Tri Rhena Tech

Interreg
Oberrhein | Rhin Supérieur

Kofinanziert von
der Europäischen Union
Cofinancé par
l'Union Européenne

ensisa
École nationale supérieure
d'ingénieurs sud Alsace

IRIMAS UNIVERSITÉ HAUTE-ALSACE
Institut de Recherche en Informatique, Mathématiques, Automatique et Signal

UNIVERSITÉ HAUTE-ALSACE

UR-AI2023
THE UPPER-RHINE ARTIFICIAL INTELLIGENCE SYMPOSIUM

ARTIFICIAL INTELLIGENCE
FOR TIME SERIES, ROBOTICS AND BEYOND

EDITED BY

JONATHAN WEBER

JEAN-PHILIPPE
LAUFFENBURGER

COLLECTION OF ACCEPTED PAPERS OF THE SYMPOSIUM
AI STUDENT SCHOOL SYNTHESIS
MULHOUSE, 16-17 NOVEMBER 2023

UR-AI 2023: Upper Rhine Artificial Intelligence Symposium

16-17 Nov 2023
Mulhouse
France

The Upper-Rhine Artificial Intelligence Symposium

UR-AI 2023

AI for Time Series, Robotics and Beyond

Conference Chairs

Jean-Philippe Lauffenburger, ENSISA-IRIMAS
Jonathan Weber, ENSISA-IRIMAS

Program Committee

Franz Quint, Karlsruhe University of Applied Sciences
Thomas Lampert, Télécom Physique Strasbourg
Ulrich Mescheder, Furtwangen University of Applied Sciences
Maja Temerinac-Ott, Furtwangen University of Applied Sciences
Enkelejda Miho, Universities of Applied Sciences and Arts Northwestern Switzerland
Andreas Christ, Offenburg University of Applied Sciences
Klaus Dorer, Offenburg University of Applied Sciences
Henrik te Heesen, Trier University of Applied Sciences
Rodolfo Orjuela, ENSISA-IRIMAS
Germain Forestier, ENSISA-IRIMAS
Alain Dieterlen, IUT-IRIMAS
Djaffar Ould Abdeslam, IUT-IRIMAS
Patrice Wira, IUT-IRIMAS

Organizing committee

Marie Wolkers, AlsaceTech
Maxime Devanne, ENSISA-IRIMAS
Thomas Josso-Laurain, ENSISA-IRIMAS
Jonathan Weber, ENSISA-IRIMAS
Jean-Philippe Lauffenburger, ENSISA-IRIMAS

Foreword

In a world where artificial intelligence (AI) is rapidly expanding across diverse sectors, it specifically plays an increasingly central role in understanding or driving complex dynamics and autonomous processes. This year, the Upper Rhine Artificial Intelligence Symposium delves into the theme *"AI for Time Series, Robotics, and Beyond,"* focusing on breakthroughs and applications that are reshaping industries, healthcare, energy, mobility, and many other essential fields. AI for time series analysis has emerged as a strategic research area, revealing hidden patterns, predicting behaviors, and detecting anomalies in datasets that evolve over time. This analytical capability is essential in fields such as finance, energy networks, predictive maintenance, and public health. Meanwhile, robotics innovations are progressively integrating into our daily lives and professional environments, demanding robust AI models for autonomy, human interaction, and real-time decision-making.

This symposium aims to provide a platform for sharing and collaboration among researchers, engineers, and decision-makers. Together, we will explore essential questions about how AI can process and interpret time-based data, the role of AI in developing adaptive and autonomous robots, and the future perspectives of AI applications beyond these domains. More broadly, this gathering is also an opportunity to reflect on the ethical implications, technical challenges, and international collaborations required to foster responsible and sustainable AI.

After 4 editions of the symposium held in Germany (respectively in Offenburg, Karlsruhe, Kaiserslautern, Furtwangen), UR-AI will celebrate its 5th anniversary in France. The milestone conference organized by the tri-national alliance TriRhenaTech, the alliance of universities of applied sciences in the Upper Rhine region, will stay two days in Mulhouse. For the first time, it is preceded by a one-day Autumn School dedicated to students willing to discover how AI can be used for time series from robotic platforms.

We invite you to dive into these cutting-edge topics, exchange innovative ideas, and build the future of artificial intelligence together in the Upper Rhine region and beyond.

Generated with ChatGPT

Hip hip UR-AI !!

Prof. Dr WEBER Jonathan
ENSISA-IRIMAS
Computer Department
Université de Haute Alsace

Prof. Dr LAUFFENBURGER Jean-Philippe
ENSISA-IRIMAS
Automatic, Signal, Image Department
Université de Haute Alsace

Table of contents

v

Keynote 1 : Towards Autonomous Vehicles : how AI is driving the mobility revolution?

Benazouz BRADAÏ

Reasearch & Innovation Director - Master Expert Autonomous Driving
VALEO

Fig. 1. Dr Benazouz Bradaï (VALEO)

Society is expecting cleaner, safer and smarter mobility. Automotive players are transforming in order to tackle those challenges. At Valeo, we make this happen with affordable, electric and smart mobility solutions.

Today, most of the new cars include ADAS "Advanced Driver Assistance Systems" features, some of them even include an Automated Driving of a Level 3 (L3) automation according to SAE classification. Valeo was a strong partner and enabler of those world first L3 vehicles. Major advancements in Automated Driving capabilities are still to come where Artificial Intelligence is a key enabler.

In this presentation we discuss the ADAS acceleration and the integration of cutting-edge AI technologies with Valeo automotive grade sensors to propel the evolution of autonomous vehicles from parking automation to advanced driving assistance and full autonomy. We will deep dive in few challenges on this evolution:

- How machine learning based perception is turning sensors inputs into useful understanding of the environment and inside the vehicle;
- How imitation and reinforcement learning are used for an efficient decision making including the prediction of other road participants like vulnerable road users;
- The end to end driving capabilities based on AI.

Finally, we will show the results of all that on Valeo Drive4U driving on open roads with only Valeo serial production sensors.

AI for Time Series

Knowledge Distillation for Robotics Time Series Classification

Javidan Abdullayev[1], Maxime Devanne[1], Jonathan Weber[1], and Germain Forestier[1]

IRIMAS, Université de Haute-Alsace
`javidan.abdullayevauha.fr`

1 Abstract

Recently, deep learning models have shown great success in a variety of fields, especially computer vision, speech recognition and natural language processing. The success of deep learning models motivated researchers to apply them to time series analysis, especially in Time Series Classification (TSC). A trend which we witness in deep learning field, state-of-the-art deep learning models become more complex over time. It is often impractical to deploy very complex deep learning models to embedded systems (edge devices, mobile phones), robots or a production enviorenment due to resource constraints. In deep learning context, knowledge distillation is a model compression technique which is used to transfer knowledge from a heavy model (deep) to a lightweight model. As a result, the lightweight model will require less resources in terms of memory and computation but will deliver competitive performance compared to the heavy model. The purpose of this paper is to introduce and explore the concept of Knowledge Distillation (KD) for time series classification with specific focus on robotics time series classification using state-of-the-art Inception architecture. In light of the fact that deep learning models are employed in the classification of time series, we believe using knowledge distillation is a viable research direction for the future.

2 Introduction

In recent years deep learning revolutionized the field of machine learning. Deep learning models have shown excellent performance in a wide range of applications, particularly computer vision [1], document retrieval [2] and speech recognition [3]. The success of deep learning architectures motivated researchers to examine them for time series analysis, in particular for the task of Time Series Classification (TSC). However, as proposed deep architectures for TSC have become increasingly complex, it often becomes impossible or at least impractical to deploy such cumbersome deep models to robots with limited resources (memory constraints, computational power, etc.).

Hence, the development of shallower models maintaining good performances is required. In this context, one of the model compression techniques is called knowledge distillation which allows for knowledge to be transferred from a larger deep model or ensemble of models (the *teacher*) to a smaller model with fewer parameters (the *student*). As a consequence, the resulting smaller model will require less computing power and less memory consumption while performing competitively with the larger model. The purpose of this paper is to introduce and explore the concept of Knowledge Distillation (KD) with specific focus on robotics time series classification. We analyze the effects of KD with the state-of-the-art deep learning model for TSC, the Inception architecture, by assessing its impact on shallower models by reducing the number of layers.

3

This idea of Knowledge Distillation was first proposed by Bucilia et al. 2006 [4]. Later, Hinton et al. 2015 [5], further developed this concept and proposed to train a *student* model using the output of the softmax function so that softmax outputs of *student* and *teacher* networks are close to each other. This concept has been mainly evaluated in Computer Vision for image classification.

In the domain of time series analysis, Deep Learning has also been shown to be very effective for TSC. Several Convolutional Neural Network (CNN) approaches have been proposed and adapted for TSC, such as multi-scale CNN [6], Fully Convolutional Network (FCN) [7] or Residual Network (ResNet) [7]. Later, in a review paper by Ismail Fawaz et al.[8], these approaches were compared with other deep learning approaches. A more recent approach adapting the Inception architecture for TSC, namely Inception-Time [9], demonstrated that considering various sizes of convolutional filters results in better classification accuracies.

3   Proposed Approach

The core idea of Knowledge Distillation implies two neural networks, a *teacher* (generally a deep model) and a *student* (generally a shallow model), as illustrated in Figure 1. The goal is to train a *student* model on a time series dataset $\mathbf{D}$ by leveraging the knowledge acquired by a pre-trained *teacher* model. The knowledge generally refers to the last layer's output of the *teacher* model. To consider Knowledge Distillation, during training, the *student* model is optimized to mimic *teacher* final predictions through a distillation loss (the Kullback-Leibler divergence) measuring the similarity between both models probability distributions. The knowledge obtained from a teacher model during the training of the student model is recorded in the distillation loss. Moreover, the *student* model is also optimized to minimize the classification error through a student loss (cross-entropy). Thus, the final Knowledge Distillation loss $\mathcal{L}_{KD}$ is defined as:

$$\mathcal{L}_{KD} = \lambda \times \mathcal{L}_{CE}(\mathbf{Y}, \hat{\mathbf{Y}}_S) + (1 - \lambda) \times \tau^2 \times \mathcal{L}_{KL}(\hat{\mathbf{Y}}_T^\tau, \hat{\mathbf{Y}}_S^\tau), \tag{1}$$

where λ controls the weight of both distillation loss $\mathcal{L}_{KL}$ and student loss $\mathcal{L}_{CE}$. The student loss $\mathcal{L}_{CE}$ corresponds to the classification loss defined as the cross-entropy between student predictions $\hat{\mathbf{Y}}_S$ and true labels $\mathbf{Y}$.

For the backbone architecture in our Knowledge Distillation framework, we choose the Inception architecture [9] as the pre-trained *teacher*. As shown in Figure 1, the *teacher* Inception network contains two residual blocks, each including three Inception modules.

For the *student* model, we follow the same Inception architecture but with fewer of Inception modules (layers). Hence, we build five different *student* models, denoted as Model_#M, by varying the number of Inception modules from 1 to 5. Detailed information about configuration of student models is given in table 1.

Model name	Teacher	Students				
	Model_6M	Model_5M	Model_4M	Model_3M	Model_2M	Model_1M
# Inception modules	6	5	4	3	2	1
Total # parameters	422 627	325 347	244 963	164 579	83 555	3 171

Table 1. Configurations of our *teacher* Inception and *student* Inception architectures
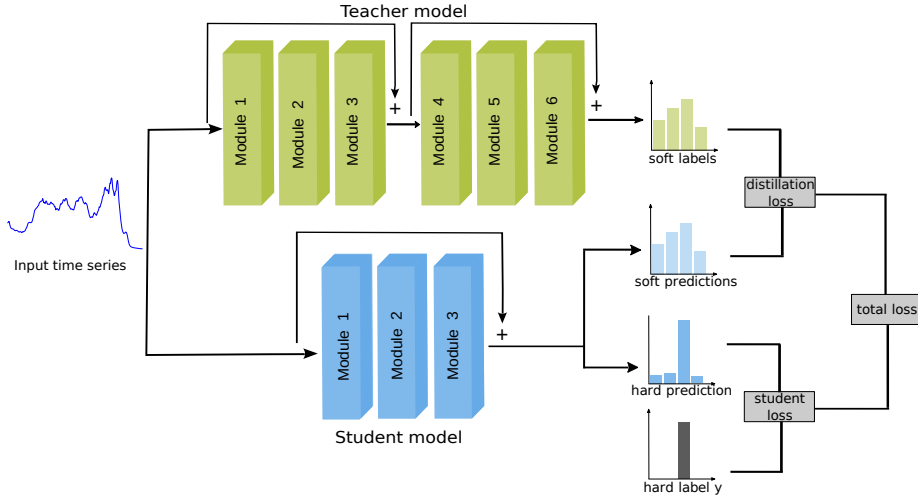
Fig. 1. Overview of our knowledge distillation architecture for time series classification

4 Experimental Evaluation

4.1 Experimental Setup

A large repository of time series datasets known as UCR Archive 2018[10] is used to evaluate student models. The repository contains 128 datasets but for a fair comparison, datasets containing unequal lengths or missing values are discarded. Consequently, we consider 112 univariate time series in our experiments.

The process of knowledge distillation start by training teacher model five times for each dataset. Among the five runs, the model with the lowest training loss is selected for knowledge distillation in order to achieve reproducible results. Then the selected teacher model is used to guide training process for the student model as described in Figure 1. In parallel, we also train the same student model, which we denote as studentAlone model, without taking into account knowledge distillation. In our experiments, aim is to assess impact of Knowledge Distillation on performance of shallower *student* models for the case of TSC. For that reason we compare the performances of *student* models benefiting from Knowledge Distillation against the *studentAlone* models trained only without the distillation loss. We train each student and studentAlone moels five times to reduce dependence on random initialization of Inception models. In order to assess each model's performance, we average the accuracy of five runs per dataset

4.2 Experimental Results

UCR Archive 2018 We first consider 112 time series datasets from the UCR Archive [10]. For each time series dataset, we compare the classification performances of both *student_#M* and *studentAlone_#M* models with various number of Inception modules. We then count the number of wins (*student* accuracy greater than *studentAlone* accuracy), losses (*student* accuracy lower than *studentAlone* accuracy) and ties (equal accuracies) for each *student_#M* and *studentAlone_#M* models. Comparative results are reported in Table 2. Furthermore, Figure 2 depicts number of wins for each model configuration. We can observe that Knowledge Distillation is particularly interesting in the case of intermediately complex models (*Model_4M*), where the *student_4M* model obtains higher accuracies than the *studentAlone_4M* on 57 time series datasets.

Model_1M			Model_2M			Model_3M			Model_4M			Model_5M		
Win	**Tie**	**Loss**	**Win**	**Tie**	**Loss**	**Win**	**Tie**	**Loss**	**Win**	**Tie**	**Loss**	**Win**	**Tie**	**Loss**
19	6	87	38	12	62	42	19	51	57	13	42	46	18	48

Table 2. Win/Tie/Loss comparison of *student_#M* and *studentAlone_#M* models with various numbers of layers. These results are based on *student* performance.
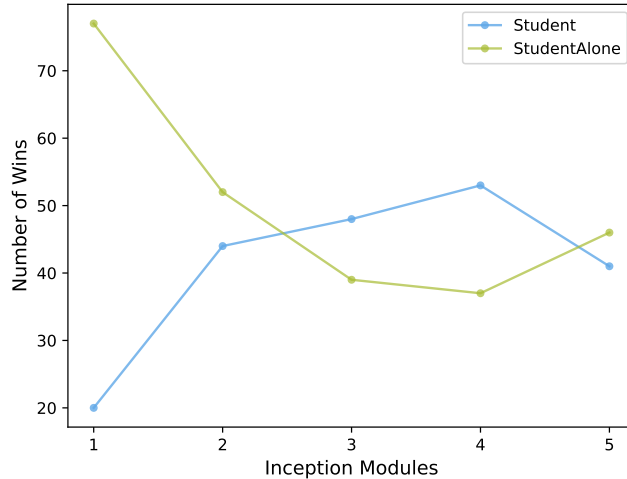

Fig. 2. Number of wins considering classification accuracy according to various numbers of layers of student models.

Based on all the results, we can conclude that in case of complex student model (student_5M) knowledge distillation does not help to improve the performance. This can be explained by the fact that as student models have similar complexity respect to a teacher model they will be able to capture discriminative patterns from the data without considering knowledge distillation thus demonstrate competitive performance with the teacher model. In case of intermediate complexity student model (student_4M), knowledge distillation helps to improve the performance significantly. The reason for this case is that since the student models have enough complexity, they can leverage and capture knowledge from a more complex teacher model. For relatively less complex student models knowledge distillation degrades performance of the student models (student_3M, student_2M, student_1M). This is due to the fact that less complex student models do not have enough capacity to leverage knowledge from a more complex student model.

Robotics Time Series Knowledge distillation is particularly interesting in the field of robotics in order to embed large deep neural networks in resource-constraint robotics systems. Hence, we focus especially on robotics time series by analyzing the results obtained for two related datasets, included in the UCR Archive, *SonyAIBORobotSurface1* and *SonyAIBORobotSurface2*. These datasets include use cases for surface detection using accelerometer data. Time series examples from SonyAIBORobotSurface1 dataset are illustrated in Figure 3. The robot has 3 accelerometers: roll, pitch and yaw. The examples from the Figure 3, represent accelerometer values that recorded in X axis. The task is to detect whether the surface being walked on is carpet or cement. The SonyAIBORobotSurface2 dataset contains same kind of examples as SonyAIBORobotSurface1 dataset

but the task in that case is to detect whether the surface being walked on is field or cement.
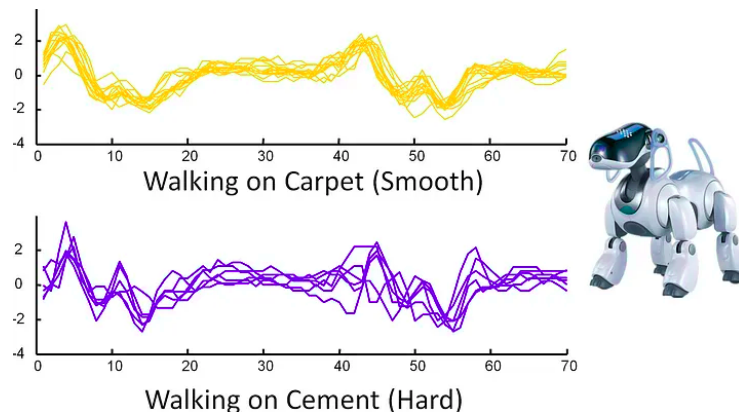

Fig. 3. Example time series from SonyAIBORobotSurface1 dataset.

Figure 4 and 5 demonstrate performance of all designed *student_#M*, *studentAlone_#M* and *teacher* models with different number of Inception modules for SonyAIBORobot-Surface1 and SonyAIBORobotSurface2 datasets respectively. From the figure 4, we can notice that all distilled student models except student_5M model demonstrate better performance than the studentAlone models. The results that we discussed in previous section is valid for the intermediate and complex student networks but not for smaller student architectures. In case of sonyAIBORobotSurface1 dataset we can observe that smaller student models perform better than studentAlone models. The distilled student model even with only one inception module, student_1M, demonstrate better performance than the studentAlone_1M model. The accuracy of student_1M, studentAlone_1M and teacher models are 87.4%, 85.4% and 86.9% From the results, it is obvious that knowledge distillation improves the performance of the studentAlone_1M model by 2%. To find an explanation for this behaviour we visited the dataset source [1], as written there the dataset has only 20 training examples. It means that the dataset is too small thus even smaller student models can capture discriminative features from the data as well as leverage knowledge from the teacher model.

We also analyze performance of knowledge distillation on another robotics dataset SonyAIBORobotSurface2. From the figure 5, we can observe similar patterns that we observed for SonyAIBORobotSurface1 dataset. Smaller and intermediate distilled student models demonstrate slightly better performance over the corresponding studentAlone models. Based on the source [2], the dataset contains only 27 training examples which shows the reason why smaller distilled student models performs better than studentAlone models. For the complex distilled student_5M model knowledge distillation does not improve the performance. From the Figure 5 we can notice an unexpected pattern in which performance of studentAlone models do not improve as we increase number of inception modules after studentAlone_3M model. We believe that this pattern is due to overfitting issue with this dataset as it contains only 27 training samples.

---

[1] https://www.timeseriesclassification.com/description.php?Dataset=SonyAIBORobotSurface1
[2] https://www.timeseriesclassification.com/description.php?Dataset=SonyAIBORobotSurface2

Fig. 4. Performance of *student_#M*, *sudentAlone_#M* and *teacher* models with different number of Inception modules on SonyAIBORobotSurface1 dataset.

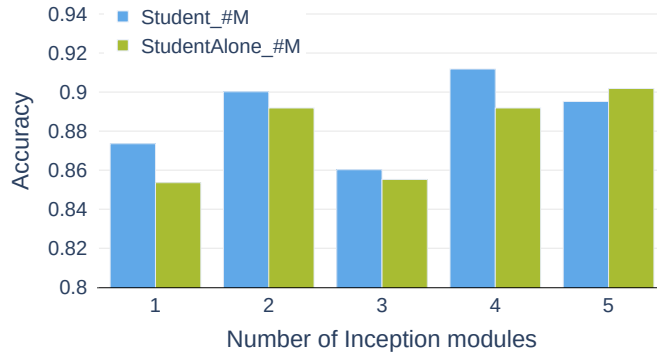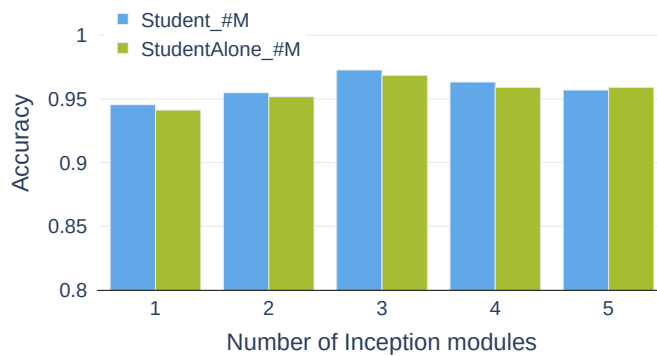

Fig. 5. Performance of *student_#M*, *sudentAlone_#M* and *teacher* models with different number of Inception modules on SonyAIBORobotSurface2 dataset.

5 Conclusion

In this paper, we analyzed the success of knowledge distillation on Inception network in the case of TSC, particularly for two robotics datasets. We assessed the impact of knowledge distillation by reducing the complexity of student models while leveraging teacher knowledge. In particular, we evaluated the impact of knowledge distillation by reducing the number of inception modules in student architectures. The experimental results conducted on the UCR Archive 2018 datasets which is the largest repository of time series datasets available and it suggests that knowledge distillation boost the performance when applied on intermediate complexity student models. This results is inline with the one that observed in [11] for image classification problem. The *student_4M* Inception model has 2 more wins over the *teacher* model. Considering that the number of parameters for the *student_4M* model is 42% less than the *teacher* Inception model then the *student_4M* model can be regarded as a very promising architecture to replace the *teacher* model.

The results that we observed on robotics dataset was a bit different. For robotics datasets knowledge distillation improved performance for even smaller student models. We strongly believe that this is due to less number of training examples in those datasets. According to the experiments on robotics datasets, even student_1M model demonstrate slightly better performance than the more complex teacher model. It is worth to note that the number of student_1M model parameters is equal to about 1% of the teacher model parameters. Based on these results, we can say that knowledge distillation is effective also for robotic datasets since it allows significant reduction in model size while maintaining good performance. Contrary to image classification problems in which knowledge distillation evaluated on few datasets, in our study we considered 112 time series datasets, demonstrating that finding an appropriate architecture for all type of datasets is not a straightforward task.

References

1. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems **25** (2012)
2. Li, H., Lu, Z.: Deep learning for information retrieval. In: Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval. (2016) 1203–1206
3. Noda, K., Yamaguchi, Y., Nakadai, K., Okuno, H.G., Ogata, T.: Audio-visual speech recognition using deep learning. Applied intelligence **42** (2015) 722–737
4. Ba, J., Caruana, R.: Do deep nets really need to be deep? Advances in neural information processing systems **27** (2014)
5. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531 (2015)
6. Cui, Z., Chen, W., Chen, Y.: Multi-scale convolutional neural networks for time series classification. arXiv preprint arXiv:1603.06995 (2016)
7. Wang, Z., Yan, W., Oates, T.: Time series classification from scratch with deep neural networks: A strong baseline. In: 2017 International joint conference on neural networks (IJCNN), IEEE (2017) 1578–1585
8. Ismail Fawaz, H., Forestier, G., Weber, J., Idoumghar, L., Muller, P.A.: Deep learning for time series classification: a review. Data mining and knowledge discovery **33**(4) (2019) 917–963
9. Ismail Fawaz, H., Lucas, B., Forestier, G., Pelletier, C., Schmidt, D.F., Weber, J., Webb, G.I., Idoumghar, L., Muller, P.A., Petitjean, F.: Inceptiontime: Finding alexnet for time series classification. Data Mining and Knowledge Discovery **34**(6) (2020) 1936–1962

10. Dau, H.A., Bagnall, A., Kamgar, K., Yeh, C.C.M., Zhu, Y., Gharghabi, S., Ratanamahatana, C.A., Keogh, E.: The ucr time series archive. IEEE/CAA Journal of Automatica Sinica **6**(6) (2019) 1293–1305
11. Mirzadeh, S.I., Farajtabar, M., Li, A., Levine, N., Matsukawa, A., Ghasemzadeh, H.: Improved knowledge distillation via teacher assistant. In: Proceedings of the AAAI conference on artificial intelligence. Volume 34. (2020) 5191–5198

Combining Variational Autoencoder and Recurrent Neural Networks for Generic Human Motion Prediction

Jonas Hansert

Karlsruhe University of Applied Sciences
`jonas.hansert@h-ka.de`

Abstract. Real-time motion prediction in a three dimensional environment is required for many application from autonomous cars to human robot collaboration to free-fall sorting machines. The most widely distributed sensors for the detection of three-dimensional environments like time of flight cameras, lidar sensors, stereo cameras or radar devices delivers point clouds or other formats that can easily converted to point clouds. The high dimensionality of point clouds and even voxel grids is a major challenge for real-time motion prediction. Most approaches use a skeleton tracking algorithm for dimensionality reduction, which itself is very error-prone. We investigated an approach consisting of a combination of two separately trained neural networks. We used a variational autoencoder for dimension reduction combined with a long short-term memory or a gated recurrent units network for time series prediction in latent space. We were able to show that it is possible to make reliable motion predictions up to one second into the future, depending on the motion.

Keywords: Human Motion Prediction, Recurrent Neural Network, Variational Autoencoder, Machine Learning, Time Series Prediction, 3d Computer Vision

1 Introduction

Predicting human motion is crucial for human-machine interaction. Humans are adept at predicting human motion, allowing us to collaborate with others. Navigate through crowds without causing accidents, solving tasks together, or to defend ourselves against attacking players in sports games are some examples. But even simple tasks such as shaking hands require a reliable prediction of human movement.

