
[6] Bundesministerium für Klimaschutz, Umwelt, Energie, Mobilität, Innovation und Technologie, Methodenberichte und anonymisierte Datensätze, https://www.bmk.gv.at/themen/verkehrsplanung/statistik/oesterreich_unterwegs/methodendaten.html, accessed March 31st, 2023.
[7] Tomschy R et al., Österreich unterwegs 2013/2014: Ergebnisbericht zur österreichweiten Mobilitätserhebung Österreich unterwegs 2013/2014, 2016.
[8] Bundesministerium für Verkehr, Innovation und Technologie, Mobilitätsverhalten von Frauen und Männern in unterschiedlichen Lebenslagen, 2017.
[9] Energie-Control Austria für die Regulierung der Elektrizitäts- und Erdgaswirtschaft, Strom - Betriebsstatistik 2021, https://www.e-control.at/statistik/e-statistik/archiv/betriebsstatistik/2021, accessed March 31st, 2023.
[10] Stadt Graz, Mobilitätsstrategie der Stadt Graz, https://www.graz.at/cms/beitrag/10191191/8038228/Mobilitaetsstrategie_der_Stadt_Graz.html, accessed March 31st, 2023.

# Vorhersage von Radsatz-Reprofilierungen durch künstliche Intelligenz

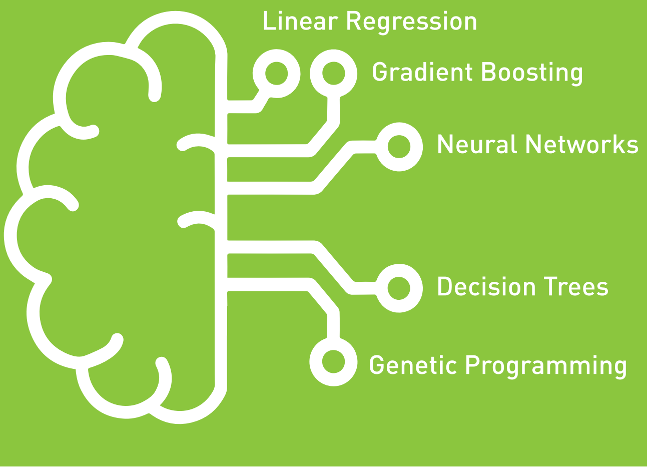
ING. FLORIAN LEBER, BA • BACHELORARBEIT

## EINLEITUNG

Die Instandhaltung von Radsätzen ist einer der sicherheitskritischsten und kostenintensivsten Vorgänge von Eisenbahnsystemen und muss gut geplant werden. Daher haben die Überwachung und Bewertung des Radsatzzustands einen großen Einfluss auf die Wartungsstrategien und deren Kosten.

Die unterschiedlichen Zustände können zu unterschiedlichen Wartungsereignissen führen. Eines der wichtigsten Wartungsereignisse ist die Reprofilierung der Radsätze. Es wurde die Hypothese aufgestellt, dass der Zustand und die Notwendigkeit einer Reprofilierung auf Basis historischer Messdaten in Kombination mit den aufgezeichneten Wartungsereignissen durch Data-Science Modelle vorhergesagt werden können. Als Vorgehensmodell wurde hierfür das CRISP-DM gewählt, welches einen strukturierten Ablauf für Data-Science-Projekte bietet.

## ALGORITHMEN



Linear Regression

Gradient Boosting

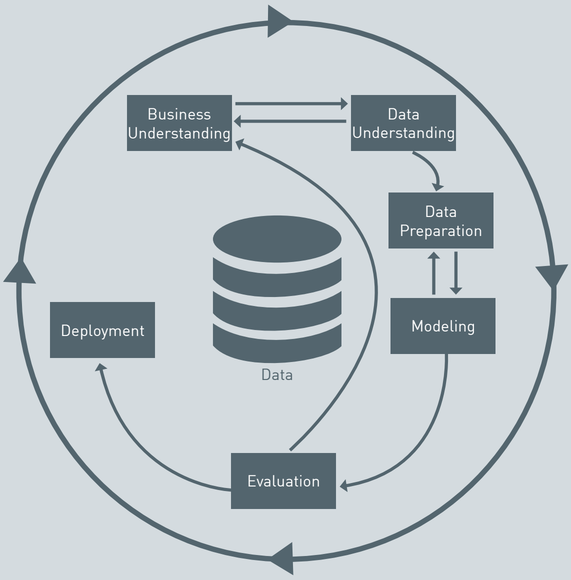
Neural Networks

Decision Trees

Genetic Programming

## CRISP-DM ALS VORGEHENSMODELL

Das CRISP-DM definiert einen Regelkreis, der sich zu Beginn ganz klar mit dem Verständnis des Geschäftsumfeldes und der Daten beschäftigt. Erst nach diesen essenziellen Schritten wird mit der Datenaufbereitung begonnen. Auf Grundlage der Daten werden verschiedene Algorithmen angewandt. Die Ergebnisse werden evaluiert und als Modell zur Verfügung gestellt.



Business Understanding

Data Understanding

Data Preparation

Modeling

Evaluation

Deployment

Data

## TECHNOLOGIEN

N
TensorFlow
scikit
learn

## ERGEBNISSE

Die Genauigkeit der Verschleißprognosen übertraf die Akzeptanzgrenze der damit arbeitenden User. Die Ergebnisse lassen sich daher wissenschaftlich und praktisch verwerten. Einerseits wurde gezeigt, dass durch den systematischen Ansatz die großen Datenmengen sinnvoll erfasst, verarbeitet und transformiert werden konnten, andererseits lieferten die Modelle nicht-triviale Erkenntnisse für die Lebensdauervorhersage.

**Quellen:**

Chapman et al. (2000), CRISP-DM 1.0: Step-by-step data mining guide

Bevan et al. (2013), Optimisation of wheelset maintenance using whole-system cost modelling. Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit 227 (6), 594–608

Zeng et al. (2021), An Optimal Life Cycle Reprofiling Strategy of Train Wheels Based on Markov Decision Process of Wheel Degradation. IEEE Transactions on Intelligent Transportation Systems, 1–11

BOOM
SOFTWARE

www.scientific-computing.at
E: scientific-computing@fh-joanneum.at

FH JOANNEUM | Austria | Styria
T: +43 (0)316 5453-0
E: info@fh-joanneum.at

www.fh-joanneum.at
www.instagram-com/fh_joanneum
www.facebook.com/fhjoanneum