For intelligent machines, predicting human motion is important for avoiding collisions and enabling successful human-robot collaboration. It has already been applied to autonomous driving, intelligent robots and human-robot collaboration. Most 3D human motion prediction methods are based on skeletons [1–4]. Skeletons have the advantage that their dimensions are small, for example 25 joint positions in the widely used Microsoft Kinect V2 skeleton tracker [5]. However, estimating the position of skeletal joints is itself a challenging task with a non-negligible error. This error becomes much larger when there are objects close to the person or when some parts are in shadow of other objects.

In our approach, we work with voxel grids and reduce their dimension with a Variational Autoencoder (VAE). In the low-dimensional latent space, we use time series prediction models, which have been used with good results in many other applications [6–8]. This has several advantages over skeleton tracking. We have information about the environment that can influence the human's motion, and we can also predict changes in that environment.

2 Hardware and software setup

For data collection, we used an assembly station with the task of building a Fischertechnik model. The step-by-step instructions were projected on the worktop and the participants could move a wooden cube to see the next step. A detailed description of the setup and the task can be found in [9]. The data is captured using a RealSense D435 depth camera mounted centrally at a height of 2.5 meters at the rear end of the worktop. In figure 1 the assembly workplace used in this work is shown.

Fig. 1: The assembly workplace used for data collection.

The complete workflow of our 3d human motion prediction algorithm is illustrated in figure 2.

The depth camera (A) is mounted centrally at a height of 2.5 meter at the rear end of the worktop, so that the worker is imaged from the front-up.

In the next step we use the GPU-Voxels library [10] (B) to convert the point cloud to a voxel grid of size 64x64x64 voxels with the origin at the left rear corner of the worktop and with the worktop as a ground plane. In a calibration phase, we record the empty workspace and mark all the detected voxels as background and remove them from the voxel grid. To speed up the following calculations we unite a cube of 8 voxels to a single voxel if a minimum of 4 voxels are occupied in this cube. In this representation each voxel has a length of 4cm and we observe a quadratic space of 1,28 meter length in each axis. Our tests showed, that this is large enough for our use case.

Next, the voxel grid is used as input for the 3d convolutional VAE (C) where the dimension is reduced to a vector of size 32. The architecture of our VAE and the reason for the latent vector size are discussed in section 4.
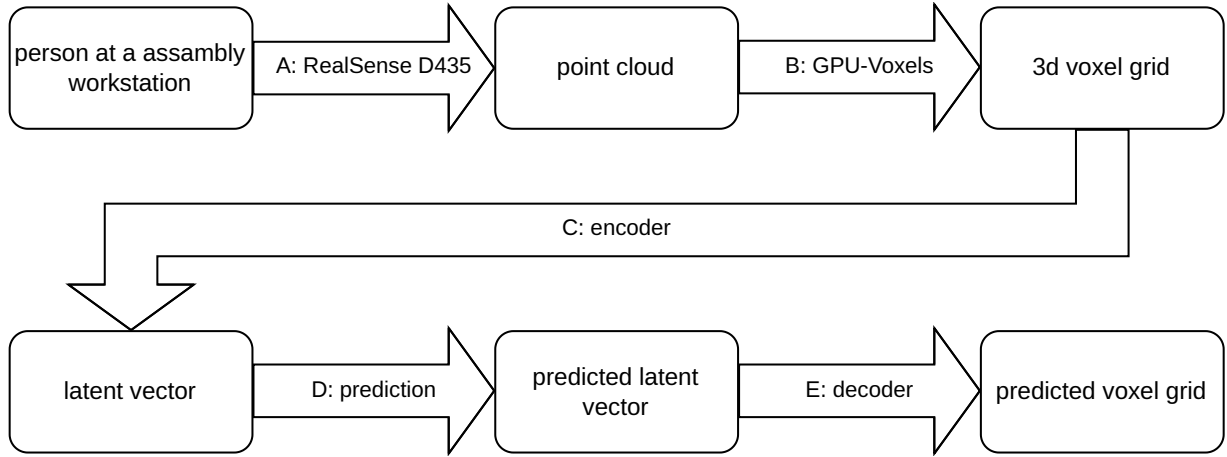
Fig. 2: The pipeline of our system for motion prediction on assembly workstations

We use this latent vector to predict the latent vector of the next time step. We use different models, like Multi-Layer Perceptron (MLP), Recurrent Neural Network (RNN) and Convolutional Neural Network (CNN), in this step (D). We train them and evaluate their accuracy. They are described in section 5.

To get the predicted voxel grid, we decode the predicted latent vector with the decoder of our VAE (E). If we want to predict more than one step, we have to calculate this for every step.

For data collection, training and evaluation, we used a workstation with a 16-core Intel I7 CPU, 64GB RAM and a NVIDIA GeForce RTX2080 Ti GPU.

3 Datasets

Because motion prediction does not require any labels in the dataset every dataset containing time series of point clouds could be used. Nevertheless the most of the widely used data sets for 3d human motion prediction only contain skeleton points or angles but not the pointcloud data or depth data we need for our algorithm. In this section we will introduce some datasets for motion prediction.

Human3.6M [13] is a large dataset with 3.6 million human poses and 17 scenarios. For motion prediction only the scenarios are from interest. The dataset contains 3d joint positions and angels and time-of-flight range data and some additional data not relevant for human motion prediction. This is the most widely used dataset for human motion prediction. Unfortunately, our account was not activated, so we did not get access to this data set.

The CMU Panoptic Dataset [14] contains, aside to a large range of multi-person scenarios, just 23 short recordings of single-person motions and each of them in different scenarios. This makes it hardly suitable to train or evaluate our neural networks.

3DPW [15] is another large dataset containing many scenes in the wild. It contains 60 video sequences and more than 51000 indoor and outdoor poses. Because it do not contain point clouds or depth data it is not suitable for our purpose.

G3D [16] is a large dataset of actions in computer games. It contains skeleton data as well as depth data. Each of the 24 action is recorded 30 times with 10 different actors. This dataset is suitable for our application but not used for the evaluation of any other

motion prediction tasks. This led to our decision, that there is no advantage in using the G3D dataset compared to our own dataset.

AMASS [17] dataset is a large dataset widely used for 3d motion prediction tasks. Unfortunately this dataset only contains skeleton and surface shape data. Caused by the missing point cloud data we can not use this dataset for our application.

The NTU RGB+D [18] Dataset contains 120 action recognition classes, most of them are single person daily actions. It contains 114,480 video samples of an average length of 1.9 seconds. If we want to train our network to predict a time step 1 second in future, this means, that we only have 0.9 seconds training data with no historical data we usually need for prediction with Long Short-Term Memory (LSTM) oder Gated Recurrent Unit (GRU) networks. This makes the NTU RGB+D Dataset not suitable for our application.

WBHM [19] dataset contains a total length of 7.68 hours. The long sequences would be perfect for human motion prediction. However, the dataset unfortunately contains neither point cloud nor depth data.

Because there is no dataset which fits well to our requirements, we collected two own datasets. In the first dataset we collected many different human poses in order to train the VAE with this data. Because balanced training data leads to better results [20], it is important that the number of frames for each human pose is uniform distributed. We recorded two persons for 15 minutes, grasping one part after the other, without building the model.

The second dataset contains the recordings of the building process of the Fischertechnik models. It combines the model building parts with short sequences of picking up new parts from the pick-by-light boxes. We recorded eight test subjects with durations between 16 and 48 minutes. In total we recorded 263 minutes.

4 Variational autoencoder architecture

The VAE is line symmetric. Due to better local dependency and fewer weights, we decided to use 3d convolution layers instead of fully connected layers in a vanilla VAE. We followed the architecture used by Brock et al. [11] but made some changes. We changed the input layer to 50x50x50, the size of our voxel grid and adapted the size of the other convolution layers to 16x16x16, 16x16x16 and 8x8x8 in the encoder part of our model. The decoder is designed symmetrical to the encoder. We added a dropout layer after each convolution layer for better generalisation changed the number of latent neurons to the desired dimension of the latent space.

To find the best dimension of the latent space, we trained the 3d convolutional VAE with latent dimensions of 16, 32 and 64. A latent dimension of 32 gave the best results with an accuracy of 0.974.

5 Time series prediction

We decided to train and evaluate the time series prediction models separated from the 3d convolutional VAE because the training is faster and because the the training data recorded for the prediction is less suited for training the VAE. Therefore, we give our hole training data set to the VAE and use the sequences of latent vectors for the training of our prediction models. We tested several different architectures and some of them with different hyper-parameters.

Multi-Layer Perceptron The architecture of MLP is poorly suited for time series prediction. Nevertheless, we did not want to ignore them because of their simple structure and the completeness of our comparison. We decided to use the past 10 time steps as input, which corresponds to 320 neurons in the input layer. We evaluated two different architectures. The first consists of one fully connected hidden layer with 500 neurons (Fig. 3 left) and the second of two fully connected hidden layers with 500 and 320 neurons . We tested both variants for the prediction of the next time step and the prediction of the next ten time steps (Fig. 3 right).
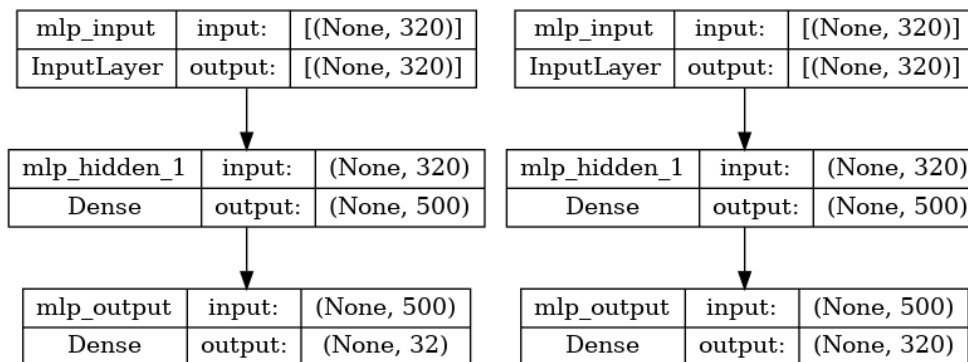
Fig. 3: MLP with 1 hidden layer for the prediction of one time step (left) and MLP with one hidden layer for the prediction of 10 time steps (right)

The mean squared error of the single layer network is with mse = 0.03 better than the two-layer network (mse = 0.04) for the prediction of one step and they are equal for the prediction of ten steps with mse = 0.07.

Convolutional Neural Network CNNs are well suited for time series prediction. With convolution over time, they are good at learning temporal relationships due to their structure. In our architecture a single 1d convolutional layer with 64 kernels of size 4. It is followed by a max pooling layer with kernel size 4 and after a flatten layer a fully connected layer (Fig. 4). For the one step prediction the output layer consists of 32 neurons and for the prediction of 10 time steps of 320 neurons.

Single Layer Long Short-Term Memory Network LSTM has been the first type of RNN to be used successfully in a wide range of applications, because they include mechanisms to avoid vanishing and exploding gradients. Our LSTM network consists of a single LSTM layer with the input and output layers of our MLP network (Fig. 5). For multi step prediction we designed the LSTM like the MLP with a larger output layer for 10 time steps.

The results of our single layer LSTM is better than the MLP but with mse = 0.025 in the case of one step prediction and mse = 0.065 for predicting the next ten steps it is still not very reliable.

Single Layer Gated Recurrent Unit Network GRUs are faster in the training process. and in the application often with better results, compared to LSTM. We decided to train a single layer GRU network and to compare it to the single layer LSTM. The architecture is similar to the architecture of the LSTM network.

cnn_input	input:	[(None, 32, 10)]
InputLayer	output:	[(None, 32, 10)]

cnn_hidden_1	input:	(None, 32, 10)
Conv1D	output:	(None, 29, 64)

cnn_max_pooling	input:	(None, 29, 64)
MaxPooling1D	output:	(None, 7, 64)

cnn_flatten	input:	(None, 7, 64)
Flatten	output:	(None, 448)

cnn_output	input:	(None, 448)
Dense	output:	(None, 32)

cnn_input	input:	[(None, 32, 10)]
InputLayer	output:	[(None, 32, 10)]

cnn_hidden_1	input:	(None, 32, 10)
Conv1D	output:	(None, 29, 64)

cnn_max_pooling	input:	(None, 29, 64)
MaxPooling1D	output:	(None, 7, 64)

cnn_flatten	input:	(None, 7, 64)
Flatten	output:	(None, 448)

cnn_output	input:	(None, 448)
Dense	output:	(None, 320)

Fig. 4: CNN for single step prediction (left) and CNN for multi step prediction (right)

lstm_input	input:	[(32, 32, 1)]
InputLayer	output:	[(32, 32, 1)]

lstm	input:	(32, 32, 1)
LSTM	output:	(32, 32, 32)

dense	input:	(32, 32, 32)
Dense	output:	(32, 32, 32)

lstm_input	input:	[(32, 32, 1)]
InputLayer	output:	[(32, 32, 1)]

lstm	input:	(32, 32, 1)
LSTM	output:	(32, 32, 32)

dense	input:	(32, 32, 32)
Dense	output:	(32, 32, 320)

Fig. 5: LSTM with 1 hidden layer (left) and LSTM for multi step prediction (right)

The single layer GRU network has mse = 0.024 for single step prediction and mse = 0.063 for multi step prediction. This is very similar to the single layer LSTM network. Because it is faster then the LSTM network in both, training and run time, we decided to use this network for our following tests.

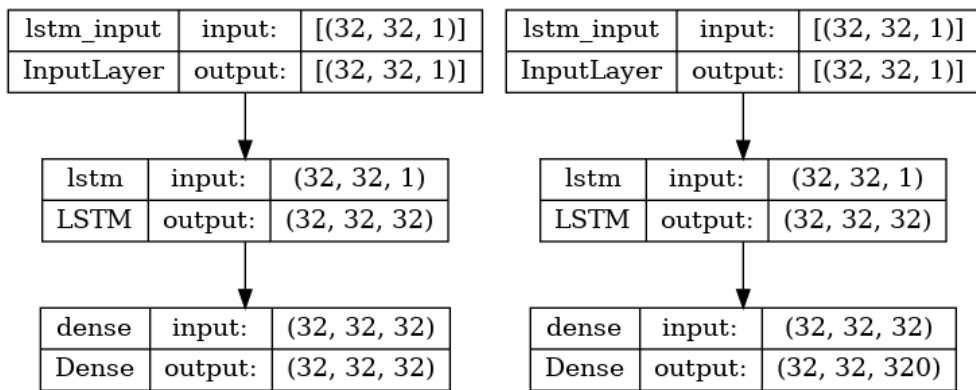**Multi-Layer Gated Recurrent Unit Network** Due to the best performance of the single layer GRU network, we tried to give more complexity to the model by adding a second GRU layer (Fig. 6). This architecture outperformed all other tested architectures with mse = 0.0235 for single step prediction and mse = 0.060 for 10-step prediction.

gru_input	input:	[(None, 32, 10)]
InputLayer	output:	[(None, 32, 10)]

gru_hidden_2	input:	(None, 32, 10)
GRU	output:	(None, 32, 64)

gru_output	input:	(None, 32, 64)
Dense	output:	(None, 32, 32)

gru_input	input:	[(None, 32, 10)]
InputLayer	output:	[(None, 32, 10)]

gru_hidden_2	input:	(None, 32, 10)
GRU	output:	(None, 32, 64)

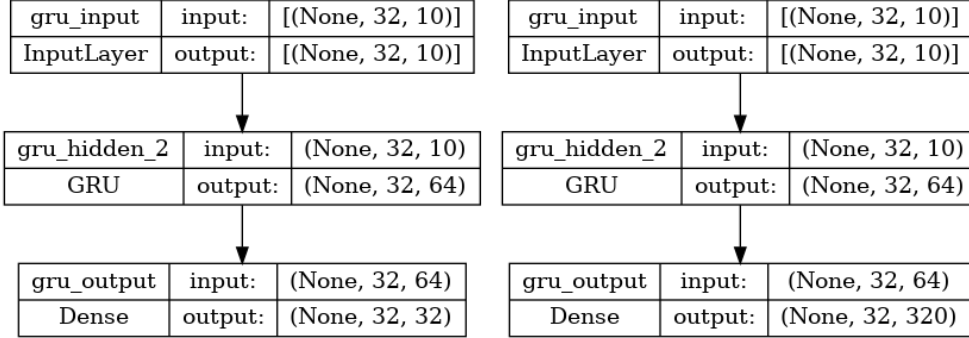| gru_output	input:	(None, 32, 64)
Dense	output:	(None, 32, 320)

Fig. 6: GRU network with two GRU layers.

6 Evaluation

In this chapter we take the best performing network architecture, the GRU network with two layers of 32 and 64 cells and evaluate it in different settings. We use the recording of one person for the evaluation. This net has a mse = 0.024 on the test data which is the same we had on the validation data (Fig. 7). For the prediction of 10 time steps in future it performs worse with mse = 0.075, but still better than all the other architectures we tested.

To have an idea how good the network architecture performs for the prediction of longer terms, we trained it for the prediction of one time step, 0.33, 0.66 and 1 second in future. The results are shown in Table 1.

predicted time	mean squared error
1 step (0.03 second)	0.024
10 step (0.33 second)	0.06
20 step (0.66 second)	0.09
30 step (1.00 second)	0.11

Table 1: Mean squared error for different prediction periods

Until now, we only looked at the predicted time series of latent vectors. Due to the properties of the VAE, this correlates to the accuracy of the predicted voxel grids. However, this needs to not be a linear relationship. For our hole algorithm, we have an
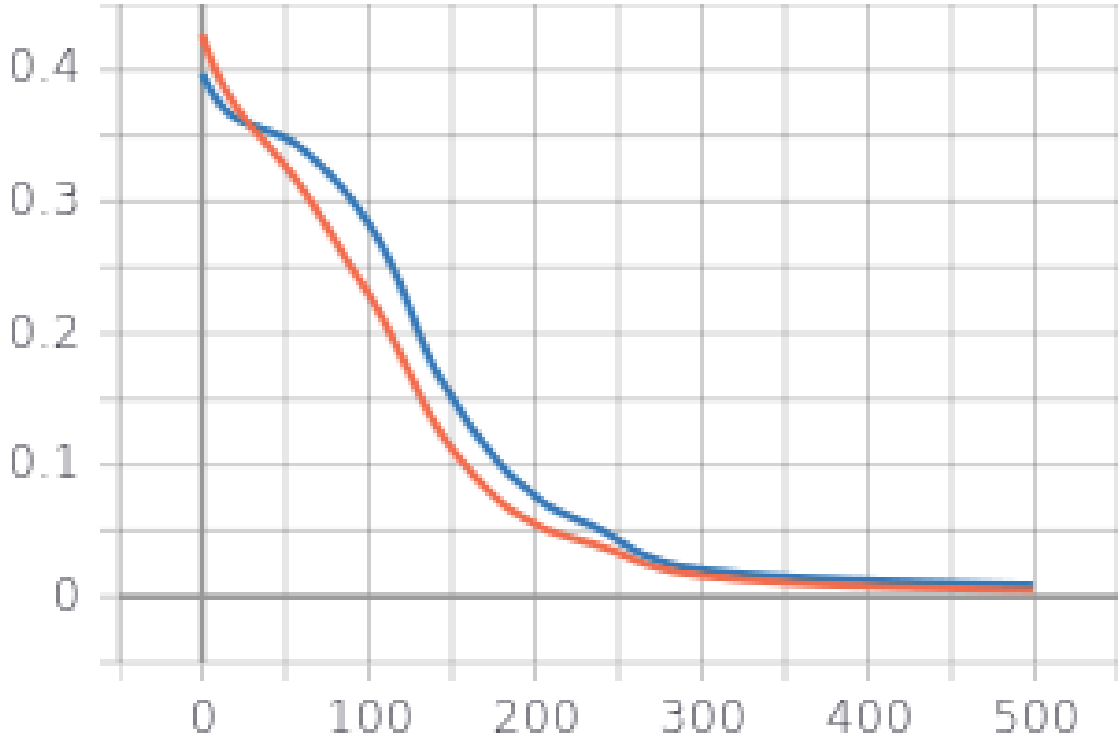
Fig. 7: The mean squared error over the training epochs on the training data (blue) and the validation date (red).

accuracy of 6.624 for the prediction of a single step 0.33 seconds in future from voxel map to voxel map.

7 Discussion

Bütepage et al. [2] [1] introduces a system using an autoencoder to predict motions based on the angels of 24 joints of the human skeleton. They compared three methods to input the timesteps to the autoencoder. In the first, the autoencoder is symmetrical, this means the decoder can be seen as the approximate inverse of the encoder. In this model the input of the autoencoder is only the actual time step and the output is one predicted timestep.

In the second autoencoder some 1d convolutional layers are added in front of the autoencoder input layer. Their kernels cover all the 24 joint angles and only fold over time. The third autoencoder uses a graph neural network to model the connections between the joints and to shrink the dimension of the input data before combining the different time steps. They got the best results with the third model. It has a mean angle error on the joint angles of 0.27 for prediction of 1 second and 0.15 for prediction of 0.08 seconds.

This result can not be compared to our result because we used voxel grids and Bütepage used joint angles. We see, that the accuracy does not decrease as fast as in our approach for long term prediction. However, Bütepage does not take into account the error in the skeleton tracking algorithm.

Martinez et al. [21] introduced a sequence to sequence model to predict 57 joint angles of the human skeleton. He used a single GRU layer with 1024 units and input and output

layers of 57 neurons. They introduced a zero velocity model as baseline. They reached a mean angle error of 1.15 for the prediction of 0.4 seconds and 0.36 for the prediction of 0.08 seconds. This result can not be compared to our result as already described above.

8 Conclusion

In summary, our algorithm is able to predict human motion 0.33 seconds into the future. For long term prediction, our algorithm, like all time series prediction algorithms, has some disadvantages compared to motion primitives.

The advantage of our algorithm is the omission of the skeleton tracking algorithm and the generation of a voxel grid that can be used by state-of-the-art robot motion planning algorithms without further conversion.

Due to the different data structures, it is not possible to compare the performance of our algorithm with state-of-the-art algorithms. Butepage et al. [2] and [1] use an autoencoder and Martinez et al. [21] use a GRU for short-term human motion prediction. As the dataset they use contains only skeleton points, it is not possible to compare our results with theirs. Generating voxel grids from the skeleton data and comparing the results on this basis could be a solution for further work.

Acknowledgment

References

1. Butepage, J., Black, M.J., Kragic, D., Kjellstrom, H.: Deep representation learning for human motion prediction and classification. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE 1591–1599

2. Butepage, J., Kjellstrom, H., Kragic, D.: Anticipating many futures: Online human motion prediction and generation for human-robot interaction. In: 2018 IEEE International Conference on Robotics and Automation (ICRA), IEEE 1–9

3. Kulić, D., Ott, C., Lee, D., Ishikawa, J., Nakamura, Y.: Incremental learning of full body motion primitives and their sequencing through human motion observation. **31**(3) 330–345 Publisher: SAGE Publications Ltd STM.

4. Luo, R., Hayne, R., Berenson, D.: Unsupervised early prediction of human reaching for human–robot collaboration in shared workspaces. **42**(3) 631–648

5. Moon, S., Park, Y., Ko, D.W., Suh, I.H.: Multiple kinect sensor fusion for human skeleton tracking using kalman filtering. International Journal of Advanced Robotic Systems **13**(2) (2016) 65

6. Cao, J., Li, Z., Li, J.: Financial time series forecasting model based on ceemdan and lstm. Physica A: Statistical mechanics and its applications **519** (2019) 127–139

7. Yamak, P.T., Yujian, L., Gadosey, P.K.: A comparison between arima, lstm, and gru for time series forecasting. In: Proceedings of the 2019 2nd international conference on algorithms, computing and artificial intelligence. (2019) 49–55

8. Livieris, I.E., Pintelas, E., Pintelas, P.: A cnn–lstm model for gold price time-series forecasting. Neural Computing and Applications **32**(23) (Dec 2020) 17351–17360

9. Swenja, S., Maximilian, P., Thomas, S.: Evolution of pick-by-light concepts for assembly workstations to improve the efficiency in industry 4.0. **204** 37–44

10. Hermann, A., Drews, F., Bauer, J., Klemm, S., Roennau, A., Dillmann, R.: Unified gpu voxel collision detection for mobile manipulation planning. 4154–4160

11. Brock, A., Lim, T., Ritchie, J.M., Weston, N.: Generative and discriminative voxel modeling with convolutional neural networks

12. Lin, T.Y., Maire, M., Belongie, S., Bourdev, L., Girshick, R., Hays, J., Perona, P., Ramanan, D., Zitnick, C.L., Dollár, P.: Microsoft coco: Common objects in context

13. : preprint.dvi

14. : Carnegie mellon university - cmu graphics lab - motion capture library (9/27/2023)

15. von Marcard, T., Henschel, R., Black, M., Rosenhahn, B., Pons-Moll, G.: Recovering accurate 3d human pose in the wild using imus and a moving camera. In: European Conference on Computer Vision (ECCV). (2018)

16. : Gaming datasets (7/28/2016)

17. Naureen Mahmood, Nima Ghorbani, Michael J. Black, Gerard Pons-Moll, Nikolaus F. Troje: Amass: Archive of motion capture as surface shapes

18. Liu, J., Shahroudy, A., Perez, M., Wang, G., Duan, L.Y., Kot, A.C.: Ntu rgb+d 120: A large-scale benchmark for 3d human activity understanding. IEEE Transactions on Pattern Analysis and Machine Intelligence **42**(10) (2020) 2684–2701

19. : Mandery2015apdf

20. Batista, G.E.A.P.A., Prati, R.C., Monard, M.C.: A study of the behavior of several methods for balancing machine learning training data. SIGKDD Explor. Newsl. **6**(1) (jun 2004) 20–29

21. Martinez, J., Black, M.J., Romero, J.: On human motion prediction using recurrent neural networks

Machine learning-based models for self-learning indoor heat warning systems in households

Oscar Villegas Mier[1], Willi Haag[1], Raghavakrishna Devineni[1], Guillerme Carraro Carella[1], Rainer Gasper[1], Jens Pfafferott[1], Michael Schmidt[1]

[1]Offenburg University of Applied Sciences
oscar.villegas@hs-offenburg.de

Abstract. With climate change and global rising temperatures heat health warning systems have become important in accurately predicting heat waves. However, most heat health warning systems rely on the ambient temperature forecast and do not take indoor building conditions into consideration. Moreover, a general heat warning system cannot accurately predict the heat stress conditions in individual buildings. To implement the prediction algorithms the study also proposes a Raspberry Pi based measurement system. Furthermore, to reduce the computational load on Raspberry Pi a Transfer learning technique is implemented from a pre trained Long Short-Term Memory (LSTM) neural network. The results show prediction accuracy of 97% with an RMSE of 0.218 for indoor temperature prediction.

Keywords: Self learning; Neural networks; Grey-box models; Blac-box models, heat warning, building thermal dynamycs

1   Introduction

Global temperatures are increasing due to global warming, leading to a higher occurrence of hot days with temperatures exceeding 30°C in Germany. Longer heat waves can have various impacts on the population leading to higher health risks as hot weather can be a major cause for stress in the human body [1]. A key challenge for maintaining health in heatwaves, relies in reducing heat strain via an adequate management system that ensures thermal comfort [2]. This is possible with different measures such as efficient ventilation and shading, using air conditioning systems, among others. A big challenge in the correct implementation of these measures is the assessment of stress in individual environments [3].

Timely preparation and prevention are the most effective measures against heat stress. Most warning systems in operation are restricted to the forecast of outdoor heat stress and they don´t account for specific conditions inside buildings [4]. As the heat impact of buildings indoors varies due to local conditions, each building must be rated separately. For this purpose, a system is needed that can be integrated into existing buildings to reliably measure and evaluate various building specific parameters to determine and predict building-specific conditions of heat stress, determine specific actions, and issue a heat warning.

To make this kind of system widely available, it should be compact, low power consuming and low cost. In recent years artificial intelligence (AI) capable, low cost, compact processing devices such as the Raspberry Pi have been more widely available in the market, but due to their size, their processing power is still constrained. This presents a challenge in terms of deploying and running complex algorithms to process the information and learn from the conditions of the environment where they are placed. The computational requirements of these algorithms may exceed the capabilities of the device. Furthermore, the limited memory capacity of these devices may pose constraints on the size and complexity of the models that can be deployed [5].

In this paper, we will discuss the development and implementation of self-learning algorithms developed with AI techniques to perform temperature predictions and deliver warnings within our low cost in-house developed heat warning system. The heat warning system was developed as part of the project heatGUIde, which is supported and funded by the Baden-Württemberg Stiftung. It was designed to monitor and predict extreme heat stress events in residential buildings. It consists of the integration of multiple sensors to measure thermal comfort parameters such as temperature, humidity, wind velocity, mean radiant temperature (from a black globe temperature sensor), light, and

CO2. The design is based on an Internet of Things (IoT) architecture, with a central gateway based on a Raspberry Pi computer as the main core for processing, and LoRa modules for data transfer.

2   System Design & Architecture

The heatGUIde heat warning system was built following methods based on the architecture of Internet of Things (IoT) systems for the data collection. The devices based on these architectures are often resource-constrained, small sized and can be battery-powered due to their low energy consumption [6]. This makes them a satisfactory solution for a low cost, multi sensor device. The sensors for the measuring system were selected based on cost, energy consumption, accuracy, and availability in the market.

The measuring system consists of a node and a gateway. The node measures the indoor comfort parameters and transfers this data to the gateway. The node consists of low-cost sensors hardwired to an ESP32 microcontroller enclosed in a 3D printed casing. The sensors measure Temperature (°C), and Relative Humidity (%rh), CO_2 Concentration (ppm), Wind Speed (m/sec), and Mean Radiant Temperature (°C).

The ESP32 reads out the measurement data from the sensors and sends the data wirelessly to the gateway. The gateway is the central processing unit of the measuring system. It consists of a Raspberry Pi, LCD screen and a LoRa Hat. LoRa radio communication is used to communicate between the node and the gateway. LoRa is a proprietary radio communication technique operating sub gigahertz radio frequency band capable of data transfer over longer distances and through obstacles like thick walls. The LoRa hat in the gateway makes the Raspberry Pi LoRa capable there by circumventing the use of standard LoRa gateways available on the market.

The gateway consists of necessary software for data collection, storage, analysis, visualization and the algorithms for data learning and prediction. The system architecture and the information flow through different software is shown in Figure 1.
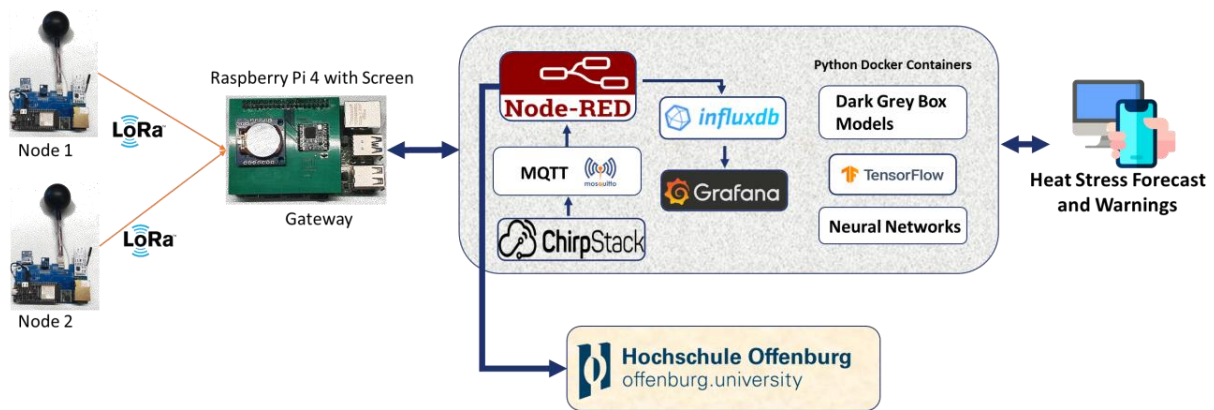

Figure 1. System architecture and data flow of the measurement system for the project heatGUIde

The data received by the gateway is decrypted by ChirpStack (LoRa WAN Server software), which is then sent to Node RED using mosquito MQTT broker. The Node RED is a graphical developing software, which directs the data from the MQTT broker to various destinations. Additionally, Node RED can also be programmed to display interactive dashboards on the gateway screen. Different models discussed in this study are deployed in different Python containers. The Node RED communicates the data with different Python containers to forecast heat stress indicators. The calculated metrics are saved in an influx Database locally and on a remote server for backup. The important results and forecasted heat warnings are displayed through interactive dashboards on the Gateway.

3 Methods

3.1 Data Acquisition

With the intention of making the AI models more generic and adaptable to different climatic conditions in Germany, the models should be fed with the local weather information. To achieve this, we considered the Test Reference Year (TRY) data with 14 parameters modelled from the weather data for the last 20 years. A Principal Component Analysis (PCA) was performed to identify the important parameters for forecasting.

The TRY data was incorporated into a real-world scenario at Project House Ulm. Project House Ulm is a single-family house, equipped with multiple sensors and data collecting systems for the purpose of studying real world conditions. For the initial analysis we created Long Short-Term Memory (LSTM) neural networks with data from Project House Ulm and the TRY data. A data model was built as follows:

Table 1. Offline: Past Data, Actual Data, without forecast as features.

Features (inputs)	Predictions (outputs)	Horizon	Historical input	Results
Indoor Temperature	Indoor Temperature	1+ days (Tested 3 days)	5 days past	R2: 0.975 RMSE: 0.59

The forecast results are shown in Figure 2 which displays the accuracy of predicted values compared to the original dataset.
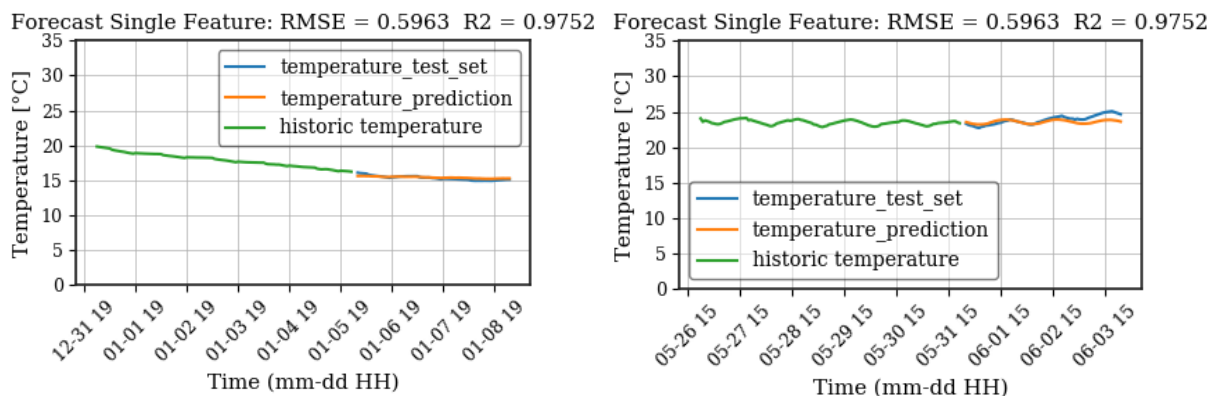

Figure 2. Validation of preliminary analysis neural network models results. The graph shows different test set validation scenarios of temperature predictions, presenting good adaptation from the data in fluctuating temperature scenarios.

The number of features selected can impact directly in the predictions using neural networks. By means of an analysis with PCA reductions we estimated the effect on the overall prediction error. The error is relatively small when using the full 12 features in the TRY dataset. However, when reduced by up to 7 dimensions, the error remains consistently low, allowing accurate prediction with less data.

Following the initial proof of concept and PCA analysis, we shifted our focus to acquiring and utilizing real data from residential building rooms. After completing the initial proof of concept and PCA analysis, we gathered the indoor data from the heatGUIde system with outdoor data sourced from the German Weather Service (DWD), along with weather forecasts from their MOSMIX model. These combined datasets were used to train several black and grey-box models.

The heatGUIde system is integral to our data acquisition process. The data collected by these nodes was pivotal for the development, training, validation, and continuous evaluation of our diverse model approaches. The system offers real-time indoor heat stress and comfort data for model predictions and evaluations. Storing and analyzing this data helps us better understand how different environmental factors collectively affect indoor heat stress conditions.

3.2 Modeling Approach

Indoor room temperature is a principal factor involved in the physical sensation of heat, and we base our heat warning calculations mainly on it. For instance, our focus relies on accurate prediction and understanding the changes in the room temperature where our heat warning system nodes are installed.

In this work, different self-learning models were explored. To study their differences, we divided the approaches into black-box and grey-box models.
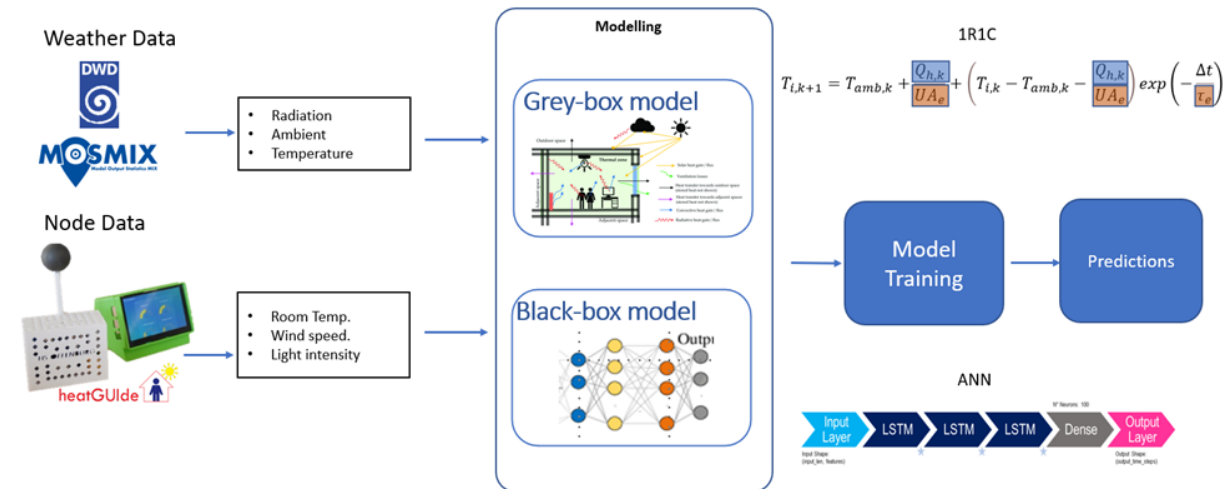

$$T_{i,k+1} = T_{amb,k} + \frac{Q_{h,k}}{UA_e} + \left(T_{i,k} - T_{amb,k} - \frac{Q_{h,k}}{UA_e} \right) exp\left(-\frac{\Delta t}{\tau_e} \right)$$

Figure 3. Data flow of the implementation of a self-learning algorithm for an indoor heat warning system

The first modeling approach explored the creation of black-box models. Black-box models are complex algorithms known for generating reliable predictions or outputs based on the self-adaptation of their input data, but with their decision-making processes being opaque [7].

Our selected black-box models were Long Short-Term Memory (LSTM) neural networks, which are a special kind of Recurrent Neural Networks (RNN), that were utilized due to their capacity to learn and remember patterns over extended sequences, crucial for time-series data inherent to temperature predictions.

In addition, we explored the use of Transfer Learning with LSTMs to reduce the processing requirements. Transfer learning is a machine learning technique that works based on using pre-trained models developed for a specific task and trained on extensive datasets and reusing the model as the starting point for a model on a second task, leveraging pre-learned patterns for enhanced learning efficiency and accuracy. It's especially useful in scenarios where the data is scarce, or computational resources are limited.

The second modeling approach uses grey-box models which are a type of predictive model that combines both theoretical and empirical approaches, offering a balance between the transparency of physical models and the predictive power of black-box models. They incorporate known physical processes with data-driven components, facilitating interpretation while maintaining predictive accuracy [8].

The selected grey-box model included in this study is a 2R2C model. An RC (Resistor-Capacitor) is a simplified representation of building thermal analysis to simulate and understand the thermal behavior and heat transfer processes within buildings. It uses electrical analogs of thermal resistances (R) and capacitances (C) to model the heat flow and thermal storage within building elements.

In comparison to a 1R1C model which is a very simple lumped parameter model, the 2R2C model introduces an additional layer of complexity and precision. While computationally more intensive, it provides detailed insights into thermal dynamics, improving the accuracy of the predictions.

4 Implementation

Each of the selected modeling approaches works with distinct methodologies which require specific tools for their correct development and implementation.

4.1 Platform and Tools

All the software frameworks on the Raspberry Pi utilize docker containerization. The model implementation, conducted using Python 3.10, is described in Figure 5, outlining the process of training and validation.
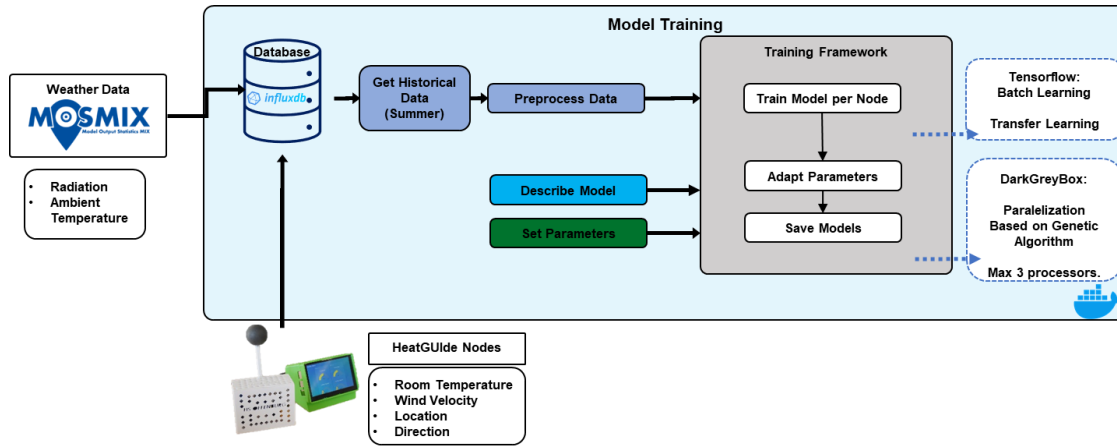

Figure 4. Model Training implementation

Both model implementations require the use of measured data. As mentioned in Section 3, the collection of real data is important for this step. The required data is first retrieved from the database, and then preprocessed with python Pandas to fill in gaps. To facilitate the training and prediction, and to match the heat warnings, the data was resampled to one-hour intervals. As some data is measured in smaller samples, data must be up sampled, and with matching sampling rates, the datasets are joined and prepared for the modelling.

4.2 Black-box models and transfer learning.

For the black-box models, python TensorFlow and Keras were employed due to their extensive libraries, community support, and scalability offering the flexibility and resources necessary for building and training complex neural networks. This combination facilitated a smooth development process, allowing for efficient design, training, and validation of the LSTM neural networks. The availability and support of a full TensorFlow installation, containerized within the Raspberry Pi environment, were significant factors influencing the choice of this framework.

The selected features were divided into two categories, and scaled separately using min-max normalization as follows: The first feature uses the indoor temperature which was time shifted to be given as the predicted value and kept in its current state as feature. Second the general conditions of the prediction, such as ambient temperature, radiation, wind speed, and time derived features, such as the time of the day, month, weekday, weekend, and season. The latter features were scaled separately to re-scale the outputs independently when performing predictions. The final structure of the Artificial Neural Network (ANN) model is shown in Table 2. The model was trained during 400 epochs, with an early stopping callback with persistence of 20 epochs, and with a batch size of 32, using the ADAM optimizer with a mean squared error as loss function.

Table 2. LSTM neural network architecture for black-box model.

Layer (type)	Output shape	param
Input Layer	(None, 1, 14)	Neurons: 14

LSTM	(None, 1, 100)	Neurons: 100, Return sequences: True, Function: Tanh
Dropout	(None, 1, 100)	Value: 0.9
LSTM	(None, 1, 50)	Neurons: 100, Return sequences: True, Function: Tanh
Dense	(None, 1)	Neurons: 1, Function Linear

Training neural networks on the Raspberry Pi presented significant challenges due to its limited processing power and memory capacity, therefore we explored two approaches: training with the full dataset and employing transfer learning.

Using the complete dataset provides the model with a broader understanding of the processes happening in the data, but it is computationally intensive for the Raspberry Pi. To mitigate this limitations as a second approach, models were initially trained on a more powerful system using a subset of the data, then fine-tuned on the Raspberry Pi using real-time data from the heatGUIde nodes. This approach ensured the models could efficiently learn and adapt to the unique characteristics of each household without overwhelming the Raspberry Pi's computational resources.

To implement transfer learning, we executed a simple strategy, which consists in pruning the original model from their output layer, substituting it for a new output layer, and restricting the training of the layers to the last one. This way the original layers remain so-called "Frozen" and only low computational requirement is needed to execute the adaptation of the output layer. The model was trained during 100 epochs, with batch size of 32, using the ADAM optimizer with a mean squared error as loss function.

4.3   Grey-box models.

The grey-box models were implemented using the DarkGreyBox [9] library for Python, which is specifically designed for the creation, training, and validation of grey-box models. It provides a data-driven approach within the classic Machine Learning (ML) framework for model performance evaluation. At the same time, it allows to setup and select the best performing from a series of competing models based on principles inspired by Genetic Algorithms (GA), addressing the main disadvantages of training grey-box models, that require initial condition values for the thermal parameters to be pre-calculated [10].

The primary challenge was finding a balance between model complexity and optimization efficiency for the 1R1C and 2R2C models. While a complex model could accurately describe a building's thermal dynamics, it would also be computationally demanding, making it impractical for real-time applications on low-power devices.

Careful experimentation and iteration led to models that were complex enough to accurately represent building dynamics but streamlined enough to allow for efficient optimization. This approach ensured the models were both accurate and practical for deployment on low-capacity systems like the Raspberry Pi.

The successful implementation of these methods implies that accurate heat stress prediction in households is achievable on low-cost, low-capacity systems. Each model is finely tuned for specific locations, ensuring precise and reliable predictions. These location-specific models are not only economically feasible but also efficient and effective in predicting indoor temperature, paving the way for widespread adoption and implementation in various households to safeguard residents against the health risks associated with heatwaves.

5   Results & Discussion

The proposed models were trained with data from the heatGUIde nodes which were tested and collected data in a real house in Offenburg Germany.

5.1   Black-box Model Analysis

The first approach encapsulates the training of neural networks utilizing the entire dataset within the Raspberry Pi environment. Validation results denote a notable precision in predictions. Graphical depictions demonstrate a robust

correlation between the predicted and actual indoor temperatures (refer to Figure 5) showing a prediction accuracy of 97% with an RMSE of 0.218.
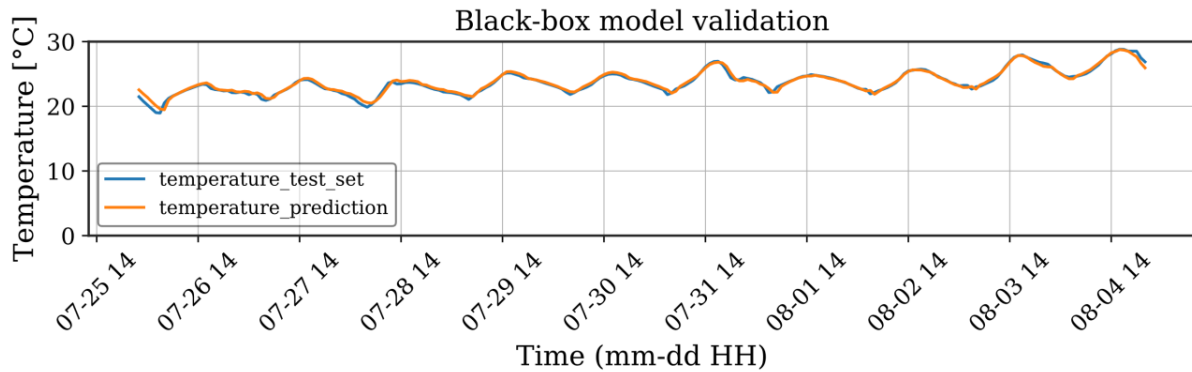

Figure 5. Test-set validation results for LSTM neural networks, trained with batch data approach.

Despite the efficacy in predictions, there's a consequential demand on the Raspberry Pi's computational resources. Table 3 describes the CPU utilization and memory allocation statistics during the training phase. These quantitative measures underscore the substantial resource and processing power requirements of this approach, posing potential feasibility challenges for sustained, real-time applications.

Table 3. Resource utilization for model training in raspberry Pi.

Method	Clock-Time (s)	CPU-Time (s)	Memory (GB)
Black-box (Full dataset)	169.63	231.91	1.78
Black-box (Transfer learning)	62.30	62.32	1.74
Grey-box	335.52	44.44	1.73

5.2   Transfer Learning Approach

Alternatively, models instantiated through transfer learning exhibited a rapid acclimatization to the unique dataset characteristics. Showing a prediction accuracy of 72% with an RMSE of 0.108 (refer to Figure 7).

This methodology manifested a substantial reduction in the computational resource's requirements, a pivotal consideration for the Raspberry Pi's limited hardware capabilities. The efficient training process, reduced resource requirements, and good results of this approach renders it a viable candidate for real-time predictive modeling in constrained environments.
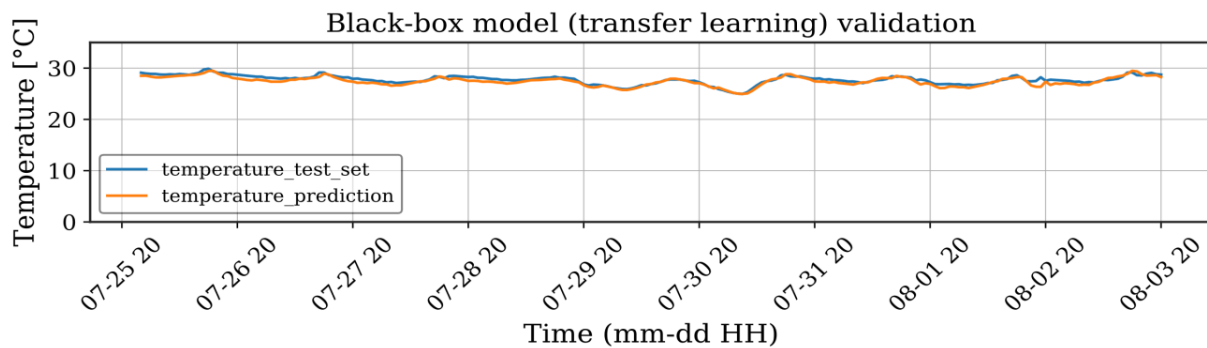
Figure 6. Test set validation for black-box model with transfer learning approach.

5.3 Grey-box Model Analysis (2R2C Model)

The 2R2C grey-box model's implementation on the Raspberry Pi underwent rigorous validation procedures. The resulting data shown in Figure 7, illustrates a coherent and reliable fit of the data within the model. Showing a prediction accuracy of 85% with an RMSE of 1.765. Despite the complexity of the 2R2C model, the Raspberry Pi proficiently managed its optimization and training phases, with the detailed resource requirements shown in Table 3. To accommodate better resources for parallel tasks required by the programs described in Section 2, we limited the parallelization to three out of the four available processors, ensuring uninterrupted program execution. This successful implementation shows the model's viability for deployment in environments with restricted computational resources.
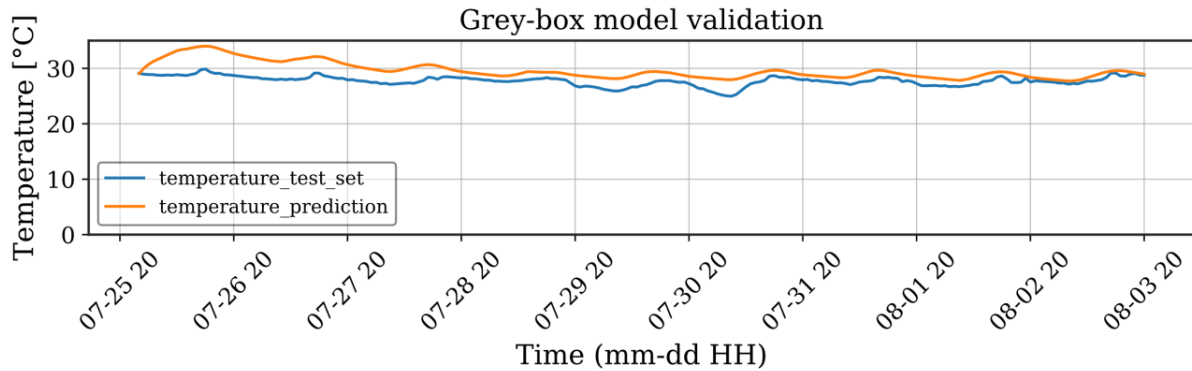

Figure 7. Test set validation for grey-box 2R2C model with adapted parameters.

5.4 Discussion

Black-box models offer an efficient, accurate, and straightforward means to deliver predictions. At the same time, they inherently suffer from a lack of interpretative transparency. This characteristic, that can be proven advantageous for quick modelling and implementation requiring little building thermal dynamics knowledge, hampers the intuitive understanding of their decision-making, complicating potential troubleshooting and replicability. Also, in the case of our research, further implementation of control strategies can become complex due to the missing binding of physical properties, to the input variables. Although this can be overcome with the use of hybrid models, their implementation becomes more complex.

Conversely, the grey-box approach exemplified by the 2R2C model brings together predictive accuracy with an unparalleled transparency, facilitating a more intuitive understanding of the intricate decision-making processes at play. This clarity not only streamlines troubleshooting but also simplifies the model's expansion and modification endeavors. The ensuing versatility and adaptability, coupled with dependable accuracy, make the grey-box models particularly appealing for diverse and dynamic real-world applications such as control related requirements.

Through comprehensive analysis and validation, our findings demonstrate that the grey-box models can provide a confluence of predictive accuracy, model transparency, and computational efficiency. This renders this approach particularly well suited for deployment in small computational devices, such as the heatGUIde system, providing in this sense pivotal insights for the future developmental trajectory and application spectrum of household AI-driven heat warning systems.

In comparison to prevailing heat warning solutions, our heatGUIde system introduces several distinguishing features that substantially elevate its value proposition for end-users seeking efficient household heat reduction solutions such as low-cost, and the inclusion of Self-Learning Predictive Models. These models give the system dynamic

adaptability by providing personalized predictions of the room conditions, which can aid better heat stress reduction measurements.

The use of the self-learning models is not only to predict the room conditions but can also be used use analyze and recommend specific actions users might take to preemptively mitigate heat stress, going beyond simple warning systems to act as an advisory companion for users in heat management.

Future work will focus on improving the learning capabilities of the system, to continually refine and enhance its predictive accuracy over time. By learning from new measured data, the system becomes progressively more adept at forecasting indoor temperatures and heat stress scenarios, ensuring that its utility improves the longer it is in use. Additionally, focus on user experience is previewed to enhance its adoption and foster seamless integration into households. Lastly as the system is designed with future integrations in mind. Its architecture and algorithms are being designed for compatibility with smart home systems. For instance, it could dynamically adjust smart blinds or ventilation systems in anticipation of changing indoor temperatures, providing not just a reactive but a preventative solution to indoor heat stress.

This advanced, yet accessible system thus stands as a significant contribution to the field, with the potential to drive further innovations and improvements in the realm of household heat management and control technologies. Our research encapsulates the next stride in indoor environmental management, offering a glimpse into the future of smart, safe, and comfortable living spaces.

6 Conclusion

Through a structured and phased data acquisition strategy, we ensured a systematic approach to understanding and analyzing indoor thermal dynamics. The initial phase provided a theoretical and practical understanding, while the usage of the data collected from the heatGUIde system, offered real-world, applicable insights, collectively laying a solid foundation for our subsequent methodological applications and algorithm developments. This diverse dataset was indispensable for the development and fine-tuning of our models, providing a rich and varied source of information to train, test, and validate our algorithms effectively and efficiently.

Considering the escalating global temperatures and the increasing prevalence of heatwaves, the successful implementation of these methods implies that accurate heat stress prediction in households is achievable on low-cost and low-capacity systems. Each model is finely tuned for specific locations, ensuring precise and reliable predictions. These location-specific models are not only economically feasible but also efficient and effective in predicting indoor temperature, paving the way for widespread adoption and implementation in various households to safeguard residents against the health risks associated with heatwaves.

Within this research we also introduced and elaborated on the heatGUIde system, a novel, user-centric solution designed meticulously for effective and efficient indoor heat stress mitigation in households. The system distinguishes itself by melding affordability, intuitive design, proactive advisory features, and the integration of self-learning algorithms. Each attribute is not standalone but intricately interwoven to construct a holistic solution that actively navigates and manages indoor environments to safeguard occupants during heatwaves.

7 Acknowledgments

8 References

[1] Umweltbundesamt, *Effects of climate change clearly noticeable in Germany.* [Online]. Available: https://www.umweltbundesamt.de/en/press/pressinformation/effects-of-climate-change-clearly-noticeable-in (accessed: Oct. 12 2023).

[2] F. S. Arsad, R. Hod, N. Ahmad, M. Baharom, and M. H. Ja'afar, "Assessment of indoor thermal comfort temperature and related behavioural adaptations: a systematic review," *Environ Sci Pollut Res*, vol. 30, no. 29, pp. 73137–73149, 2023, doi: 10.1007/s11356-023-27089-9.

[3] K. Lundgren Kownacki, C. Gao, K. Kuklane, and A. Wierzbicka, "Heat Stress in Indoor Environments of Scandinavian Urban Areas: A Literature Review," *International journal of environmental research and public health*, vol. 16, no. 4, 2019, doi: 10.3390/ijerph16040560.

[4] J. Pfafferott, S. Rißmann, G. Halbig, F. Schröder, and S. Saad, "Towards a Generic Residential Building Model for Heat-Health Warning Systems," *International journal of environmental research and public health*, vol. 18, no. 24, 2021, doi: 10.3390/ijerph182413050.

[5] M. Zhang *et al.,* "Deep Learning in the Era of Edge Computing: Challenges and Opportunities," in *Fog Computing*, A. Zomaya, A. Abbas, and S. Khan, Eds.: Wiley, 2020, pp. 67–78.

[6] G.C. Madhu, P. Vijayakumar, and X.Z. Gao, "Resource constrained IOT environments: A survey," *Journal of Advanced Research in Dynamical and Control Systems*, vol. 9, no. 16, pp. 445–457, 2017. [Online]. Available: https://www.researchgate.net/publication/329310950_Resource_constrained_IOT_environments_A_survey

[7] J. Yu, W.-S. Chang, and Y. Dong, "Building Energy Prediction Models and Related Uncertainties: A Review," *Buildings*, vol. 12, no. 8, p. 1284, 2022, doi: 10.3390/buildings12081284.

[8] Y. Li, Z. O'Neill, L. Zhang, J. Chen, P. Im, and J. DeGraw, "Grey-box modeling and application for building energy simulations - A critical review," *Renewable and Sustainable Energy Reviews*, vol. 146, p. 111174, 2021, doi: 10.1016/j.rser.2021.111174.

[9] *GitHub - czagoni/darkgreybox: DarkGreyBox: An open-source data-driven python building thermal model inspired by Genetic Algorithms and Machine Learning.* [Online]. Available: https://github.com/czagoni/ darkgreybox/tree/master (accessed: Oct. 13 2023).

[10] Z. Shi, G. Newsham, A. Pardasani, and H. B. Gunay, "On Formulation and Training of Grey-box Thermal Model for Low-rise Residential Buildings," in *Proceedings of Building Simulation 2019: 16th Conference of IBPSA*, Rome, Italy, 2020, pp. 838–844. Accessed: Dec. 10 2023. [Online]. Available: http://www.ibpsa.org/ proceedings/BS2019/BS2019_210251.pdf

Reducing Complexity of Deep Learning for Time Series Classification Using New Hand-Crafted Convolution Filters

Ali Ismail-Fawaz[1], Maxime Devanne[1], Stefano Berretti[2], Jonathan Weber[1], and Germain Forestier[1,3]

[1] IRIMAS, Université de Haute-Alsace, Mulhouse France
{ali-el-hadi.ismail-fawaz, maxime.devanne, jonathan.weber, germain.forestier}@uha.fr
[2] MICC, University of Florence, Folrence Italy
stefano.berretti@unifi.it
[3] DSAI, Monash University, Melbourne Australia

Abstract. Deep learning for Time Series Classification (TSC) has become one relevant subject in the literature for this task. It is used for wide applications in multiple domains ranging from medical data, action recognition and robotics. In the last decade, Convolutional Neural Networks (CNNs) have shown to be the best base architecture to use when dealing with deep learning for TSC ever since the release of the UCR archive, the largest repository for TSC datasets. The UCR archive includes a variety of 128 datasets of univariate time series data, where the task is to correctly classify the samples to their corresponding annotation. Deep learning models face two main challenges. The first one is represented by overfitting and the consequent incapacity of generalizing to new unseen samples. With CNN based architectures, this is commonly due to the fact that the learned filters tend to detect specific patterns in the training set instead of generic ones. The second challenge is complexity wise, which limits its usability in real world scenarios such as embedded systems. In this work, we propose to address these two challenges with one solution: hand-crafting some generic non-learned convolutional filters to detect generic patterns. These hand-crafted filters can replace the usability of the first layer in the CNN model, resulting in a significant reduction in the number of parameters. The proposed architecture is evaluated on 128 datasets of the UCR archive and the results reveal a significant improvement in performance compared to other approaches as well as the reduction in terms of complexity.

Keywords: Time Series Classification, Deep Learning, Time Series, Hand-Crafted Filters

1 Introduction

Time Series is a type of sequential data that is almost in every domain these days. Many tasks can be applied over this kind of data, including averaging [1], data augmentation [2], clustering [3], regression [4], classification [5, 6]. Time Series Classification (TSC) is extensively investigated in the literature such as the task of surface recognition by robots [7]. For instance, TSC can be used for evaluating surgical performance, [8] human motion action recognition, [9, 10] surface type detection from robots movements [7] etc. The availability of the UCR archive [11] made it possible to test multiple machine learning

tools to be benchmarked over the 128 datasets available. Ismail Fawaz et al. [12] presented a comprehensive review with exhaustive experiments to compare deep learning models for the task of TSC. This review concluded that Convolutional Neural Networks (CNNs) based architectures are more suitable for the task of TSC on the UCR archive. One of the winning CNN based architecture is the Fully Convolutional Network (FCN) [13], made of three convolutional layers in cascade followed by Batch Normalization and ReLU activation function. Moreover, the authors in [14] proposed InceptionTime, adapted from Inception-v3 for image classification. InceptionTime is currently the state-of-the-art for the task of TSC on the UCR archive. Some other work addressed deep learning for TSC using self-supervised approach [15] and knowledge distillation [16]. Even though deep learners present a success for the task of TSC, nevertheless, they do suffer from two main problems. First, deep learners tend to overfit on the training examples, which leads, in the case of CNNs, to learning filters to detect specific unique patterns in the training examples. Second, deep learners have large complexity, this limits their deployment into real world machines such as embedded systems.

In this work, we address these two problems and propose new hand-crafted convolution filters for time series data. We define three different hand-crafted filters to detect generic patterns that are independent from the data: increasing trend, decreasing trend and peaks. These filters, given that they are generic and non-learned, can help overcome the overfitting problem when replacing the first convolutional layer in the network. By combining the proposed hand-crafted convolution filters with the FCN [13] architecture, we proposed the Custom Only FCN (CO-FCN), which replaces the first learned layer in FCN by the hand-crafted filters.

We show that the CO-FCN reduces the number of parameters of FCN by almost 47%. Evaluation on the 128 datasets of the UCR archive show that CO-FCN can outperform FCN with a statistical significance in difference of performance. This is the higlight of our contributions in this work:

- Proposing new hand-crafted convolution filters;
- Adapting an existing CNN network for Time Series Classification to use the hand-crafted filters to reduce its number of parameters and increase its performance;
- Extensive experiments on 128 datasets of the UCR archive to evaluate the performance of the hand-crafted filters and the proposed architecture.

2 Proposed Method

2.1 Definitions

First, we list some definitions to facilitate the understanding of the rest of this work:

Univariate Time Series: Let $\mathbf{x}$ be a univariate time series of length L, a sequence of data points equally separated in time.

Univariate Time Series Dataset: A dataset $\mathcal{D} = \{(\mathbf{x}_0, y_0), ..., (\mathbf{x}_N, y_N)\}$ is a set of N pairs of univariate time series of length L and a label y associated to it.

One Dimensional Convolution: An operation using a filter $\mathbf{w}$ of length k on a time series $\mathbf{x}$ to obtain $\mathbf{s} = \mathbf{x} * \mathbf{w}$ as follows:

$$\forall t \in [0; L-1] \quad \mathbf{s}[t] = \sum_{i=0}^{k-1} \mathbf{x}[t+i].\mathbf{w}[i] \tag{1}$$

A detailed version of convolution on time series can be seen in Figure 1.

Activation of a Filter: When the convolution operation results in a positive response, the filter is considered as activated.

Increasing Trend: A sub-sequence of a time series **x**, where the values are strictly increasing in time.

Decreasing Trend: A sub-sequence of a time series **x**, where the values are strictly decreasing in time.

Stationary Trend: A sub-sequence of a time series **x**, where the values vary of a small difference ϵ.

Peak: A sub-sequence of a time series **x**, where the values changed with a large variation increasingly and then decreasingly.

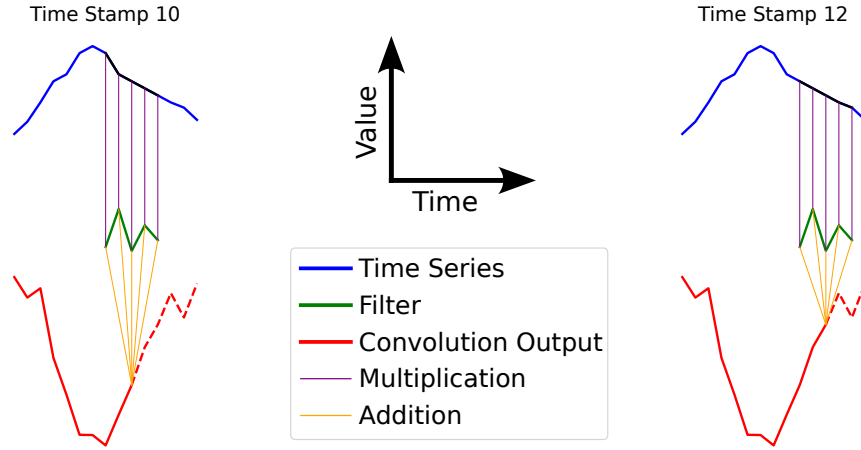In what follows, we detail the hand-crafted convolution filters.

Fig. 1: One dimensional convolution filter being convolved with a one dimensional time series.

2.2 Hand-Crafted Convolutional Filters

A summary of the three hand-crafted filters proposed in this work is given in Figure 2. The information gathered from the gradient of the time series can give rise to the increasing and decreasing trends. For this reason, both increasing and decreasing trends detection filters are simply an oscillation between -1 and 1. Motivated by the Sobel filters proposed on image contour detection [17], we adapt to a one-dimensional case the detection of peaks by mimicking the inverse of the second order derivative of the Gaussian.

2.3 FCN Adaptation: Custom Only FCN (CO-FCN)

By replacing the first convolutional layer in the original FCN architecture and replacing it by the hand-crafted filters, we obtain the Custom Only FCN (CO-FCN) presented in Figure 3. The FCN model presents $264,704$ trainable parameters, where as CO-FCN has only $122,496$ parameters to train.
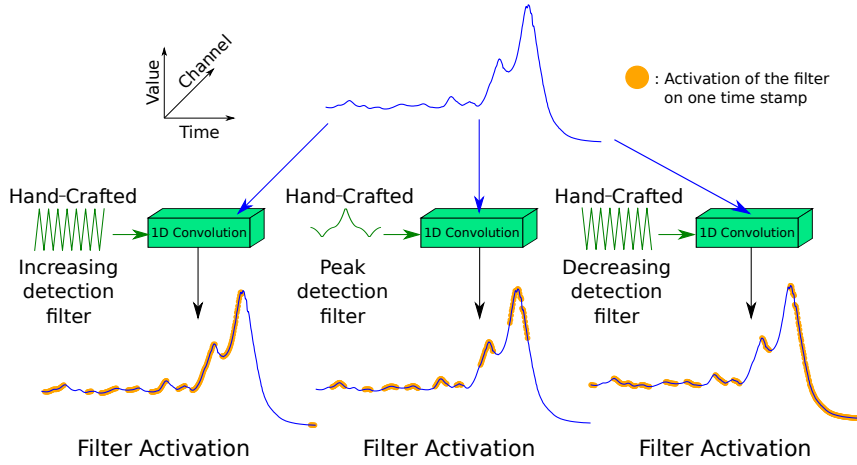
Fig. 2: The three hand-crafted convolutional filters applied on the Meat dataset of the UCR archive.
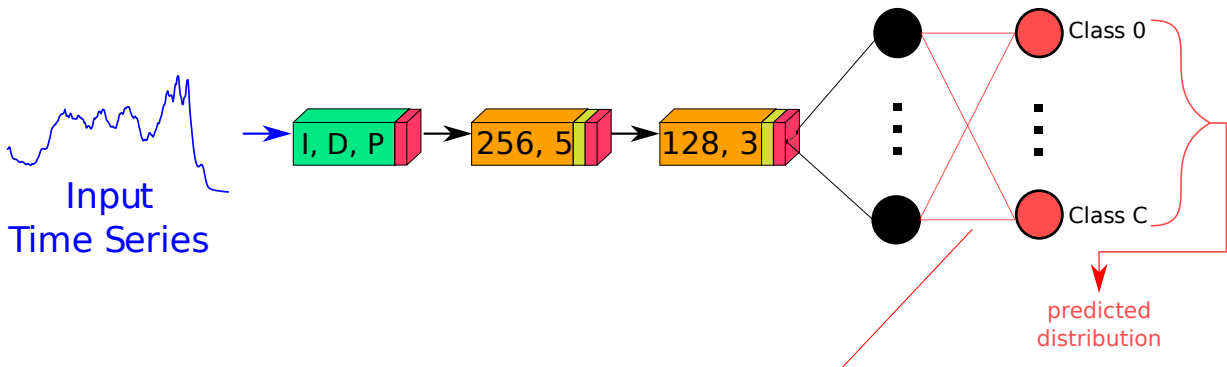

Fig. 3: The Custom Only FCN (CO-FCN) architecture. The hand-crafted filters are used in the first layer (in green) followed by a ReLU activation (in magenta). The second and third layers are made of convolution blocks (in orange: (n_filters, kernel_size)) followed by batch normalization (in oily) and ReLU activation. The last layers are composed of a 1D global average pooling (in black) and a linear classification layer (in red).

3 Results and Discussion

3.1 Experimental Setup

To have a fair comparison between FCN and CO-FCN, we used the same number of epochs, the same optimizer and its initial parameters and the same batch size. We also used a learning rate decay that monitors the training loss function. Five different initialization are done and the performance presented in this work is the average over all of them. The best model on the training loss during training is saved and used for the evaluation phase.

We use the set of 128 datasets of the UCR archive [11], where each dataset is split into train and test sets. The evaluation is done over the test set using the classification accuracy. All of the datasets are z-normalized before training in order to have a zero mean and unit variance.
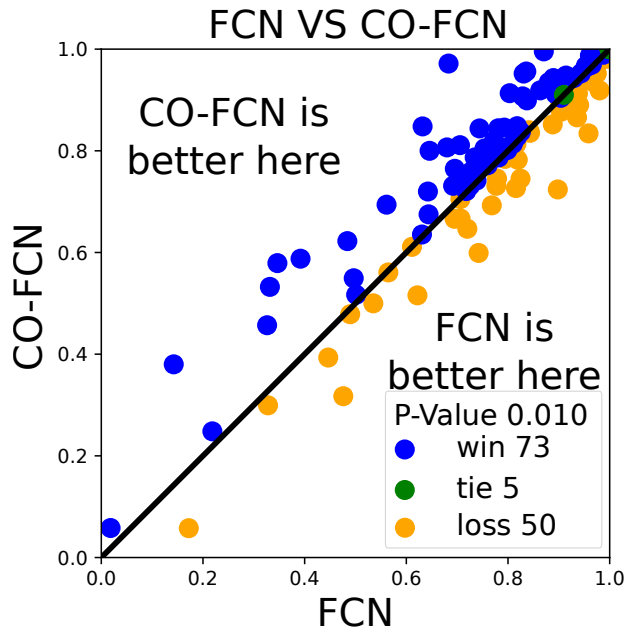
Fig. 4: One-vs-One scatter plot between FCN and CO-FCN. Each point represents one dataset, on the x-axis the accuracy of the FCN is presented and the one of CO-FCN on the y-axis.

3.2 CO-FCN vs FCN

To compare the performance of those two models, we evaluated their performance using the accuracy metric on the test set of all the datasets of the UCR archive. In Figure 4, the One-vs-One scatter plot between FCN and CO-FCN is presented. We counted the number of times each model wins and the number of ties, the difference in performance helps to produce the p-value (see legend in Figure 4) produced by the Wilcoxon Signed Rank test [18]. This p-value represents a % of statistical significance for the difference in performance between two comparates. This %, the p-value, if less than a given threshold (usually set to 5%) means that the difference in performance is statistically significant. If that last condition is not true then no conclusion can be made on the significance of difference in performance. This comparison concludes that CO-FCN outperforms FCN with a difference in performance that is statistical significant.

3.3 Comparing with State-of-the-Art

To compare the performance of CO-FCN to other deep learning approaches, we present the Multi Comparison Matrix (MCM) [19] in Figure 5. The MCM presents a pairwise comparison and a multi-classifier comparison at the same time. Each cell presents the Win/Tie/Loss count between two classifiers and the difference in average accuracy and the p-value using the Wilcoxon test. The MCM in Figure 5 also orders the competitors following their average performance on the accuracy metric over all datasets used. CO-FCN, even though coming third on the average performance, it can be seen that it beats significantly FCN and is not statistically significant than ResNet, which is 4.1 larger than CO-FCN in terms of number of parameters.
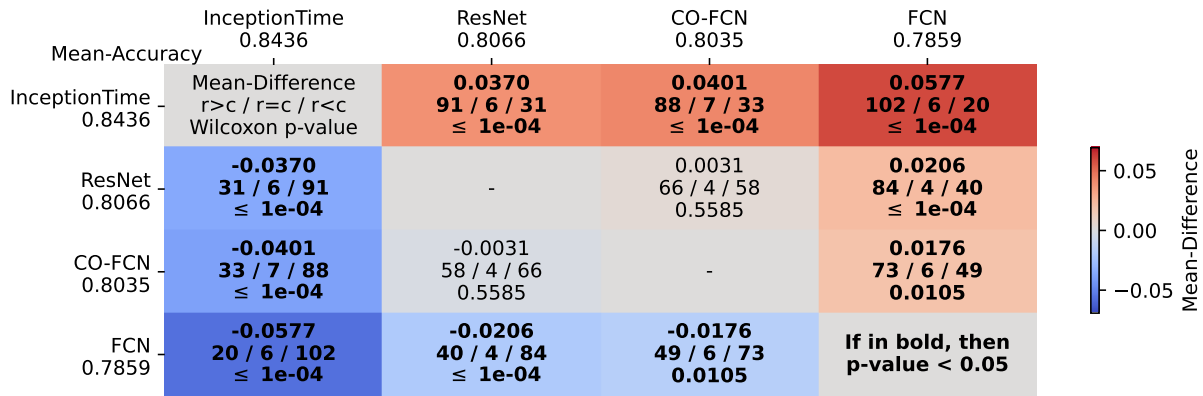
Mean-Accuracy	InceptionTime 0.8436	ResNet 0.8066	CO-FCN 0.8035	FCN 0.7859
InceptionTime 0.8436	Mean-Difference r>c / r=c / r<c Wilcoxon p-value	**0.0370** **91 / 6 / 31** **≤ 1e-04**	**0.0401** **88 / 7 / 33** **≤ 1e-04**	**0.0577** **102 / 6 / 20** **≤ 1e-04**
ResNet 0.8066	**-0.0370** **31 / 6 / 91** **≤ 1e-04**	-	0.0031 66 / 4 / 58 0.5585	**0.0206** **84 / 4 / 40** **≤ 1e-04**
CO-FCN 0.8035	**-0.0401** **33 / 7 / 88** **≤ 1e-04**	-0.0031 58 / 4 / 66 0.5585	-	**0.0176** **73 / 6 / 49** **0.0105**
FCN 0.7859	**-0.0577** **20 / 6 / 102** **≤ 1e-04**	**-0.0206** **40 / 4 / 84** **≤ 1e-04**	**-0.0176** **49 / 6 / 73** **0.0105**	**If in bold, then p-value < 0.05**

Mean-Difference scale: 0.05 / 0.00 / −0.05

Fig. 5: A Multi Comparison Matrix (MCM) benchmarking the state of the art models including the proposed CO-FCN architecture.

4 Conclusions

We presented in this work new hand-crafted convolution filters that are non-learned and generic to any time series data. These filters are able to replace the first convolutional layer of the Fully Convolutional Network proposed for TSC. This replacement constructs the CO-FCN with almost half the number of parameters of FCN. The proposed model is not only less complex than the FCN but also outperforms it on the majority of the UCR archive datasets. This new approach would help embed deep learners for TSC tasks into small embedded systems such as robotics systems.

5 Acknowledgements

References

1. Ismail-Fawaz, A., Ismail Fawaz, H., Petitjean, F., Devanne, M., Weber, J., Berretti, S., Webb, G.I., Forestier, G.: Shapedba: Generating effective time series prototypes using shapedtw barycenter averaging. In: ECML/PKDD Workshop on Advanced Analytics and Learning on Temporal Data. (2023)
2. Ismail Fawaz, H., Forestier, G., Weber, J., Idoumghar, L., Muller, P.A.: Data augmentation using synthetic data for time series classification with deep residual networks. In: ECML/PKDD Workshop on Advanced Analytics and Learning on Temporal Data. (2018)
3. Holder, C., Middlehurst, M., Bagnall, A.: A review and evaluation of elastic distance functions for time series clustering. Knowledge and Information Systems (2023) 1–45
4. Guijo-Rubio, D., Middlehurst, M., Arcencio, G., Silva, D.F., Bagnall, A.: Unsupervised feature based algorithms for time series extrinsic regression. arXiv preprint arXiv:2305.01429 (2023)

5. Middlehurst, M., Schäfer, P., Bagnall, A.: Bake off redux: a review and experimental evaluation of recent time series classification algorithms. arXiv preprint arXiv:2304.13029 (2023)

6. Ismail-Fawaz, A., Maxime, D., Stefano, B., Jonathan, W., Germain, F.: Lite: Light inception with boosting techniques for time series classification. In: International Conference on Data Science and Advanced Analytics (DSAA). (2023)

7. Vail, D., Veloso, M.: Learning from accelerometer data on a legged robot. IFAC Proceedings Volumes **37**(8) (2004) 822–827

8. Ismail Fawaz, H., Forestier, G., Weber, J., Idoumghar, L., Muller, P.A.: Evaluating surgical skills from kinematic data using convolutional neural networks. In: Medical Image Computing and Computer Assisted Intervention–MICCAI 2018: 21st International Conference, Granada, Spain, September 16-20, 2018, Proceedings, Part IV 11, Springer (2018) 214–221

9. Devanne, M., Wannous, H., Berretti, S., Pala, P., Daoudi, M., Del Bimbo, A.: 3-d human action recognition by shape analysis of motion trajectories on riemannian manifold. IEEE transactions on cybernetics **45**(7) (2014) 1340–1352

10. Devanne, M., Wannous, H., Berretti, S., Pala, P., Daoudi, M., Del Bimbo, A.: Space-time pose representation for 3d human action recognition. In: New Trends in Image Analysis and Processing–ICIAP 2013: ICIAP 2013 International Workshops, Naples, Italy, September 9-13, 2013. Proceedings 17, Springer (2013) 456–464

11. Dau, H.A., Bagnall, A., Kamgar, K., Yeh, C.C.M., Zhu, Y., Gharghabi, S., Ratanamahatana, C.A., Keogh, E.: The ucr time series archive. IEEE/CAA Journal of Automatica Sinica **6**(6) (2019) 1293–1305

12. Ismail Fawaz, H., Forestier, G., Weber, J., Idoumghar, L., Muller, P.A.: Deep learning for time series classification: a review. Data mining and knowledge discovery **33**(4) (2019) 917–963

13. Wang, Z., Yan, W., Oates, T.: Time series classification from scratch with deep neural networks: A strong baseline. In: 2017 International joint conference on neural networks (IJCNN), IEEE (2017) 1578–1585

14. Ismail Fawaz, H., Lucas, B., Forestier, G., Pelletier, C., Schmidt, D.F., Weber, J., Webb, G.I., Idoumghar, L., Muller, P.A., Petitjean, F.: Inceptiontime: Finding alexnet for time series classification. Data Mining and Knowledge Discovery **34**(6) (2020) 1936–1962

15. Ismail-Fawaz, A., Devanne, M., Weber, J., Forestier, G.: Enhancing time series classification with self-supervised learning. In: 15th International Conference on Agents and Artificial Intelligence: ICAART 2023, INSTICC (2023)

16. Ay, E., Devanne, M., Weber, J., Forestier, G.: A study of knowledge distillation in fully convolutional network for time series classification. In: 2022 International Joint Conference on Neural Networks (IJCNN), IEEE (2022) 1–8

17. Bogdan, V., Bonchiş, C., Orhei, C.: Custom extended sobel filters. arXiv preprint arXiv:1910.00138 (2019)

18. Wilcoxon, F.: Individual comparisons by ranking methods. In: Breakthroughs in statistics. Springer (1992) 196–202

19. Ismail-Fawaz, A., Dempster, A., Tan, C.W., Herrmann, M., Miller, L., Schmidt, D.F., Berretti, S., Weber, J., Devanne, M., Forestier, G., et al.: An approach to multiple comparison benchmark evaluations that is stable under manipulation of the compare set. arXiv preprint arXiv:2305.11921 (2023)

Keynote 2 : AI in the Contemporary and Future Battlefield: Revolutionizing Defense

Richard ARNING

Head of Department
Deutsch-Französisches Forschungsinstitut Saint-Louis (ISL)

Fig. 1. Dr Richard ARNING (ISL)

As the landscape of warfare rapidly evolves in the digital age, artificial intelligence (AI) stands at the forefront of transformation. In this keynote address at the AI Conference hosted by TrirhenaTech, we delve into the critical topic of "AI in the Contemporary and Future Battlefield." This discussion will illuminate the pivotal role of AI in defense applications and explore cutting-edge research activities in the field, with a special focus on the contributions of the Franco-German Defence Research Institute Saint-Louis (ISL).

With the convergence of AI and defense, we will navigate the profound impact of AI technologies in enhancing military capabilities, decision-making processes, and strategic operations. By examining real-world examples and future projections, we will unveil how AI is shaping the contemporary battlefield and what it means for the defense landscape of tomorrow.

Join us for an insightful exploration of AI's influence on defense, with a particular emphasis on the collaborative efforts of ISL in pushing the boundaries of technological innovation to ensure the safety and security of our nations. This keynote promises to provide a comprehensive overview of AI's pivotal role in defense and a glimpse into the future of military strategies, intelligence, and security.

Disclaimer: The above abstract was entirely composed by Chat GPT.

AI for Robotics

Intelligent vehicles semantic segmentation using evidential deep learning

Dănuţ-Vasile Giurgi[1], Mihreteab Negash Geletu[2], Thomas Josso-Laurain[1], Maxime Devanne[1], Jean-Philippe Lauffenburger[1], Mengesha Mamo Wogari[2]

[1] IRIMAS-UR7499 – Universite de Haute-Alsace, Mulhouse, France
[2] Addis Ababa University, Addis Ababa, Ethiopia
Email: vasile.giurgi@uha.fr

Abstract. Autonomous cars encounter momentous challenges in the perception tasks. The driving surrounding areas are more and more congested and the weather conditions differ significantly. Sensors-wise the capacities have increased, leading to an increasing interest in big data management such as artificial intelligence. Currently, neural networks have proved their efficiency, but restraints in complex situations are still present. In this work, a cross-fusion technique that combines lidar and camera data using an encoder-decoder-based model is proposed. The multi-modal architecture fuses different sources of information to circumvent encountered limitations. The considered perception task is semantic segmentation of the different obstacles that may be encountered. The decision-making part of the architecture is extended with the evidence theory, introducing belief functions that contribute in handling uncertainties. Thus, the evidential formulation is versatile and yields more precise predictions and a better understanding of the vacuous data. The dataset used in this work employs the KITTI dataset for semantic segmentation. The results show the interest of integrating evidential theory into neural networks fusing information from two heterogeneous sensors.

Keywords: cross-fusion, evidential deep-learning, perception, uncertainty.

1   Introduction

One important task in autonomous driving is to perceive the environment to achieve semantic analysis. This allows sending accurate information for the path planning and the control part to be applied. For the vision of the car, perception adopts substantially the deep learning-based approaches and the multi-sensor information to achieve strong capabilities. A big part of the actual self-driving car features rely on neural networks, especially deep learning, as their standard approach for perception tasks.

The popularity of this field soared when a deep convolutional neural network architecture named AlexNet outperformed other methods in a visual recognition challenge called ImageNet [1]. As a result, deep learning is now widely applied to various perception tasks in self-driving cars, including classification, object detection, semantic segmentation, and more [2,3].

In this algorithm, the output of the prediction frequently depends on the sigmoid function for binary predictions and the softmax function for multi-class predictions. These outputs represent probabilities assigned to a group of exclusive prediction categories. Nonetheless, probabilities do not always consistently manage uncertainty effectively, as they can struggle to distinguish between the absence of information and conflicting information. Withal, performances are often affected, and probabilistic models can struggle

to predict uncertainties and conflicting information. Hence for uncertainties, missing information, and imprecision, there are other effective techniques that exist such as the theory of evidence [4].

This theory, known as the Belief Functions (BF) theory or Dempster-Shafer theory (DS/DST), was initially proposed by Dempster and Shafer [4]. DS theory serves to represent belief elements for uncertain models and emphasizes several key features such as generality, operationality, or scalability [5]. In autonomous driving tasks, such as obstacle avoidance, belief functions have demonstrated their ability to provide accurate results. For instance, they improve the performance of LiDAR sensor occupancy grid maps by representing conflicts in a more meaningful manner [6].

In detecting pedestrians applications, evidential combination rules have consistently outperformed some Bayesian approaches [7]. Moreover, in multi-modal perception, evidential theory sets itself through the handling of missing information, imprecision, and ignorance.

In a multimodal approach, Xu et al. [8] integrated KITTI semantic segmentation images from various sensors, including cameras and different LiDAR layers. Their approach enables the expansion of object classes or the integration of additional sensors, resulting in improved performance and a better understanding of the drivable area.

In the context of self-navigation tasks relying on neural networks, as demonstrated in [9], a deep learning architecture based on Multi-Layer Perceptron (MLP) has been devised for the categorization of arbitrary LiDAR entities to enhance perception. This architecture departs from the probabilistic approach and adopts an evidential inference method, drawing inspiration from Denœux's generalized logistic classifier [10].

The application of evidence theory is employed to represent elements of evidence (beliefs) in the context of uncertain predictions by models. Consequently, the adoption of an evidential formulation shows promising outcomes (road segmentation and multi-object detection) in the realm of self-navigation embedded systems, which are the primary focus of this work.

The objective of this work is to develop an evidential deep learning model that integrates information from various sensors (camera and lidar) to enable autonomous driving capabilities. The work aims to offer decision-making rules that can explicitly produce cautious judgments to handle conflicts, whether originating from a single source of information or multiple sources (data fusion).

A cross-fusion model is augmented with evidence theory to achieve semantic segmentation performances within the KITTI dataset[11]. Thus, the workflow starts with a probabilistic approach, where the predictions are road, vehicles, and background and it is thereafter extended to an evidential formulation where the predictions are enlarged with a new class named ignorance (introduced by the the evidence theory), despite being trained on the road, vehicle, and background labels.

Through the leverage of theory-based belief functions, the model improves the decision-making part and allows representing uncertain prediction, which constitutes the primary contribution of this work.

The paper's organization continues through the subsequent sections: Background (covering the belief functions theory basics) Proposed method (illustrating some implementation details about the evidential formulation), Semantic segmentation results (encompassing data pre-processing and presenting results), and Conclusion.

2 Background

Evidence theory is a formal framework for making decisions in the presence of uncertainty. One method of applying evidence theory is through the utilization of the Dempster-Shafer rules. For a more comprehensive discussion, please refer to the following source: [12, 13].

Consider a finite set comprising elements denoted as:

$$\Omega = \{\omega_1, \omega_2, \omega_3, ..., \omega_n\} \tag{1}$$

called the *frame of discernment* (FoD). This collection represents mutually exclusive elements of a single cardinality, called *singletons*.

A *basic belief assignment* (BBA) or a piece of evidence is characterized by a function (mass functions) $m : 2^\Omega \to [0, 1]$ such that:

$$m(\emptyset) = 0 \tag{2}$$

$$\sum_{A \subseteq \Omega} m(A) = 1 \tag{3}$$

The quantity $m(A)$, known as the mass function, measures the credibility that one adheres exactly to hypothesis A; and (2) serving as an indicator of the closed world assumption [14]. If $m(A) > 0$, A represents a *focal element* of m.

Considering a basic belief assignment m, two notions can be disclosed, *a credibility or a belief function* (Bel) and a *plausibility function* (Pl) using the following expressions:

$$Bel(A) = \sum_{B \subseteq A} m(B) \tag{4}$$

$$Pl(A) = \sum_{B \cap A \neq \emptyset} m(B) = 1 - Bel(\bar{A}) \tag{5}$$

$Bel(A)$ can be understood as the degree of total support to A, whereas $Pl(A)$ represents the extent to which doubt is absent. A.

If the frame of discernment is represented by a single focal element that is Ω, then the BBA m is considered *vacuous* and characterizes total *ignorance*.

Two mass functions m_1 and m_2 representing independent pieces of evidence can be merged by Dempster's rule which is defined as follows:

$$(m_1 \oplus m_2)(A) = \frac{1}{1-k} \sum_{B \cap C = A} m_1(B) m_2(C) \tag{6}$$

For all $A \subseteq \Omega$, $A \neq \emptyset$, and $(m_1 \oplus m_2)(\emptyset) = 0$. The constant k represents the degree of conflict of the two BBAs and is expressed as:

$$k = \sum_{B \cap C = \emptyset} m_1(B) m_2(C) \tag{7}$$

In this work, the previously established concepts from evidence theory are integrated into an evidential framework that is plugged into a deep learning lidar-camera cross-fusion architecture. The primary objective is to leverage the respective advantages of these two frameworks to realize scene segmentation. The evidence theory is thus combined with the

streamlined neural network referred to as Lite-CF, described in the next section. Originally designed as a probabilistic model for road detection, Lite-CF generates probability distributions from logits through a softmax layer [15]. To improve the handling of prediction uncertainty, an evidential approach recreates the last part of the architecture by replacing the softmax layer and the associated logits with a belief theory-based approach.

3 Proposed method

To handle data from various sensors effectively, architectures have been devised, which integrate multiple fusion methods such as late fusion, deep fusion, early fusion, and others. In a prior study [15], a lidar-camera fusion approach named lite cross-fusion was introduced, employing a fully convolutional neural network for road detection [16]. This deep fusion network demonstrated superior performance compared to early or late fusion methods. Consequently, it was integrated into another work [15], resulting in a reduction of computation complexity by over 15%. The model is built upon an encoder-decoder architecture that employs dilated convolution to consolidate contextual details while preserving resolution. This road detection fusion network serves as the foundation for the research presented in this paper. On another hand, considering the combination between evidence theory and neural network, [17] proposes an evidential classifier with a distance to prototypes approach that replaces the softmax decision layer.

Considering the two previous methods, namely the aforementioned reduced cross-fusion road detection (Lite-CF), and the evidential classifiers, this paper introduces a fusion of these two approaches. The combined architecture, Lite CF-Evi is a combination designed for semantic segmentation tasks. The overall architecture of the evidential Lite-CF is given in Fig. 1: The system comprises an encoder-decoder network, an evidential formulation layer, and a decision-making unit. In the encoding segment, there exist two processing pipelines, each consisting of 13 layers: one for feeding the LiDAR input and the other for camera frames. At each layer, information from one modality is combined with the corresponding layer from the other modality through a trainable weighted sum operation ($*b_i$ and $*a_i$ respectively, where i is the layer number). These fusion weights are adaptable, allowing the fusion's position and its extent to be fixed by the data.

After the LiDAR and camera inputs are transformed into Basic Belief Assignments (BBAs) within the evidential formulation layer, decisions can be rendered concerning specific elements within the power set 2^{Ω}. In the context of semantic segmentation, this power set encompasses elements such as "road", "vehicle", and "background" elements in the probabilistic version, and additionally "ignorance" in the evidential formulation. Consequently, the evidential approach enables having an imprecise class prediction.

3.1 Evidential Inference

The evidential formulation layer takes as its input, the feature maps generated by the decoding section. When the decoder reaches its maximum resolution, Basic Belief Assignments (BBAs) are generated by assessing the distances between the corresponding feature maps (i.e., L18 in Fig. 1) and prototypes. In this way, the prototypes are learnt naturally, in an automatic manner. The technique is called distance to prototypes and it can be described in three steps [18] as follows:

Step 1: Calculate the distance to prototype: Consider x to be a feature vector symbolizing features of a pixel to be labeled possibly as road ω_1, vehicle ω_2 or
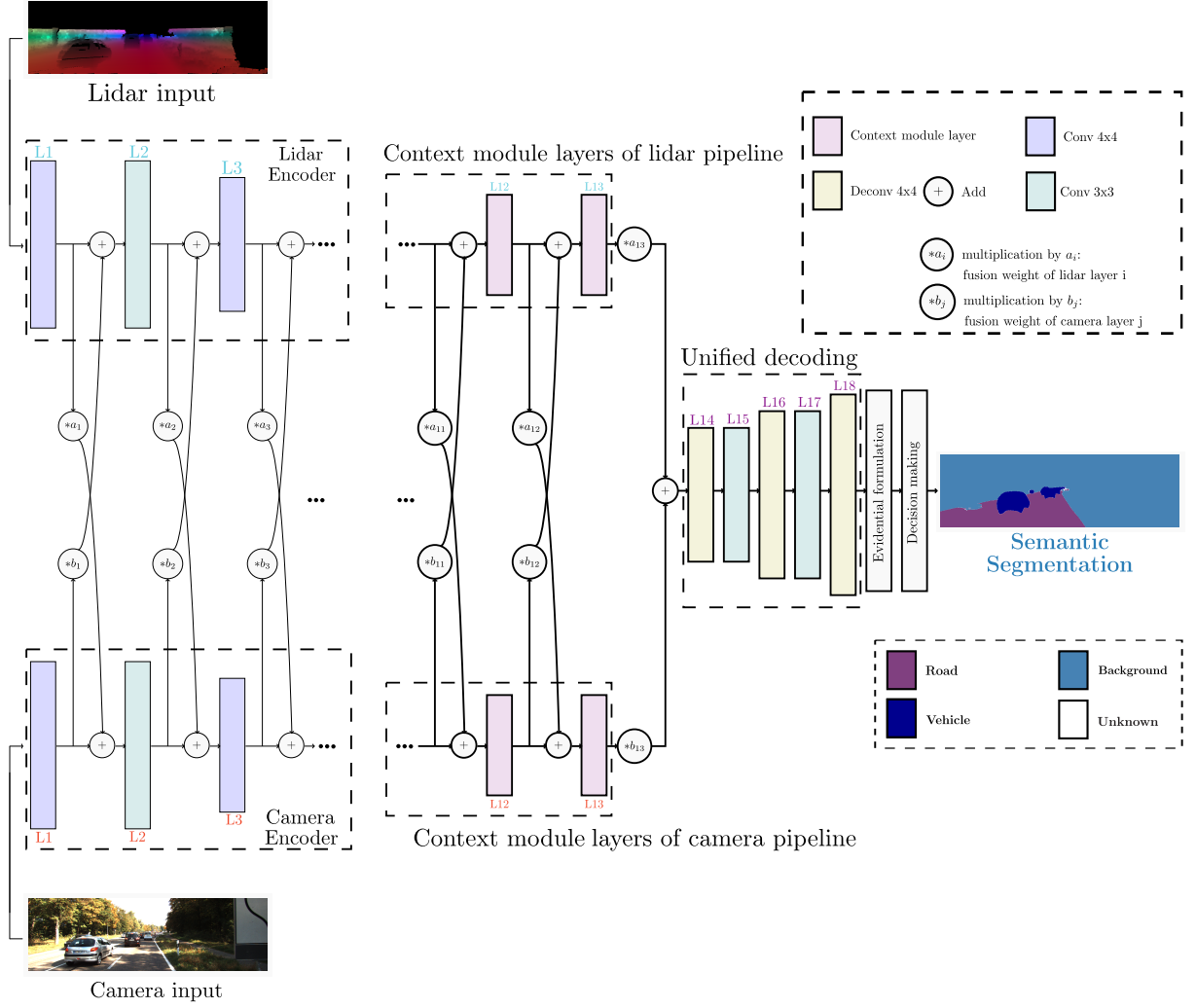
Fig. 1: Architecture Evidential Lite-CF (Lite CF-Evi)

background ω_3 (i.e., the FoD $\Omega = \{\omega_1, \omega_2, \omega_3\}$). The Euclidean distance d^i is realized between $\boldsymbol{x}$ and each prototype $\boldsymbol{p^i}$:

$$d^i = \|\boldsymbol{x} - \boldsymbol{p^i}\| \qquad i = 1, \cdots, n. \tag{8}$$

Step 2: Establish the correspondence of mass functions to prototypes and their interference: Each prototype $\boldsymbol{p^i}$ has a degree of membership u_j^i to each class ω_j, with a constraint $u_1^i + u_2^i + u_3^i = 1$. Using the class membership u_j^i and the distance d^i, a BBA m^i is constructed as:

$$\begin{aligned} m^i(\{\omega_j\}) &= \alpha^i u_j^i \phi^i(d^i), \qquad j = 1, 2, 3 \\ m^i(\Omega) &= 1 - \alpha^i \phi^i(d^i), \end{aligned} \tag{9}$$

where $0 < \alpha^i < 1$ and the function ϕ^i is defined as:

$$\phi^i(d^i) = \exp\!\big(-\gamma^i(d^i)^2\big), \qquad \gamma^i > 0 \tag{10}$$

Step 3: DS combination rule: The mass functions from step 2 are merged using Dempster's rule (see (6)). The outcome combined with BBAs serves as the evidence for determining the pixel class.

The parameters linked to the prototype $\boldsymbol{p^i}$ (i.e., α^i, u^i_j, and γ^i), are intended to be incorporated into the evidential deep learning-based architectures as weighting factors. However, the learnable weights are not inherently restricted. Consequently, they are redefined and expressed in terms of certain real-numbered variables η^i, ξ^i, and β^i_j:

$$\gamma^i = (\eta^i)^2 \tag{11}$$

$$\alpha^i = \frac{1}{1 + \exp\{-\xi^i\}} \tag{12}$$

$$u^i_j = \frac{(\beta^i_j)^2 + \epsilon}{\sum\limits_{k=1}^{2} \left((\beta^i_k)^2 + \epsilon\right)} \tag{13}$$

Equation (13) is slightly adjusted from the expression given in [18]. To prevent the membership values u^i_j from becoming zero, a small positive value denoted as ϵ is introduced. This precautionary measure is taken to limit the conflicts that could lead to Dempster's *total conflict*.

3.2 Decision making

After assessing the BBAs representing the evidence for each pixel, the ultimate objective is to decide the pixel classes. Initially, a decision based on interval distance is considered, as outlined in [19]. Here's an example with only 3 classes (in a semantic segmentation task: road, vehicle, and background):

Case i) The decision is constrained to singletons: The possible judgment elements are ω_1 (road), ω_2 (vehicle) and ω_3 (background). In this situation, the expression from [19] becomes:

$$\hat{X} = \arg \min_{X \in \{\omega_1, \omega_2, \omega_3\}} d_{BI}(m, m_X) \tag{14}$$

Case ii) The decision is not restricted: It could be interesting to consider assigning ambiguous pixels to less precise classes within Ω. The method can minimize classification errors by circumventing decisions that have more of an arbitrary nature.

4 Semantic segmentation results

4.1 Dataset

The aforementioned evidential cross-fusion model Lite-CF-Evi architecture is evaluated in the context of segmentation tasks against the KITTI semantic segmentation dataset. This dataset provides only 200 camera images similar to Stereo and Flow 2012/2015 datasets. Unfortunately, the dataset has no LiDAR frames. Consequently, the corresponding 3D point-cloud data for the camera images needs to be extracted from the big original raw dataset, as outlined in [20] with all the images from the KITTI dataset for all the tasks. Hence, 127 out of the 200 camera images have been successfully identified, along with their respective LiDAR frames. These LiDAR frames are subsequently projected and upsampled to create dense depth images. A 3D LiDAR point x is mapped into a point y in
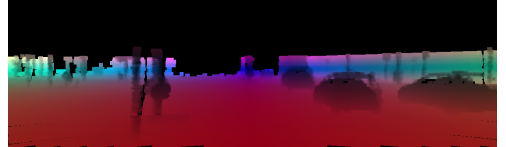
the camera plane according to the KITTI projection P, rectification R and translation T matrices:

$$y = P\,R\,T\,x \tag{15}$$

As the projected LiDAR scan is sparse, up-sampling is employed to generate a dense depth map, as depicted in Fig. 2. The up-sampling process is implemented following the method outlined in [15] and [21].

(a) Camera image

(b) Projected and up-sampled LiDAR points into the camera frame

Fig. 2: LiDAR pre-processing method

After the up-sampling process, the newly constructed dense depth images from LiDAR are integrated into the Lite-CF-Evi model in parallel with the camera images to feed the two pipeline inputs of the architecture.

Concerning the ground truth, the masks are simplified to 3 classes: road (magenta), vehicle (dark blue), and background (blue), according to the original annotation. The road class is preserved, however, the vehicle class incorporates car, truck, and bus annotations of the original ground truth. In turn, the background class encapsulates all the other classes, except for the above-mentioned ones. Figure 3a shows an illustration with an example of the original ground truth, while Figure 3b describes the simplified ground truth. The dataset consisted of 127 images: 114 for training and 13 for validation. This method has been exclusively assessed using the specially reconstructed KITTI semantic dataset, which includes the added LiDAR frames for the evidential cross-fusion architecture. To the best of the author's knowledge, this dataset has not yet been examined by any other methods, since LiDAR point clouds were included. The ground-truth masks are one-hot encoded and class weight is applied to address the unbalanced data. Consecutively, the model is trained for 500 epochs using mean squared error loss and Adam optimizer.

To measure the performances, the model is evaluated using the intersection-over-union metric, denoted as IoU, in accordance with the PASCAL VOC benchmark [22]:

$$IoU = \frac{TP}{TP + FP + FN} \tag{16}$$

with TP, FP, and FN, respectively, true positive, false positive, and false negative. The Lite-CF-Evi is evaluated for 3 classes in a probabilistic manner. It can be observed that the global mean IoU, 0.92707, in the evidential architecture is higher than 0.92384 for the probabilistic model. Individually over each class, the evidential model outperforms the probabilistic one (Table 4.1), and visually the results are better for the Lite-Cf Evi.

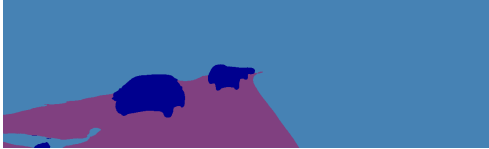Table 4.1: Model performance comparison

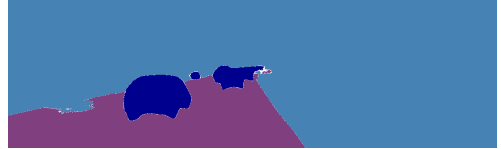| Model arch.	mean IoU	mean IoU$_{\text{road}}$	mean IoU$_{\text{vehicle}}$	mean IoU$_{\text{background}}$
Probab. Lite-CF	0.92384	0.92713	0.87118	0.97322
Lite-Cf Evi	**0.92707**	**0.93163**	**0.87446**	**0.97513**
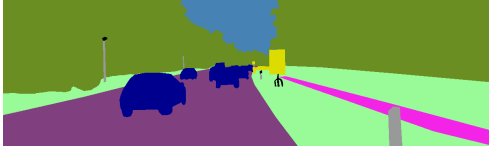
(a) Original ground truth: semantic

(b) Simplified ground truth, 3 classes: road, vehicle, background

(c) Probabilistic Prediction Lite-CF: road, vehicle, background

(d) Evidential Prediction with Lite-CF-Evi: road, vehicle, background, ignorance (white)

Fig. 3: Semantic segmentation results

One interesting part of the evidential formulation is that the decision-making can be adapted to derive from a fixed number of classes (equal to the number of singletons) to the maximum number of acts, $|2^{\Omega}| - 1$. However, often the desired decision elements are considered just the singletons, which are the main classes and the uncertainties. The third image below (Fig. 3c), on the left side represents the predicted image with the probabilistic model.

Finally, the second image from the right bottom part, Fig. 3d represents the predicted image with the evidential model (Lite-CF-Evi). It can be observed that classes road, vehicle, and background exhibit slightly higher accuracy in their predictions, with road class being notably precise. Furthermore, an additional class, denoted as "ignorance" (depicted in white), effectively captures pixels associated with uncertain predictions. This approach prevents the mis-classification of uncertain pixels into incorrect categories, a scenario that may arise when utilizing a probabilistic approach.

"Ignorance" primarily manifests itself at the class boundaries, where the model frequently provides errors in its predictions. Likewise, pixels from distant objects often lack sufficient information, suggesting that the model encounters challenges in classifying them due to data uncertainty. Consequently, these pixels are classified as "ignorance," offering improved comprehension and demonstrating the effectiveness of evidential reasoning in managing uncertainties.

5   Conclusion

In this paper, a camera-lidar fusion is proposed by using a deep learning architecture combined with evidence theory for intelligent vehicles perception. The combination is realized at the very last level, replacing the softmax decision with a decision based on distance to prototypes. The introduction of ignorance as a decision element further improves efficiency. Hence, distant points and ambiguous features can be categorized as "ignorance" rather than being erroneously assigned to specific predictions. Future directions involve enhancing the Lite CF-Evi model for various class configurations and more intricate tasks while maintaining the computational efficiency needed for real-time applications. Additionally, a more in-depth examination of the distribution and impact of "ignorance" is intended to be explored.

References

1. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al.: Imagenet large scale visual recognition challenge. International journal of computer vision **115**(3) (2015) 211–252
2. Giurgi, D.V., Josso-Laurain, T., Devanne, M., Lauffenburger, J.P.: Real-time road detection implementation of unet architecture for autonomous driving. In: IEEE 14th Image, Video, and Multidimensional Signal Processing Workshop (IVMSP), IEEE (2022) 1–5
3. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical image computing and computer-assisted intervention, Springer (2015) 234–241
4. Dempster, A.P.: A generalization of bayesian inference. Journal of the Royal Statistical Society: Series B (Methodological) **30**(2) (1968) 205–232
5. Beynon, M., Curry, B., Morgan, P.: The dempster–shafer theory of evidence: an alternative approach to multicriteria decision modelling. Omega **28**(1) (2000) 37–50
6. Laghmara, H., Boudali, M.T., Laurain, T., Ledy, J., Orjuela, R., Lauffenburger, J.P., Basset, M.: Obstacle avoidance, path planning and control for autonomous vehicles. In: 2019 IEEE Intelligent Vehicles Symposium (IV), IEEE (2019) 529–534
7. Xu, P., Davoine, F., Denœux, T.: Evidential combination of pedestrian detectors. In: British Machine Vision Conference. (2014) 1–14
8. Xu, P., Davoine, F., Bordes, J.B., Zhao, H., Denœux, T.: Multimodal information fusion for urban scene understanding. Machine Vision and Applications **27**(3) (2016) 331–349
9. Capellier, E., Davoine, F., Cherfaoui, V., Li, Y.: Evidential deep learning for arbitrary lidar object classification in the context of autonomous driving. In: 2019 IEEE Intelligent Vehicles Symposium (IV), IEEE (2019) 1304–1311
10. Denœux, T.: Logistic regression, neural networks and dempster-shafer theory: A new perspective. Knowledge-Based Systems **176** (2019) 54–67
11. Alhaija, H., Mustikovela, S., Mescheder, L., Geiger, A., Rother, C.: Augmented reality meets computer vision: Efficient data generation for urban driving scenes. International Journal of Computer Vision (IJCV) (2018)
12. Shafer, G.: A mathematical theory of evidence. Princeton University Press (1976)
13. Denœux, T., Dubois, D., Prade, H.: Representations of uncertainty in ai: beyond probability and possibility. In: A guided tour of artificial intelligence research. Springer (2020) 119–150

14. Martin, A., Jousselme, A.L., Osswald, C.: Conflict measure for the discounting operation on belief functions. In: 2008 11th International Conference on Information Fusion. (2008) 1–8
15. Geletu, M.N., Josso-Laurain, T., Devanne, M., Wogari, M.M., Lauffenburger, J.P.: Deep learning based architecture reduction on camera-lidar fusion for autonomous vehicles. In: 2nd International Conference on Computers and Automation (CompAuto), IEEE (2022) 25–31
16. Caltagirone, L., Bellone, M., Svensson, L., Wahde, M.: Lidar–camera fusion for road detection using fully convolutional neural networks. Robotics and Autonomous Systems **111** (2019) 125–131
17. Tong, Z., Xu, P., Denœux, T.: An evidential classifier based on dempster-shafer theory and deep learning. Neurocomputing **450** (aug 2021) 275–293
18. Denœux, T.: A neural network classifier based on dempster-shafer theory. IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans **30**(2) (2000) 131–150
19. Dezert, J., Han, D., Tacnet, J.M., Carladous, S., Yang, Y.: Decision-making with belief interval distance. In: International conference on belief functions, Springer (2016) 66–74
20. Geiger, A., Lenz, P., Stiller, C., Urtasun, R.: Vision meets robotics: The kitti dataset. International Journal of Robotics Research (IJRR) (2013)
21. Premebida, C., Carreira, J., Batista, J., Nunes, U.: Pedestrian detection combining rgb and dense lidar data. In: 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE (2014) 4112–4117
22. Everingham, M., Eslami, S., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The pascal visual object classes challenge: A retrospective. International journal of computer vision **111**(1) (2015) 98–136

Enhancing Independence through Intelligent Robotics: An AI-Driven Assistive Robotics Interface

Katrin-Misel Ponomarjova[1], Thomas M. Wendt[1], Anke Fischer-Janzen[1], Sinan Süme[1], Bastian Kayser[2]

[1] Offenburg University of Applied Sciences
katrin-misel.ponomarjova@hs-offenburg.de
[2] Cadida Software GmbH
kayser@cadida.de

Abstract. Applying methods in artificial intelligence to the field of assistive robotics has the potential to increase the independence of people with disabilities. The usage of AI to realize a shared control in this context is controversial, due to the high complexity of everyday tasks and the needed safety requirements. This paper presents the development of a user interface for AI-driven assistive robotic arms (ARA) that aims to assist people with physical disabilities in performing daily activities. This interface allows the user to select object manipulation tasks based on the objects recognized in a live video stream. Further, we compare several state-of-the-art, real-time object detection models to facilitate automatic robotic control. The results demonstrate the feasibility of the model and its potential integration into the overall robotic system.

Keywords: Assistive Robotics; Real-Time Object Detection; Human-Robot Interaction.

1 Introduction

The growing need for technological assistance in caretaking leads to increased developments in the field of assistive robotics. Currently, assistive robotic systems offer the user the possibility to directly control an ARA through various control inputs, such as eye tracking [1]. Physically impaired people with paraplegia or tetraplegia can be assisted with assistive robotic arms (ARA). Few models are available on the market, such as JACO, MICO (Kinova) or the iARM (Assistive Innovations bv). These ARA can be mounted on an electric wheelchair or on the side of the bed for bed bound patients. Reasons for such paralysis can be spinal injuries, stroke and various diseases such as cerebral palsy, amyotrophic lateral sclerosis and multiple sclerosis. According to recent statistics published by the German Federal Statistical Office, 7,8 million people in Germany have a severe disability of which approximately 1,6 million have a spinal or limb disability [2].

The design of interfaces to support these people is a crucial part of making a system usable and accessible, and presents a variety of challenges. Combining an ARA with other technologies, such as smart home appliances, lead to new opportunities. For example, the work of Brunete et al. shows that the user could interact over a tablet with individual input interfaces to control a mobile platform, a robotic arm, as well as IoT devices such as the heating, lighting, and shutters [3]. Furthermore, ARA can be controlled by joysticks, speech recognition, head or eye tracking, or even Brain Computer Interfaces (BCI). With the exception of joystick and voice control, most of these solutions are controlled via a display or tablet that provides the directional control of the robotic arm or selection of individual tasks the robot is capable of performing [4].

In this context the question arises as to whether usability, measured e.g. by ease of use [5], can be facilitated through the use of AI by displaying selectable items for interaction on these displays. Available state-of-the-art computer vision models enable us to detect objects in real time. In addition, advances in robot automation allow robots to generalize control tasks such as object manipulation to novel objects and environments.

In addition, a compromise must be found between the autonomy of the robot control and the controllability and interpretability by the user. Current work in AI and robotics shows promising results for the use of multimodal large language models (LLMs) to control robot agents [6, 7,8]. Here, tasks are given in natural language and the robot navigates and executes the given tasks based on the visual input. While this approach is promising for a variety of different use cases, most of the identified systems do not provide any visual feedback on which task or target objects have been identified or will be interacted with. In the context of assistive robotics, this can lead to a feeling of being at the mercy of a "black box" model.

In this work, we aim to develop a system that focuses on usability and accessibility by hiding the underlying complexity of robot control. This can be achieved through the use of an object detection model which simplifies the control and allows the user to accomplish complex tasks of everyday life.

2 Methods

2.1 System overview

To meet the needs of paraplegic people, the HIRAC (Hardware Independent Robotic Assistance Controller) project is developing a robotic system that enables people to perform activities of daily living.

The system consists of an ARA that can be mounted on an electric wheelchair or at the bedside. The different components communicate via the Robot Operating System 2 (ROS2 Humble Hawksbill). A camera is mounted on the robotic arm, allowing the user to move the field of view, which is necessary for bed-bound users. The touch display is mounted in the user's line of sight and displays a live video stream for ease of use.

2.2 Design of the User Interface (UI)

The four principles of accessibility outlined in the Web Content Accessibility Guidelines can be applied to the user interface presented in this work. They include perceptibility, usability, understandability, and robustness [9]. They resemble the seven principles of universal design presented by Story, which were developed as a benchmark for product design so that "people of all ages and abilities" can use them without the need of adaptation [5]. They consist of Equitable Use, Flexibility in Use, Simple and Intuitive Use, Perceptible Information, Tolerance for Error, Low Physical Effort, and Size and Space for Approach and Use [5]. Based on these principles, a requirements analysis was conducted to determine the critical design elements for a UI with high usability satisfaction. Existing solutions for this use case were identified and compared to this system. For the target group of paraplegic and tetraplegic persons, refinements in the accessibility were made. Regarding these insights, a mockup of the user interface and the user flow are presented in the following.

2.3 Evaluation methodology of the object detection network

The YOLO algorithm has become widely adopted in a variety of applications where real-time inference is required. It performs exceptionally well in terms of inference speed and detection

accuracy when compared to other real-time object detection algorithms such as Faster R-CNN (Faster Region based Convolutional Neural Networks) and SSD (Single Shot Detection) [10]. In addition, previous versions of YOLO have been successfully integrated with ROS2 [11], which is an essential part of the system. For these reasons, we decided to focus our attention on it.

Proposed method

Projects requiring real-time detection must consider the trade-off between accuracy and speed. Larger models tend to be slower in terms of FPS [12], but generally show better performance in terms of mAP (Mean Average Precision).

Therefore, we compare two different model sizes, nano ("n") and small ("s"), from YOLOv5 [13] and YOLOv8 [14] (both pre-trained on the MS COCO dataset), trained with and without data augmentation. A Nvidia RTX A2000 Laptop GPU (4096MiB) is used for training and evaluation.

As for the evaluation metrics used in this comparison, the mAP50-95 allows us to assess the precision of the detections, while the FPS will allow us to measure the real-time processing speed of the model. The FPS in this paper is calculated on the basis of average speed of the detection in milliseconds when processing the validation set, as shown in Equation (1). These metrics, as well as the Average Precision and Recall, can be found in Tab. 2.

$$FPS = \frac{1000}{preprocess_{ms} + inference_{ms} + postprocess_{ms}} \quad (1)$$

Dataset preparation

We create a dataset of images and annotations of keys, pens, cups and glasses by merging three datasets [15, 16, 17]. Initial poor performance in the "pen" class due to class imbalance was addressed by increasing the number of examples in this class from three additional datasets [18, 19, 20]. The final dataset used for training can be found on Github [21].

A data augmentation strategy is then implemented consisting of the modifications that showed the highest performance improvement: crops of up to 50% and Gaussian blur of up to 2px. This allowed us to generate 4198 additional images in the training set (Tab. 1).

As the objects to be detected are close and large, it is possible to train on smaller image sizes to decrease the training time. The images are trained over 10 epochs to see initial results. The image size used to train these networks is 320x320, then increased to 640x640 for the best performing networks trained over 40 epochs.

Table 1. Distribution of the dataset into training, validation and test sets

	Original			Data aug.
Class	*Train (70%)*	*Valid (20%)*	*Test (10%)*	*Train (70%)*
Keys	603	193	86	1809
Cup/Glass	1062	268	138	3186
Pen	434	139	75	1302
Total	**2099**	**600**	**299**	**6297**

3 Results

3.1 Design of a user interface for Assistive Robotics

After identifying the target group, a requirements analysis was carried out. Regarding the outcomes of Story [5], it is theorized that the user's cognitive workload can be reduced by automating the grasping process and task completion. With regard to the other benchmarks mentioned above, we focused on the usability and accessibility of the system. Several challenges were identified during the design process:

- How to present the available tasks to the user?
- How to ensure that the object detection model is able to identify the object and enable the task execution?
- What safety measures are necessary to ensure the safety of the user?
- What measures need to be taken to ensure usability and accessibility?

Safety regulations were not tested on the design, due to its early state. Controlling a robot at such a short distance to the user by AI raises new challenges and questions. Regulations of AI application are currently debated by the European Commission [22].

First, the tasks were determined. Activities of Daily Living (ADL) are describing a set of everyday tasks for self-maintaining. They are listed by the International Classification of Functioning, Disability and Health (ICF) [23]. Severe physically disabled people mostly need assistance from another person for these tasks, as robots are not able to help in some ADL such as bathing. We narrowed down the range of tasks which are feasible for the robot such as picking and placing objects, pouring a drink and handing it to the user, as well as handing food to the user. With these tasks, basic independence can be regained.

Secondly, the object detection model which will be presented in the next section is able to handle a moving workspace, considering that the ARA will be mounted on an electric wheelchair. The position of the robot has to be adapted to present the desired object. This leads to the importance of directional control and a fixed robot observation orientation. This was achieved by arrows on the sides, top and bottom of the touch screen visible in Fig. 1 which move the robot in the x- and y-planes. The robot's end effector is tilted around the y-axis at an angle of 35° to provide a bird's eye view of the scene. When the required object is detected in the scene, the system displays the bounding box around the object.

The bounding boxes are clickable. Once one is selected, the available tasks are presented to the user. When a task is selected, a progress bar is displayed representing the time the robot needs to complete the task.

Thirdly, rudimental safety features must be considered, such as a stop button that interrupts the task itself. Furthermore, measures such as a reset button to bring the robot back into position are needed to free the arm for movement or to resolve inadequate robot positions.

Lastly, the usability and accessibility of the system was examined. The need to eliminate seizure triggers was identified and implemented. Basic settings such as adapting the font size, adding pictograms to the texts, simple language and the possibility of feedback were included. Specific settings for the robot include the ability to adjust the speed, which was mentioned by participants in a previous study. It was considered important to design the UI in such a way that other input modalities, such as eye tracking, could be implemented at a later stage to ensure accessibility, for example for people with locked-in syndrome. All identified features are shown in the user flow in Fig. 1.
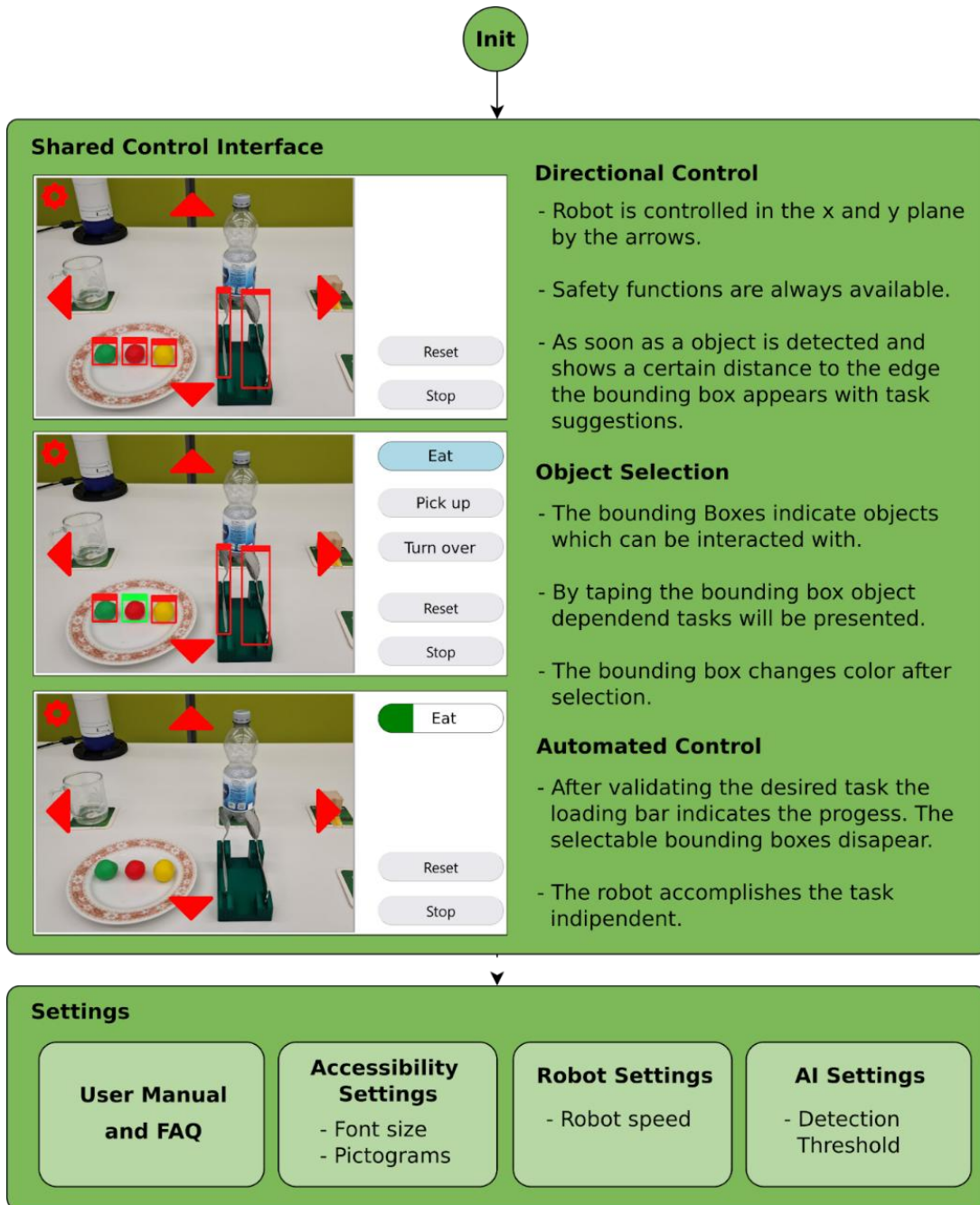
Fig. 1. User flow and mockup design of the user interface

3.2 Performance evaluation of computer vision models

Tab. 2 shows the results of training each network over 10 epochs with and without augmented data. Except for YOLOv8 trained on augmented data, there is a noticeable decrease in FPS as the network size was increased from "nano" to "small". There is also a noticeable improvement in mAP for models trained on augmented data. The last rows represent the training of YOLOv8s and YOLOv5n over 40 epochs on augmented data.

Table 2. Results of training various size networks on datasets with and without augmentation

	Image size	Aug.	P	R	mAP50-95: all	mAP50-95: keys	mAP50-95: pen	mAP50-95: cup	mAP50-95: glass	FPS
YOLOv5n	320px	No	0.959	0.928	0.703	0.684	0.572	0.766	0.791	434.78
YOLOv5s	320px	No	0.94	0.928	0.71	0.717	0.54	0.773	0.808	260.55
YOLOv8n	320px	No	0.942	0.919	0.702	0.682	0.563	0.766	0.798	454.55
YOLOv8s	320px	No	0.936	0.934	0.723	0.725	0.576	0.795	0.795	333.88
YOLOv5n	320px	Yes	0.952	0.945	0.739	0.712	0.654	0.789	0.802	370.37
YOLOv5s	320px	Yes	0.941	0.947	0.738	0.736	0.619	0.788	0.81	336.63
YOLOv8n	320px	Yes	0.953	0.93	0.741	0.734	0.631	0.791	0.807	370.37
YOLOv8s	320px	Yes	0.945	0.949	0.748	0.751	0.633	0.8	0.809	419.90
YOLOv5n (40 ep.)	320px	Yes	0.954	0.944	0.759	0.753	0.661	0.8	0.82	434.78
YOLOv5n (40 ep.)	640px	Yes	0.954	0.954	0.775	0.782	0.666	0.825	0.83	487.8
YOLOv8s (40 ep.)	640px	Yes	0.951	0.953	0.77	0.783	0.635	0.824	0.838	135.14

YOLOv5n was particularly fast to train due to its small size and the dimensions of the training images (320x320px). The model achieved comparably high FPS on the validation set. Even when the size of the training images was increased to 640px, the training time was still about five times shorter than for YOLOv8s. This is an advantage in the future, as it could allow for faster parameter and dataset optimization.

Fig. 2. Predictions on images from outside the dataset, the confidence of the object detections were pen 0.9, cup 0.88 and keys 0.92.

Considering the high FPS achieved with YOLOv5n (trained on 640x640px images over 40 epochs), the short training time and the improved mAP across classes, it is the object detection model we have chosen for this project. Fig. 2 shows exemplary predictions by YOLOv5n on images from outside of our dataset, demonstrating the model's ability to generalize.

4 Discussion

4.1 Discussion on user flow and design elements of the UI

The user flow shown in Fig. 1 shows the steps the user has to take to perform a task with the robot. The design of the interface was chosen to present buttons and the video image as large as possible to improve the usability for paralyzed people and to ensure the usability with other input methods such as eye tracking or BCIs. A reason is the low accuracy of head-mounted eye tracking devices. In BCIs flickering buttons are sometimes presented at different frequencies [24]. These differences are visible in the recorded brain activity data and a selection of the button can be made. As the eye is also constantly moving, large buttons provide a larger field in which the user could focus. For this reason, the directional control buttons were placed within the video frame.

The overlap of bounding boxes and arrows would create a new challenge. The risk of pressing the wrong button, known as the Midas Touch Problem in eye tracking and gaze recognition, has to be eliminated. As a precaution, the clickable bounding boxes are only displayed if the object is a certain distance from the arrows. This also leads to a better view of the object, which is helpful for trajectory planning. Before completing the task, the user is also prompted to confirm their selection. To avoid annoying behavior by appearing and disappearing bounding boxes, the detection threshold of the object can be manipulated within a certain range. This results in either faster detection of the objects which can be selected by the user or less flickering, which can reduce seizure triggers.

Finally, the approach only allows the user to click on one bounding box and execute a task with it. Complex tasks such as pouring a glass of water could be represented by selecting multiple bounding boxes e.g., glass and bottle. However, in the real world, the glass and the bottle would be seldom in the picture at the same time. A pouring task is represented as grasping the bottle, remembering that a bottle was grasped and as soon as the user selects the cup after the user moves it into the field of view, the pouring task is proposed.

4.2 Discussion on the evaluation and results of the object detection algorithm

The object detection model was able to achieve satisfactory results even after initial difficulties with, for example, the "pen" class. The object itself has a more challenging shape to detect in comparison to cups, for example. In addition to this, training data on pens was difficult to come by.

More generally, it was interesting to note that an increase in network size from "nano" to "small" did not always translate to an increase in performance. Some possible reasons for this are that the data was limited, the dataset not as diverse as would be ideal, and as the complexity of the model increased with size. It may not be large and diverse enough to support the model, potentially leading to the drop in performance. In addition, the bounding boxes were not all manually verified in each image. Some bounding box errors were corrected, such as the bounding boxes not fitting tightly enough around each object, leading to an increase in

performance, but if any noisy or incorrect labels remained, a larger model would tend to amplify the impact of the errors, leading to a decrease in performance.

The main bottleneck to improve the performance of the object detection model would be the quality of the dataset, which could be solved in the future by creating a new dataset using frames from a stream of robotic arm in motion as this would lead to more domain-aware data than the available datasets, taking into account realistic angles the field of view could have from the perspective of a camera mounted on a robotic arm. It would also be possible to integrate objects such as specific types of cups commonly used in assistive care.

It is planned to combine the object detection network with the DeepSORT tracking algorithm. This type of algorithm would enable stable tracking of objects, even in situations where there is temporary partial or total occlusion of an object. When the object exits the field of view, its existence would not be instantly forgotten in a case where there are multiple objects of the same class available, for example. Solving these challenges will provide new insight in the design of AI-driven ARA user interfaces.

5 Conclusion

In this work we presented a user flow adapted for the use by physically impaired people. By automating task execution with a robot, it is assumed that the cognitive load of the user can be reduced. This shared control was realized by using object detection and linking tasks to the object classes. The best model was determined by evaluating FPS and mAP of different variations of YOLOv5 and YOLOv8. After training the models with data from various databases, YOLOv5n showed the best performance for this application. Part of future work is the development of an extended number of available tasks. Further, the usability will be tested with users and feedback will be collected.

6 Acknowledgement

References

1. Dragomir, A., Pana, C.F., Cojocaru, D., Manga, L.F.: Human-Machine Interface for Controlling a Light Robotic Arm by Persons with Special Needs, doi: 10.1109/ICCC51557.2021.9454664, 2021.
2. Destatis 2022. O. V., Press release No. 259 of 22 June 2022: 7.8 million severely disabled people living in Germany, Statistisches Bundesamt (Hrsg.), published 22.06.2022. URL: https://www.destatis.de/DE/Presse/Pressemitteilungen/2022/06/PE22_259_227.html, last accessed on 22.09.2023.
3. Brunete, A., Gamabo, E., Hernando, M., and Cedazo, R.: Smart Assistive Architecture for the Integration of IoT Devices, Robotic Systems, and Multimodal Interfaces in Healthcare Environment, Sensors 21 (6), DOI: 10.3390/s21062212, 2021.
4. Fischer-Janzen, A.: Eye Tracking in Robot Control Tasks, URLS: https://github.com/AnkeLinus/EyeTrackingInRobotControlTasks.git, https://doi.org/10.5281/zenodo.10003313, published: 13.10.2023, last accessed on 13.10.2023.
5. Story, M.F.: Maximizing Usability: The Principles of Universal Design, Assistive technology, 10, (1), DOI: 10.1080/10400435.1998.10131955, 1998.

6. Brohan, A., Brown, N., Carbajal, J., et al.: RT-1: Robotics Transformer for Real-World Control at Scale. arXiv, 13. December 2022. https://doi.org/ 10.48550/arXiv.2212.06817, 2022.

7. Driess, D., Xia, F., Sajjadi, M.S.M., Lynch C. et al. "PaLM-E: An Embodied Multimodal Language Model". arXiv, 6 March 2023. http://arxiv.org/abs/2303.03378.

8. Vemprala, S., Bonatti, R., Bucker, A., and Kapoor, A.. "ChatGPT for Robotics: Design Principles and Model Abilities", o. J.

9. Thesmann, S.: Interface Design, Ed. 2, Springer Vieweg, Wiesbaden, ISBN: 365803856X, 2016.

10. Srivastava, S., Divekar, A.V., Anilkumar, C., naik, I., Kulkarni, V. and Pattabiraman V.: "Comparative analysis of deep learning image detection algorithms". J Big Data 8, 66 (2021). https://doi.org/10.1186/s40537-021-00434-w.

11. Ivorra, E., Ortega, M., Catalán, J.M., Ezquerro, S., Lledó, L.D., Garcia-Aracil, N., and Alcañiz, M.: "Intelligent Multimodal Framework for Human Assistive Robotics Based on Computer Vision Algorithms", 2018, DOI: 10.3390/s18082408.

12. Li, C., Li, L, Geng, Y., Jiang, H., Cheng, M., Zhang, B., et al.: "YOLOv6 v3.0: A Full-Scale Reloading". arXiv, 13 January 2023. https://arxiv.org/pdf/2301.05586.pdf.

13. Jocher, G.: YOLOv5 by Ultralytics, version 7.0, available at: https://github.com/ultralytics/yolov5, DOI: 10.5281/zenodo.3908559, last accessed on 06.07.2023, 2020.

14. Ultralytics: YOLOv8, available at: https://github.com/ultralytics/ultralytics, last accessed on 06.07.2023.

15. Tazas y vasos Dataset, Vision, Open Source Dataset, available at https://universe.roboflow.com/ vision-gxbqw/tazas-y-vasos, Roboflow Universe, Roboflow, 2022, last accessed on 2023-07-04.

16. Emad, S.: Keysdetection Dataset, Open Source Dataset, available at https://universe.roboflow.com/ shehab-emad-n2q9i/keysdetection/dataset/2, Roboflow Universe, Roboflow, 2023, last accessed on 2023-07-12.

17. Menshikov, S.: Pen Dataset, Open Source Dataset, available at https://universe.roboflow.com/ sergey-menshikov/pen-suawu, Roboflow Universe, Roboflow, 2022, last accessed on 2023-07-12.

18. pen-ufs_dataset, Open Source Dataset, available at https://universe.roboflow.com/pen-rgrwx/pen-ufs, Roboflow Universe, Roboflow, 2022, last accessed on 2023-10-11.

19. Dwi: pen-ctcrc_dataset, Open Source Dataset, available at https://universe.roboflow.com/dwi/pen-ctcrc, Roboflow Universe, Roboflow, 2022, last accessed on 2023-10-11.

20. taitora8002@yahoo.co.jp: YOLO_PEN Dataset, available at https://universe.roboflow.com/ taitora8002-yahoo-co-jp/yolo_pen/dataset/1, Roboflow Universe, Roboflow, 2022, last accessed on 2023-10-11.

21. Ponomarjova, Katrin-Misel: HIRAC-dataset, URL: https://github.com/katrinmisel/HIRAC.git, published: 14.10.2023, last accessed on 14.10.2023.

22. Proposal for a Regulation of the European Parliament and of the Council, Laying Down Harmonized Rules on Artificial (Artificial Intelligence Act) and Amending Certain Union Legislative Acts, available: https://www.ce-richtlinien.eu/richtlinien/KI/Richtlinie/1Proposal-KI.pdf, last accessed on 09.10.2023

23. Rehadat, ICF, Aktivitäten und Partizipation, https://www.rehadat-icf.de/de/klassifikation/ aktivitaeten-partizipation/, last accessed on 22.09.2023.

24. Clark, A., and Ahmad, I.: Interfacing with Robots without the use of Touch and Speech in PETRA 2021, June 29-July 2, 2021, Corfu, Greece, DOI: 10.1145/3453892.3461330, 2021.

Design of a Robotic Therapy Companion from a Value-Sensitive and Technical Perspective

Vivienne Jia Zhong, Phillip Gachnang, Theresa Schmiedel

School of Business, FHNW University of Applied Sciences and Arts Northwestern Switzerland
viviennejia.zhong@fhnw.ch,
phillip.gachnang@fhnw.ch,
theresa.schmiedel@fhnw.ch

Abstract. The nursing sector is facing a profound shortage of skilled workers. Social robots have the potential to relieve the nursing staff. However, given that social interaction is indispensable in the nursing sector, it is essential to equip social robots with comprehensive artificial social interaction capabilities to support the staff effectively. Human values are an essential building block of social interactions and drive individual's behaviors. Therefore, human values need to be reflected in human-robot interaction. In this conceptual paper, we present a value-sensitive approach to the use case of a humanoid social robot in elderly care and conceptually outline how we intend to build to a cognitive system using control theory to enable the humanoid social robot exhibiting value-sensitive actions.

Keywords: Humanoid Robots; Social Robots; Human-Robot-Interaction; Elderly Care; Value Sensitive Design; Use Case; Control Theory; Cognitive System.

1 Relevance of Value-Sensitive Robot Design in the Care Sector

According to the OECD, long-term care is facing a profound shortage of skilled workers worldwide [1], which might lead to unsatisfying quality for care receivers. Social robots have the potential to address this issue by autonomously taking repetitive, non-nursing tasks off the hands of the often-overworked nurses.

However, for a successful deployment in elderly care, it is crucial to design social robots as a participatory and socially interactive entity in everyday life [2]. Such design requires the necessary technical functions to account for the social interaction capabilities that enable the robot to relieve overworked professionals in care institutions effectively.

Human values are an essential building block of social interactions and should, thus, be considered in robot design. However, despite their relevance, values have only received little attention in the robotics industry [3]. Value sensitive design (VSD) is an approach that considers human values in the design of technology and has increasingly raised robot researcher's interest in recent years [4].

In our research, we design a humanoid social robot to be able to function as a therapy companion. The remainder of this conceptual paper is structured as follows: Section 2 provides an overview of robots in the care context. Section 3 introduces the VSD approach, which guides our research. Section 4 outlines our research approach, and in Section 5, we envision a cognitive system with values in mind as a potential technical solution. Finally, we discuss our research in Section 6 before concluding our work in Section 7.

2 Social Robots in the Care Sector

Robots represent systems that have the "ability to perform intended tasks based on current state and sensing, without human intervention" [5]. They are increasingly used for applications in the healthcare sector. As a specific type of robots, social robots are able to communicate socially and interact with humans and other robots. These robots come in various appearances, with the ability to recognize, react, and activate certain emotions of human users based on artificial intelligence (AI) [6].

In light of the ongoing global shortage of skilled care workers, researchers have started investigating the potential of social robots in the nursing sector. The application contexts of social robots are versatile. Research has examined the use of social robots to improve elderly people's psychological well-being [7] and cognitive functions [8]. Further, social robots can provide companionship for lonely and socially isolated elderly people [9] and offer valuable services such as reminding of medication to support older adults' independent living [10]. While a large body of research focuses on zoomorphic robots with animal-like shapes, such as the robotic baby harp seal Paro, which appears to have a positive impact on the quality of life of the elderly [11], the use of humanoid robots has also shown benefits for older adults and caregivers in elderly care [12].

However, it is important to note that the breadth of social robotics research underscores that the mere presence of functionalities is not sufficient for users to adopt social robots in the long run, and therefore, other social factors need to be considered in the design of social robots [13]. In particular, researchers have increasingly recognized the mutual shaping relationship between technology and society [14]. While scholars have begun explicitly incorporating values into social robot design [15], more research that demonstrates a concrete approach to the design and development of value-sensitive social robots is still needed.

3 Value Sensitive Design

To account for the mutual shaping relationship between technology and society, value sensitive design adopts an interactional stance on technology and values [4]. Values are subconscious needs, that members of a group share (e.g., autonomy, security, or collaboration) [16], and which influences their behavior (e.g., intending to use certain technologies or not) [17]. Indeed, research has indicated that a misalignment between social robots and values might present a barrier to their adoption [18–20].

When designing values in technology, a core question is whose values should be considered. Compared to other human-centered methods, an advantage of VSD is that it extends the focus from the values of mere end-users to those of stakeholders that are affected by the use of a technology [21]. Additionally, as values are not isolated but rather interrelated, VSD recognizes the emergence of value tensions that can occur on various levels: within an individual, within a stakeholder group, among different stakeholder groups. VSD stresses that even though values might be in opposition, at the same time this tension can be solved by balancing values in relation to each other through identifying design requirements that help solve the tension [4].

To ease researchers to investigate and incorporate stakeholder values in the technology, VSD proposes a tripartite methodology that compasses three different types of investigation: Conceptual, empirical, and technical investigations [4]. While conceptual investigation explores the research object from various standpoints (e.g., analytic and theoretical) — particularly identifying stakeholders and their values, the context in which the technology is deployed is examined in empirical investigation. This investigation typically addresses how stakeholders

experience values in a specific context by applying methods from social science research. Finally, in technical investigation, researchers examine the alignment between the values and an existing technology design. Moreover, technical investigation also concerns the development of technology according to the identified values [4].

4 Approach to Value-Sensitive Social Robot Design

4.1 Use Case of a Therapy Companion

Motivated by the vision of freeing up nurses for their core nursing tasks, we study the use case of a therapy companion, which serves the purpose of reminding residents of their therapy appointments and eventually accompanying them. This use case encompasses several interaction scenarios, such as searching residents within the nursing home, engaging in conversations with residents to remind them of their appointments and to motivate them for their therapy if necessary, and accompanying residents to the therapy room, etc. Realizing this use case implies two main challenges, namely understanding which values the social robot should act upon and how to realize this technically.

The first challenge concerns designing values for the social robot. So far, existing works have mainly focused on the evaluation of social robots according to a set of predefined values. For instance, there are models assessing the value of trust on user acceptance of social robots [22]. Furthermore, frameworks are proposed to examine how predetermined moral elements are manifested in robot-supported care practices [23]. Although these post-evaluations advance the understanding of the role of value in user adoption of social robots, the design of social robots that incorporates stakeholder values, particularly the operationalization of values into the various interaction features of the social robot, is under-researched.

The second challenge concerns the technology. Thanks to the extensive research on individual AI systems such as navigation, orientation, recognition, and conversation exist, all these AI systems have achieved substantial improvements within the last decade and have been successfully applied individually. However, the technical challenge of integrating all the individual AI technologies into a complex cognitive system to achieve a common goal with values in mind remains unsolved.

4.2 Research Approach

In our research, we employ VSD to design values in the social companion robot. VSD encourages the application of the three investigation types (conceptual, empirical and technical) in an iterative and integrative way. To facilitate this, we adopt the wave approach proposed in [21], which provides specific recommendations for iterating these three investigation types in an integrative manner throughout our design process.

We start with a context analysis to investigate the current practice of therapy companion, which enables us to identify several stakeholders for this use case. Among them, we classify residents, care givers and therapists as direct stakeholders and resident's relative and the management of the nursing home as indirect stakeholders. Following this, we conceptually examine stakeholder-relevant values through existing research (e.g., [24]). The results of the conceptual analysis are then leveraged in our empirical investigation through various qualitative methods. For instance, in value-oriented interviews [4], we ask stakeholders to explicitly assess values identified from the conceptual investigation with regard to the use case, while also implicitly eliciting values from their conception of the robot-supported therapy companion.

For the operationalization of values, we utilize the value hierarchy [25] to translate identified values into design requirements for the use case. Additionally, we collect design requirements from stakeholders through empirical methods (e.g., workshops). When new values emerge from the empirically collected design requirements, we revisit the value hierarchy [25] to derive design requirements for these newly identified values. To guide the technical development, we establish an understanding of the various interaction scenarios between the robot and residents. We suggest using design patterns [26] to create these interaction scenarios and specify human-robot interaction (HRI) based on the derived design requirements. Design patterns have been used in previous works to enable effective and pleasant HRI [27, 28].

Implementing a technical artifact based on the derived design requirements and HRI specifications means creating a cognitive architecture that orchestrates all the individual AI systems necessary for this use case. This alone is a complex endeavor. Enabling the architecture for value sensitivity further raises the bar of complexity. To tackle this challenge, we envision a cognitive interaction system utilizing control theory.

5 Control Theory as Concept of a Cognitive Interaction System

Values are mainly visible through the actions a robot takes while interacting with a human. Therefore, we focus on HRI and see everything the robot does as an action, such as speaking kindly, moving side by side with people, showing empathic facial expressions, etc. and the humans' responses as reaction.

In order to translate value-sensitive behavior to a technical level, we apply control theory, because a system that reacts on actions can be modeled in control theory to control the system's state. Control theory has the objective to develop a model governing a system input to drive the system to a desired state. Such a system can be described in the following Eq. 1:

$$A(y) = f(v) \tag{1}$$

Whereby y represents the unknowable state of the system we are willing to control. On the other hand, $v \in U_{ad}$ is the control, the variable we can choose to act and change the state. If $A: D(A) \subset Y \longmapsto Y$ and $f: U_{ad} \longmapsto Y$ are two linear or nonlinear functions where the operator A determines the equation that must be satisfied by the state y, D the set of decision, Y the solution space and all possible states, and U_{ad} the set of all controls. Therefore, the function f indicates how the control v effects the state of the system [29].

The main goal of control theory is to model the system to be controlled by finding the controls v, that lead to an associated state $y(v)$. Such a model allows to build a control system managing, manipulating, or directing the state of the system to be controlled into a desired state. There are two fundamental types of control systems: the open-loop control system and the closed-loop control system. We focus on the closed-loop control because it includes the feedback concept that is essential in HRI to reflect which values to emphasize in the robot's behavior based on the humans' reaction to the robot. Therefore, the state of a system is fed back to the control system, which continuously determines the control v based on the feedback of the system to affect the state y towards a desired state.

Such a control system could also be applied to HRI, if we see the *interaction* as the system to be controlled, the *set of values to be respected* as the state of the system and the *robot's actions* as control variables. Important is that human (re)action also influences the state of the interaction in regard to the respected values and therefore, creates a new state within the interaction system.

The observation of the human's (re)action using sensors builds the feedback as depicted in Fig. 1. In such an interaction system, the robot's behavior or actions are decided by the control system based on the current state, represented by the respected values by the human's reaction, with the goal to reach a new desired state of respected values. Modern control systems are carried out in the state space and can deal with multiple-input and multiple-output (MIMO) systems, which is important in the application of control theory in HRI and control multiple values by multiple actions. Therefore, actions may affect several respected values. With state spaces, a robot is able to respect multiple values with tensions in the action to take and read multiple values out of the humans reaction to find a new desired state of the interaction to choose the next action.

Imagine an interaction between a robot and a human, where the robot is introduced to the human. The robot first approaches the unknown human with low emphasis on the value of security but high emphasis on the value of openness. However, the camera notices that the human backs away and the control system adjusts the desired values of security and openness for this interaction, interpreting the backing away as an indication for unsecure feelings of the human and, thus, as a need to increase attention to the value of security. Instead of forcing a certain distance to the human, the robot actions are adjusted to the human's reaction and the robot stops to keep a secure distance and emphasizes the value of security more than before while keeping the value of openness as high as before in the actions.
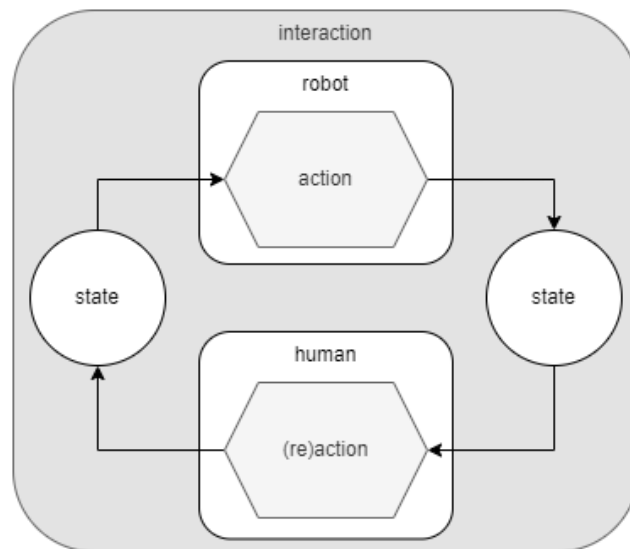
Fig. 1. Interaction feedback loop

In the context of HRI, it is important to have room for fluctuations of the values within the interaction system to appropriately represent a desired value set in the states. Forcing the state of the interaction into certain set of respected values can lead to negative or opposite effects. A robot cannot just act happy to force happiness into an interaction with an upset human. The robot should rather try to express respect for values in small steps to incrementally lead the interaction into a desired set of respected values and avoid any enforcement of values. Fernandez-Cara and Zuazua [29] stating in their summary to control theory the following:

> *"To control a system arising in Nature or Technology, we do not have*
> *necessarily to stress the system and drive it to the desired state*
> *immediately and directly. Very often, it is much more efficient to control*
> *the system letting it fluctuate, trying to find a harmonic dynamics that*
> *will drive the system to the desired state without forcing it too much."*

We propose to design a cognitive system with values in mind as a decision-making system (DMS) based on control theory and control systems. Perception can be defined as the human fundamental form of cognitive contact to the world, while it refers in robotics to the ability to perceive the environment [30]. Therefore, the robot can estimate on the sentiment of (re)action of the human how the robot's action affected the system. Cognition is defined as the ability of humans to know, learn, and understand things. Hence, designing cognitive systems implies making them capable of reasoning about their actions [30]. Therefore, the robot must choose the next action accordingly to get closer to the desired state, the prescribed set of values to respect.

6 Discussion

In the present work, we highlighted the need for considering human values in social robot design and outlined our approach to incorporating values in a social robot application in a nursing home. Finally, we present a novel cognitive interaction system based on VSD and control theory.

In particular, we employ VSD to account for human values. A major differentiation of our research from other VSD robotics work (e.g., [15]) is that we strongly strive for an integrative approach as recommended by VSD. We deliberately iterate the value analysis through conceptual and empirical investigations because it is particularly important to understand how the conceptually identified values are manifested in the usage scenario and allows stakeholders to express context-specific values. The same applies to the derivation of design requirements. Through conceptually deriving requirements by using the value hierarchy [25] and empirically with stakeholders, we can triangulate these derived design requirements, thereby creating a design that better reflects the stakeholder values. Furthermore, while previous works only derive design requirements but do not specify how to design HRI based on these requirements [15, 23], our research proposes the use of design patterns, which facilitate the specification of HRI across various robotics platforms and ease the technical development.

Control theory has primarily found applications in physical human robot interaction [30–33]. In contrast, our research explores the possibilities to apply this theory in social interactions by defining the interaction as the system to be controlled and the respected values within the interaction as the states. Notably, the application of control theory and the implementation of a control system in social HRI comes with its own challenges. The value preferences underlying the state of an HRI, must be prescribed and measurable. Sentiment analysis of the human's behavior (speech, mimics, gestures) might allow for a measurement, but concluding from human behavior to a desired set of values including their appropriate emphasis is a difficult endeavor. While subtle emotional human twists may probably be impossible to detect by the robot in the first place, we intend that the robot is able to distinguish states that indicate, for example, emergency situations, boredom, fear, and satisfaction to appropriately react to these. We assume we can address this challenge with the application of fuzzy logic, but further research must be carried out to validate this assumption.

7 Conclusion

In light of the pressing issues presented in long-term care, social robots have the potential to support the care givers in their daily tasks and enhance the quality of life for care receivers. Given the fundamental role of values in user acceptance of robots, we urge all researchers and practitioners to incorporate values in robot design as a whole. The presented conceptual paper gives an insight into our research approach to value-sensitive social robots and outlines how to

technically tackle the value integration based on control theory. As part of our future work, we aim to validate our approach to operationalizing values in HRI and refine the proposed technical concept.

References

1. OECD: Who Cares? Attracting and Retaining Care Workers for the Elderly, https://www.oecd-ilibrary.org/content/publication/92c0ef68-en, (2020).
2. Breazeal, C.: Socially intelligent robots. Interactions. 12, 19–22 (2005). https://doi.org/10.1145/1052438.1052455.
3. Cheon, E., Su, N.M.: Integrating Roboticist Values into a Value Sensitive Design Framework for Humanoid Robots. In: 2016 11th ACM/IEEE International Conference on Human-Robot Interaction (HRI). pp. 375–382 (2016). https://doi.org/10.1109/HRI.2016.7451775.
4. Friedman, B., Hendry, D.G.: Value Sensitive Design: Shaping Technology with Moral Imagination, (2019).
5. International Organization for Standardization: ISO 8373:2021, Robotics – Vocabulary, https://www.iso.org/obp/ui/en/#iso:std:iso:8373:ed-3:v1:en.
6. Sarrica, M., Brondi, S., Fortunati, L.: How many facets does a "social robot" have? A review of scientific and popular definitions online. Inf. Technol. People. 33, 1–21 (2019). https://doi.org/10.1108/ITP-04-2018-0203.
7. Moyle, W., Jones, C.J., Murfield, J.E., Thalib, L., Beattie, E.R., Shum, D.K., O'Dwyer, S.T., Mervin, M.C., Draper, B.M.: Use of a robotic seal as a therapeutic tool to improve dementia symptoms: a cluster-randomized controlled trial. J. Am. Med. Dir. Assoc. 18, 766–773 (2017).
8. Kim, G.H., Jeon, S., Im, K., Kwon, H., Lee, B.H., Kim, G.Y., Jeong, H., Han, N.E., Seo, S.W., Cho, H.: Structural brain changes after traditional and robot-assisted multi-domain cognitive training in community-dwelling healthy elderly. PloS One. 10, e0123251 (2015).
9. Banks, M.R., Willoughby, L.M., Banks, W.A.: Animal-assisted therapy and loneliness in nursing homes: use of robotic versus living dogs. J. Am. Med. Dir. Assoc. 9, 173–177 (2008).
10. Jayawardena, C., Kuo, I.-H., Broadbent, E., MacDonald, B.A.: Socially Assistive Robot HealthBot: Design, Implementation, and Field Trials. IEEE Syst. J. 10, 1056–1067 (2016). https://doi.org/10.1109/JSYST.2014.2337882.
11. Wang, X., Shen, J., Chen, Q.: How PARO can help older people in elderly care facilities: A systematic review of RCT. Int. J. Nurs. Knowl. 33, 29–39 (2022). https://doi.org/10.1111/2047-3095.12327.
12. Persson, M., Redmalm, D., Iversen, C.: Caregivers' use of robots and their effect on work environment - a scoping review. J. Technol. Hum. Serv. 40, 251–277 (2022). https://doi.org/10.1080/15228835.2021.2000554.
13. de Graaf, M.M.A., Ben Allouch, S., van Dijk, J.: Why Do They Refuse to Use My Robot? Reasons for Non-Use Derived from a Long-Term Home Study. In: Proceedings of the 2017 ACM/IEEE International Conference on Human-Robot Interaction. pp. 224–233. Association for Computing Machinery, Vienna, Austria (2017). https://doi.org/10.1145/2909824.3020236.
14. Šabanović, S.: Robots in Society, Society in Robots. Int. J. Soc. Robot. 2, 439–450 (2010). https://doi.org/10.1007/s12369-010-0066-7.
15. Umbrello, S., Capasso, M., Balistreri, M., Pirni, A., Merenda, F.: Value Sensitive Design to Achieve the UN SDGs with AI: A Case of Elderly Care Robots. Minds Mach. 31, 395–419 (2021). https://doi.org/10.1007/s11023-021-09561-y.

16. Ros, M., Schwartz, S.H., Shoshana, S.: Basic Individual Values, Work Values, and the Meaning of Work. Appl. Psychol. 48, 49–71 (1999).
17. Maio, G.R.: The Psychology of Human Values. Routledge, London (2016).
18. Persson, M., Redmalm, D., Iversen, C.: Caregivers' use of robots and their effect on work environment - a scoping review. J. Technol. Hum. Serv. 40, 251–277 (2022). https://doi.org/10.1080/15228835.2021.2000554.
19. Turja, T., Aaltonen, I., Taipale, S., Oksanen, A.: Robot acceptance model for care (RAM-care): A principled approach to the intention to use care robots. Inf. Manage. 57, 103220 (2020). https://doi.org/10.1016/j.im.2019.103220.
20. Carros, F., Meurer, J., Löffler, D., Unbehaun, D., Matthies, S., Koch, I., Wieching, R., Randall, D., Hassenzahl, M., Wulf, V.: Exploring Human-Robot Interaction with the Elderly: Results from a Ten-Week Case Study in a Care Home. In: Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems. pp. 1–12. ACM, Honolulu HI USA (2020). https://doi.org/10.1145/3313831.3376402.
21. Schmiedel, T., Zhong, V.J., Eyssel, F.: Towards a Wave Approach for Value Sensitive Design in Social Robotics. In: Companion of the 2023 ACM/IEEE International Conference on Human-Robot Interaction. pp. 592–596. Association for Computing Machinery, New York, NY, USA (2023). https://doi.org/10.1145/3568294.3580154.
22. Heerink, M., Kröse, B., Evers, V., Wielinga, B.: Assessing Acceptance of Assistive Social Agent Technology by Older Adults: the Almere Model. Int. J. Soc. Robot. 2, 361–375 (2010). https://doi.org/10.1007/s12369-010-0068-5.
23. van Wynsberghe, A.: Designing Robots for Care: Care Centered Value-Sensitive Design. Sci. Eng. Ethics. 19, 407–433 (2013). https://doi.org/10.1007/s11948-011-9343-6.
24. Rider, E.A., Kurtz, S., Slade, D., Longmaid, H.E., Ho, M.-J., Pun, J.K., Eggins, S., Branch, W.T.: The International Charter for Human Values in Healthcare: An interprofessional global collaboration to enhance values and communication in healthcare. Patient Educ. Couns. 96, 273–280 (2014). https://doi.org/10.1016/j.pec.2014.06.017.
25. van de Poel, I.: Translating Values into Design Requirements. In: Michelfelder, D.P., McCarthy, N., and Goldberg, D.E. (eds.) Philosophy and Engineering: Reflections on Practice, Principles and Process. pp. 253–266. Springer Netherlands, Dordrecht (2013). https://doi.org/10.1007/978-94-007-7762-0_20.
26. Kahn Jr., P.H., Freier, N.G., Kanda, T., Ishiguro, H., Ruckert, J.H., Severson, R.L., Kane, S.K.: Design Patterns for Sociality in Human-Robot Interaction. In: HRI 2008 - Proceedings of the 3rd ACM/IEEE International Conference on Human-Robot Interaction: Living with Robots. pp. 97–104 (2008). https://doi.org/10.1145/1349822.1349836.
27. Sauppé, A., Mutlu, B.: Design patterns for exploring and prototyping human-robot interactions. In: Conference on Human Factors in Computing Systems - Proceedings. pp. 1439–1448 (2014). https://doi.org/10.1145/2556288.2557057.
28. Pollmann, K., Ziegler, D.: A Pattern Approach to Comprehensible and Pleasant Human–Robot Interaction. Multimodal Technol. Interact. 5, 49 (2021). https://doi.org/10.3390/mti5090049.
29. Fernández Cara, E., Zuazua Iriondo, E.: Control theory: history, mathematical achievements and perspectives. (2003).
30. Maroto-Gómez, M., Alonso-Martín, F., Malfaz, M., Castro-González, Á., Castillo, J.C., Salichs, M.Á.: A Systematic Literature Review of Decision-Making and Control Systems for Autonomous and Social Robots. Int. J. Soc. Robot. 15, 745–789 (2023). https://doi.org/10.1007/s12369-023-00977-3.
31. Yu, H., Huang, S., Chen, G., Pan, Y., Guo, Z.: Human–Robot Interaction Control of Rehabilitation Robots With Series Elastic Actuators. IEEE Trans. Robot. 31, 1089–1100 (2015). https://doi.org/10.1109/TRO.2015.2457314.

32. Li, Y., Sena, A., Wang, Z., Xing, X., Babič, J., Asseldonk, E. van, Burdet, E.: A review on interaction control for contact robots through intent detection. Prog. Biomed. Eng. 4, 032004 (2022). https://doi.org/10.1088/2516-1091/ac8193.

33. Mukherjee, D., Gupta, K., Chang, L.H., Najjaran, H.: A Survey of Robot Learning Strategies for Human-Robot Collaboration in Industrial Settings. Robot. Comput.-Integr. Manuf. 73, 102231 (2022). https://doi.org/10.1016/j.rcim.2021.102231.

Using deep learning models to accelerate the design of soft robots with genetic algorithms

Loïc Mosser[1], Laurent Barbé[1], Lennart Rubbert[1] and Pierre Renaud[1]

ICube, Université de Strasbourg - CNRS - INSA Strasbourg, France
`mosserl@unistra.fr`

Abstract. The motion of soft robots is intrinsically linked to their shape. Design of soft robots is then still a challenge, with a very large design space to explore in terms of possible shapes. Generative methods can be of interest, but they require intensive use of robots motion prediction. We assess the interest of using deep learning models to accelerate the synthesis. The case of pneumatically-actuated structures is considered. We show first that a Resnet model can accurately describe the structure motion after learning on a dataset based on finite element simulations. Second, we show that the model accuracy can be maintained during a synthesis, outside the initial dataset, using transfer learning.

Keywords: Soft Robot, Resnet, Transfer learning, Genetic Algorithm

1 Introduction

Soft robots are structures made of flexible material, for which motion is obtained by various means such as cable or pneumatic actuation [1]. The latter is largely considered with so-called soft pneumatic actuators (SPA) [2], the elementary components of a soft robot. Pneumatic chambers are then distributed in the SPA body and they allow deformation of the structure. Design methods for these actuators still need to be developed [3, 4]. With the freedom of shape allowed by additive manufacturing techniques [5], the design space is in particular growing dramatically and efficient design methods are yet to be proposed.

The gold standard to assess a SPA motion is to use finite element analysis (FEA) [6, 7]. Designing SPA using FEA and evolutionary algorithms such as genetic algorithms could be interesting to get a generic design method [8], but the computational cost of FEA limits the feasibility. In the literature, deep learning (DL) models have been proven to be relevant for the prediction of soft structure deformation under various loadings [9–11]. During a SPA design, the problem of prediction using DL is however quite different: the model has to estimate the displacements of a structure, while the latter is modified by the design process. At the same time, the loading is also modified as it is related to the pressurization of the SPA.

In this paper, we thus assess the adequacy for such use of a CNN model. First, we describe the SPAs to be modeled. A simplified design problem is considered to study the limitations of a CNN model when the training dataset is of limited size compared to the full design space. We then study the impact of bias in the training dataset in a situation where the design space can be exhaustively explored. Finally, we assess the model capacity to remain accurate while being used in other areas of the large design space, using transfer learning (TL).

2 SPA definitions and CNN construction

2.1 SPA under consideration

Our assessment is based on the design of SPAs defined by a discrete distribution of soft material through a 25x25x75 matrix of 0.4 mm voxels, to be compatible with experimental assessment in the future. The prismatic outer shape is considered fixed, considering it is imposed by size requirements. The air supply is fixed at the center of the bottom surface. A functional SPA is composed of a hollow structure connected to the air supply within the prismatic outer shape. Examples of three-dimensional SPAs generated randomly accordingly are given in Fig. 1-a.
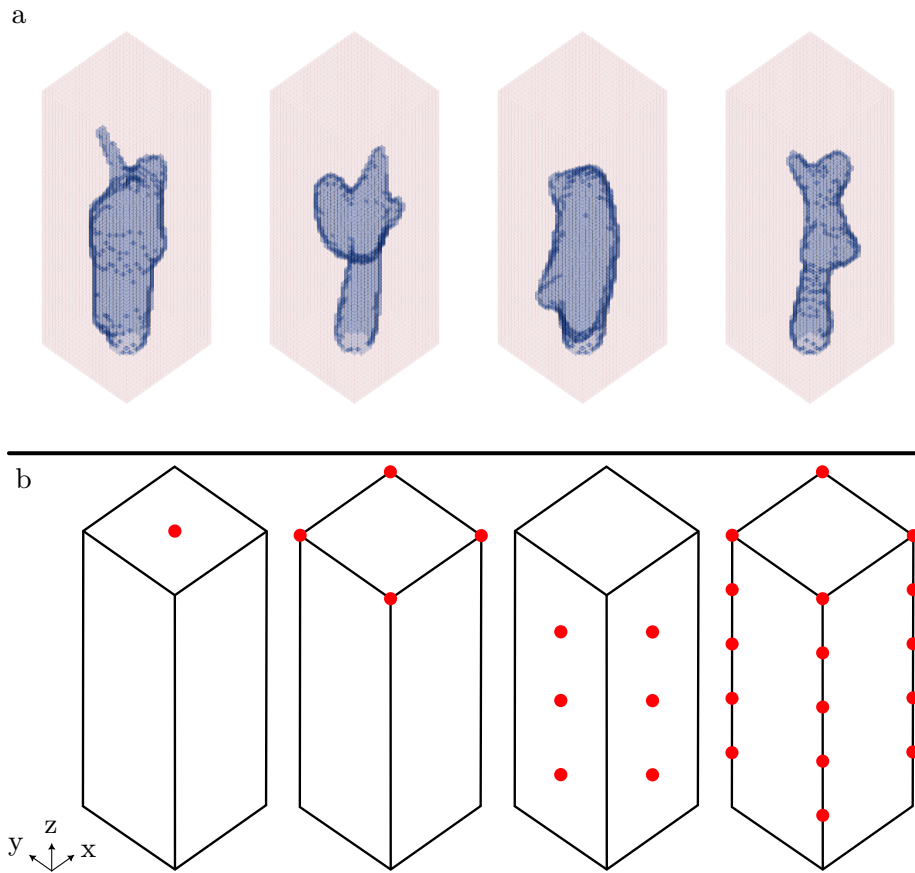
Fig. 1: In (a), examples of SPA randomly generated for the learning dataset (blue : void, red : filled), in (b) the points of interest used to define the SPA behavior with, from left to right, 1, 4, 12 and 20 points.

Our goal is to estimate the 3D displacements at n points of interest, located on the external border of the SPA. The output of the CNN model is thus a vector of size $3n$. In the following, we estimate the prediction accuracy in 4 situations, *i.e.* with n=1, 4, 12, 20 as represented in Fig. 1-b. As a reference and for CNN training, FEA is performed using Comsol. The SPA body is considered as composed of a soft linear elastic material ($E = 2$ MPa, $\nu = 0.3$).
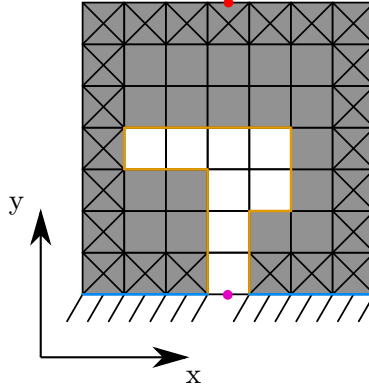
Fig. 2: Schematic representation of the reduced problem with the point whose displacement is being tracked (red point), the outer edge (crossed-out voxels), the pneumatic input (violet point), the fixed surface at the bottom of the design (blue surface) and the pneumatic chamber (yellow) with full (grey) and empty (white) voxels.

2.2 Simplified design of SPA

Implementing a deep learning model for a large design problem makes it difficult to identify the influence of biases during learning. Here, with a structure composed of $25 \times 25 \times 75$ voxels, there are 10^{14096} possible designs of SPA. Thus, a simplified problem of SPA design is considered for initial evaluation of CNN behavior. The design is 2D, and described by 5×5 matrix. SPA material is the same. There are then 1,338,341 possible SPAs, all of which can be simulated using FEA: the whole design space can therefore be explored. One example is depicted in Fig. 2.

2.3 CNN model construction

In [12], a Resnet model was successfully used to estimate the behavior of composite structures under static loading. The composite structure is then defined as an array, with 2 possible materials for each cell. The model is then employed to predict the structure mechanical behavior, e.g. displacement at a specific point. This is closely related to our problem, so we decided to assess a similar approach. The implementation and training of the Resnet is carried out using Tensorflow and Keras on the same computer.

To determine the number of parameters required to predict the mechanical behavior of SPA, the number of residual convolution layers (RCL) is increased gradually, until an RMSE of the order of 10^{-2} mm is reached. RCL are defined using full pre-activation (Fig. 5) as it has shown great generalization performances. The Resnet takes as input the matter distribution matrix defined as a matrix boolean values and outputs the $3n$ vector representing the displacements of n points defined in Fig. 1-b and expressed in millimeters.

3 Initial assessment of CNN model prediction accuracy

3.1 Creation of the dataset

For this initial assessment, the simplified design is being used. There is one point of interest ($n = 1$) on the outer edge of the SPA (Fig. 2, red point). The SPA structure is defined (Fig. 2) by a 5×5 matrix. The dataset is generated by considering 3 rules: 1) the

outer strip of material needs to remain present to keep the SPA sealed ;2) the pneumatic input is set at the center of the lower surface of the design (Fig. 2, violet point) ; 3) the lower surface is attached to the base, so it has no displacement. During the generation of designs, all voxels composing the inner structure of the SPA, that is submitted to the internal pressure, must be connected to the pneumatic supply, as shown in Fig 2. Designs that do not follow this rule are not considered.

With a reduced problem formulated in this way, we use a greedy algorithm to determine the set of possible SPAs. The dataset contains about 1.4×10^6 SPAs. From this dataset, we analyze the training results of a Resnet network. The dataset is first ranked using the value of displacement at the point of interest. This creates on purpose a bias in the training data, which impact is analysed in the following.

3.2   CNN model construction

The constructed Resnet architecture is composed of an input convolution layer (CL), 11 residual convolution layers, a last CL followed by 2 dense layers of 128 neurons each. Each CL have 64 3×3 kernels. RCL are defined using full pre-activation as it has shown great generalization performances. The Resnet takes as input the matter distribution matrix defined as a matrix of 5×5 boolean values and outputs the displacement of the monitored displacement along $\vec{y}$. In order to facilitate the representation of results, we limit ourselves to the study of vertical displacement coordinates.

An intial training of the Resnet is done using 100,000 SPA that randomly picked in the dataset. 90 % are used as a training set and 10 % as a validation set. RMSProp is being used for the training as it here provides better performances than Adam. The remaining individuals are used to test the model performance. The training is done through 300 epochs with a learning rate of 10^{-4} and early-stopping. The performance of this initial training, evaluated on the test set, is shown in Fig. 3. The results show that the Resnet is able to give an estimate of displacement with an accuracy of less than 10 µm for the entire dataset. The R^2 coefficient is then of 0.99.
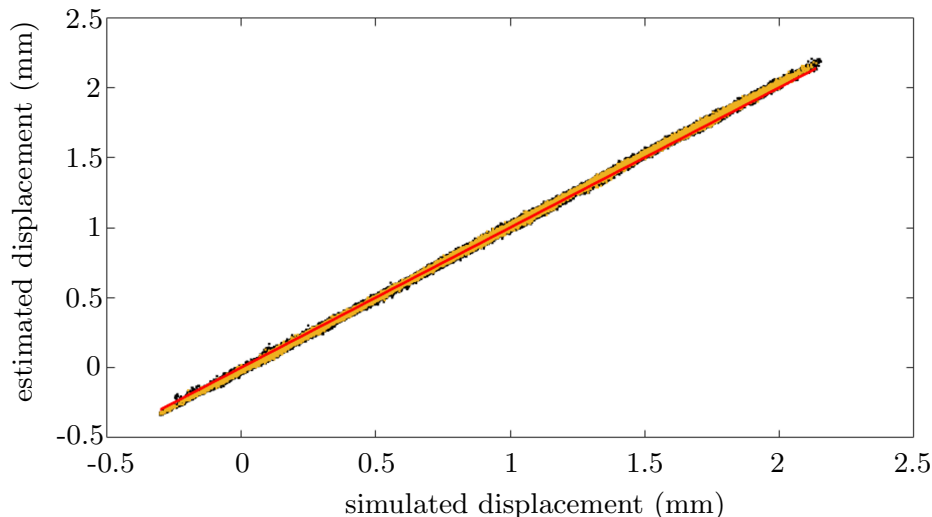
Fig. 3: Learning results on the whole dataset without the introduction of bias with training data (yellow) and test data (black).

3.3 CNN behavior

We seek to quantify the loss of performance of the network's estimation when its learning is conducted on a non-representative set of all SPAs. To this end, the complete dataset obtained after the initial sorting operation is segmented in 2 domains, using a threshold on the achievable displacement at the point of interest. The domain with the smallest displacements is used to build the training dataset and the test dataset. The other domain is used as the bias control dataset. 4 situations are considered, with 20%, 40%, 60% or 80% of designs with smallest displacements. The training dataset contains 100,000 SPAs for each assessment.

The training results are available on Fig. 4 and on Table 1.

considered displacement	training RMSE	training R^2	Test RMSE	Test R^2	bias control RMSE	bias control R^2
80% selection	$9.0 \cdot 10^{-3}$	0.999	$9.0 \cdot 10^{-3}$	0.999	$1.14 \cdot 10^{-1}$	0.350
60% selection	$8.0 \cdot 10^{-3}$	0.999	$8.0 \cdot 10^{-3}$	0.999	$2.54 \cdot 10^{-1}$	-1.745
40% selection	$7.0 \cdot 10^{-3}$	0.999	$7.0 \cdot 10^{-3}$	0.999	$3.54 \cdot 10^{-1}$	-3.687
20% selection	$6.0 \cdot 10^{-3}$	0.998	$6.0 \cdot 10^{-3}$	0.998	$8.43 \cdot 10^{-1}$	-38.752

Table 1: Presentation of training results on 100,000 data points taken from a set of chambers with displacement associated with the best 80, 60, 40 and 20 percent of the data in the dataset.

When we evaluate the network's performance on the training domain, we find that it does not differ from the initial training, for both training (Fig. 4, yellow) and test (Fig. 4, black) datasets. The CNN is then able to provide a satisfactory estimate of vertical displacement at the monitored point. The CNN performance is, however, insufficient on the bias control dataset (Fig. 4, blue). We note that this estimator loses precision when the maximum displacement of SPAs considered in the training is diminished.

The bias introduced in this learning process allows us to understand the possible issues during the design of SPAs. During the initial generation of SPAs, it is difficult to propose a dataset that offers an exhaustive representation of all possible displacements. If the initial SPA generation process is based on unguided random generation, we can anticipate that a bias will be introduced, that will impact the model built by learning. We can also anticipate that other biases, such as a bias linked to the size of the pneumatic chambers, will impact our initial dataset. So, rather than focusing on the initial generation method, in the following we investigate the efficiency of correcting this bias through learning transfer steps.

4 CNN for SPA design with transfer learning

4.1 CNN model learning

We now focus on the SPA under consideration initially, as described in 2.1. The constructed Resnet architecture (Fig. 5) is composed of an input convolution layer (CL), 16 residual convolution layers, a last CL followed by an average pooling layer and 2 dense layers of 128 neurons each. Each CL have 64 $3 \times 3 \times 3$ kernels. RCL are defined using full pre-activation (Fig. 5) as it has shown great generalization performances. The Resnet takes as input the matter distribution matrix defined as a matrix of $25 \times 25 \times 75$ boolean
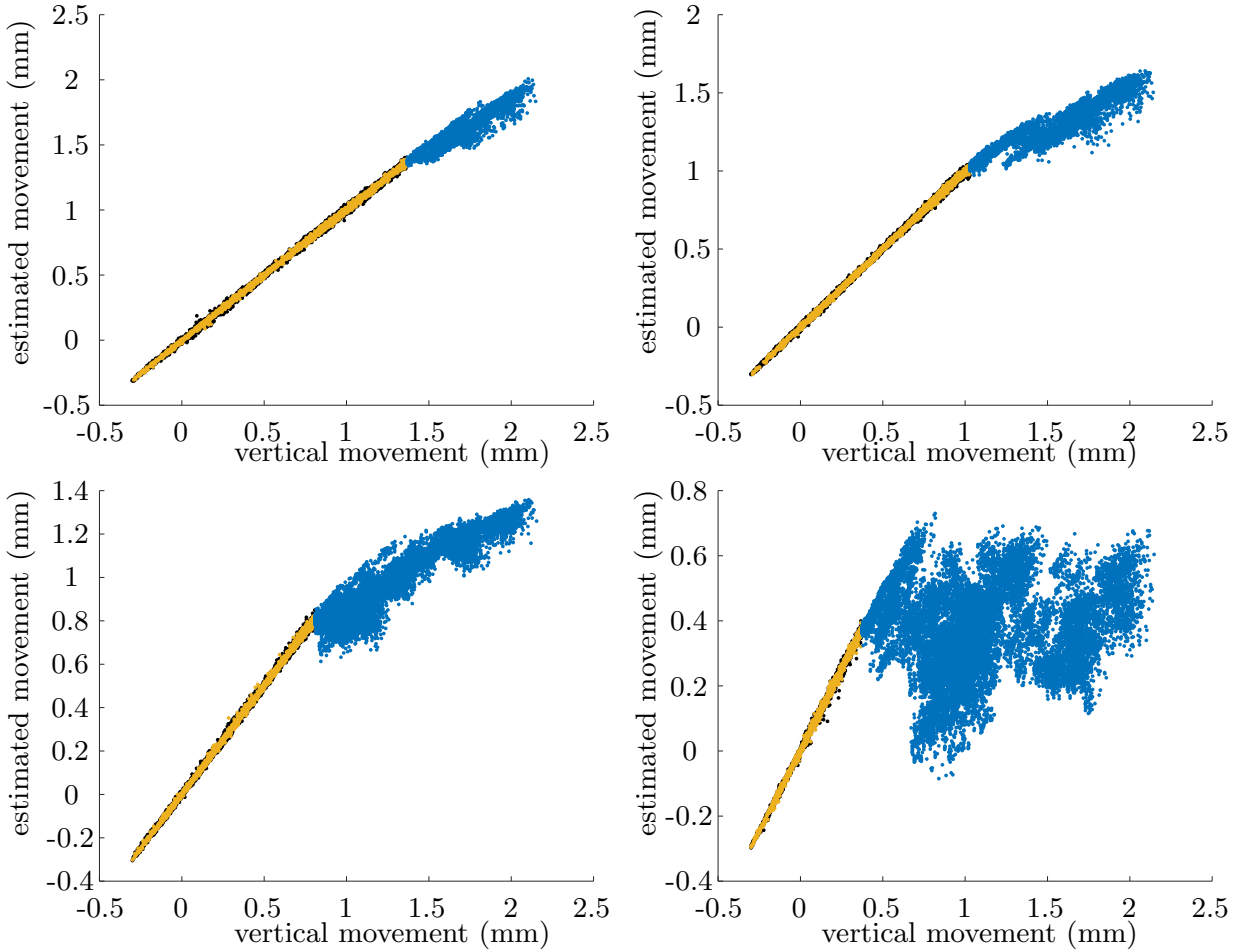
Fig. 4: Plot of estimated displacements versus simulated displacements for the different drives performed on the lowest 80% (top-left), 60% (top-right), 40% (bottom-left) and 20% (bottom-right) displacements.

values and outputs the $3n$ vector representing the displacements of n points defined in Fig. 1-b and expressed in millimeters.

For its training, an initial dataset of 100,000 designs is first randomly generated. The simulation time is about 8 days using 1 PC (Intel i9-10900KF, 64 GB of RAM, NVIDIA RTX-3090).

The training of the Resnet is done using 60,000 SPA randomly drawn in the initial dataset (90 % as training set and 10 % as validation set) using RMSProp as it here provides better performances than Adam. The remaining 40,000 SPA are used to test the model performance. The training is done through 20 epochs with a learning rate of 10^{-4} and early-stopping. The performance of this initial training, evaluated on the test set, is shown in the table 2. The results show that, with initial training, the Resnet is able to give an estimate of displacement with an accuracy of less than 10 µm for the 4 situations under consideration. In addition, one estimation of SPA performance requires a computational time of less than 1 ms, compared with 30 seconds for FEA.

4.2 CNN accuracy with transfer learning

During the SPA design, the design space is being explored to gradually tend to obtain larger displacements, meaning better performances out of the initial dataset domain.
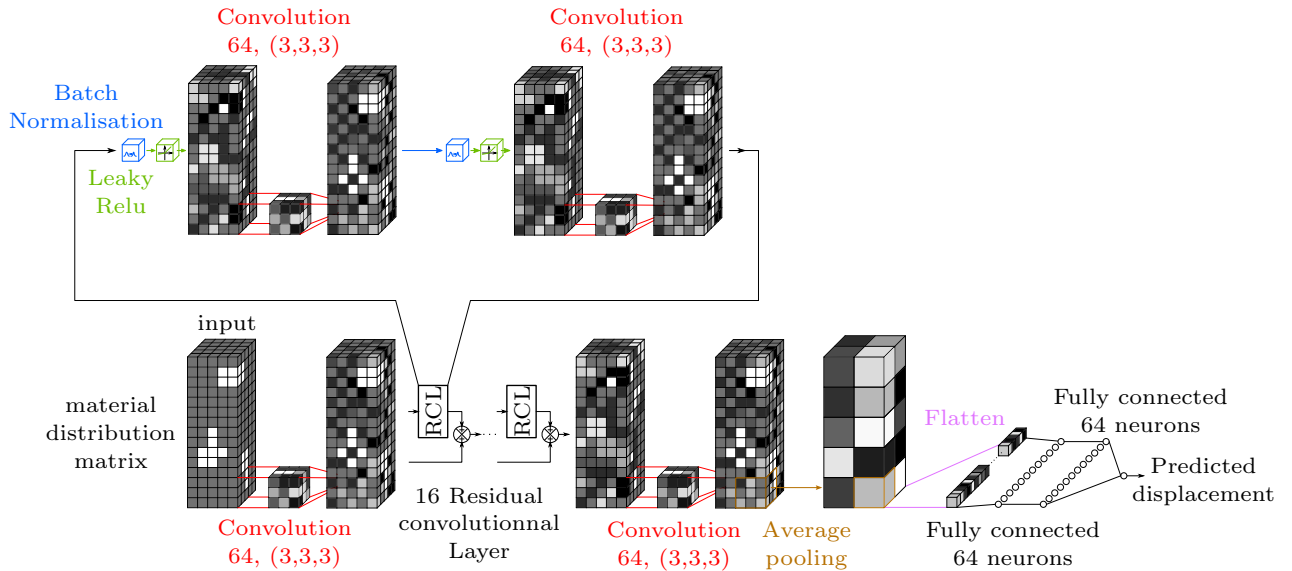
Fig. 5: Resnet representation with its input as the material distribution matrix and its output as a dense layer of $3n$ neurons.

When the Resnet is being used in a domain out of the initial dataset, prediction accuracy may decrease significantly as we showed earlier on the simplified problem. To assess this, a specific dataset of 830 SPA designs with greater displacements and far from the initial dataset has been generated specifically (Fig. 6, blue and orange dots). We define our test dataset with 800 SPA. The initial performance of the CNN is indicated in the second column of Table 2. For example, precision is almost decreased by a factor 20 when 1 point of interest is considered, which means a strong loss of accuracy.

A transfer learning (TL) step is then considered. It is achieved through 10 epochs with a learning rate of 10^{-5} with 3000 SPA. 1500 SPA are randomly drawn in the training dataset of 60,000 SPA and the 30 extra SPA geometry are duplicated 50 times to have an equal representation in the TL dataset. Performances of the test dataset are indicated in the last column of Table 2. It highlights the relevance of TL in design space exploration out of the initial domain. The loss of precision is then only by a factor 2, which could be acceptable as it is only used to identify best design subspaces.

Considered output	Metrics	Initial training	Before TL	After TL
1 point	R^2	0.99	0.90	0.96
	RMSE	$7.7 \cdot 10^{-3}$	$1.4 \cdot 10^{-1}$	$1.6 \cdot 10^{-2}$
4 points	R^2	0.99	0.92	0.94
	RMSE	$9.0 \cdot 10^{-3}$	$1.8 \cdot 10^{-1}$	$1.8 \cdot 10^{-2}$
12 points	R^2	0.99	0.69	0.92
	RMSE	$6.8 \cdot 10^{-3}$	$1.3 \cdot 10^{-1}$	$1.3 \cdot 10^{-2}$
20 points	R^2	0.98	0.81	0.89
	RMSE	$7.3 \cdot 10^{-3}$	$6.5 \cdot 10^{-2}$	$1.1 \cdot 10^{-2}$

Table 2: Resnet performance as a function of the number of points of interest, the use after initial training, without and with transfer learning.
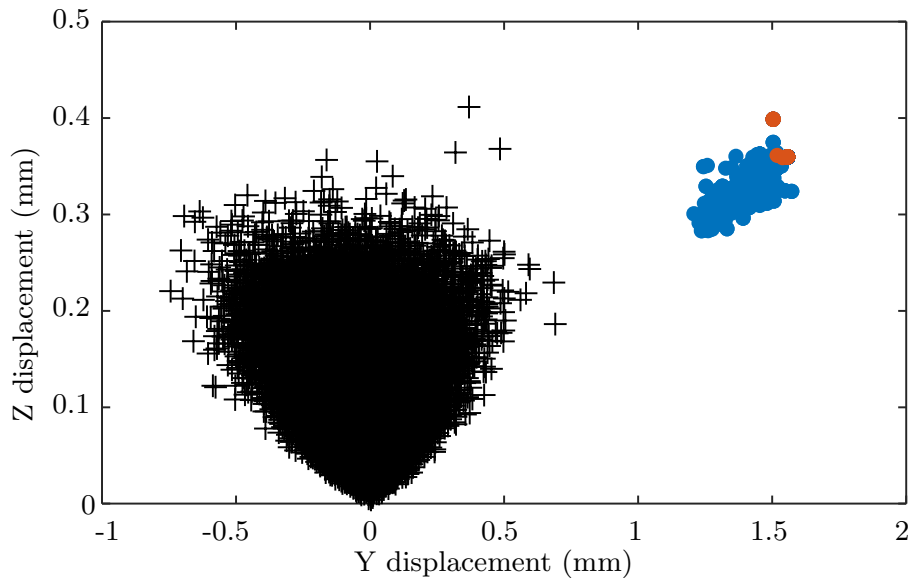
Fig. 6: Displacement of 1 point of interest of the SPA. Black/Blue/Orange crosses represent the initial dataset/the test dataset for TL/ the train dataset for TL

5 Conclusion

We investigated the use of a Resnet model to estimate the behavior of SPA during its design. First, it reduces the computation time from 30 s to less than 1 ms in comparison with FEA. The Resnet accuracy on a randomly generated population of designs is in the order of 10 μm. This performance is obtained when focusing on 1 or multiple, here up to 20, points of interest. We have identified that learning the CNN on an initial population of SPAs may be subject to certain biases. We then proposed to carry out a transfer learning step on a reduced number of new data to correct the estimate proposed by the CNN. The capacity to use transfer learning to improve the Resnet accuracy when it is used during the synthesis, far from the initial dataset, was also evaluated. The loss of precision can be reduced by a factor of 20 to 2 thanks to the TL.

The overall impact of computational time reduction obviously depends on the evolutionary process, which is another aspect to investigate. A perspective will then to include more advanced material models, to refine the design process, for instance including material non-linearities that can be present with very soft materials.

References

1. Rus, D., Tolley, M.T.: Design, fabrication and control of soft robots. Nature **521**(7553) (may 2015) 467–475
2. Walker, J., Zidek, T., Harbel, C., Yoon, S., Strickland, F.S., Kumar, S., Shin, M.: Soft Robotics: A Review of Recent Developments of Pneumatic Soft Actuators. Actuators **9**(1) (jan 2020) 3
3. Chen, F., Wang, M.Y.: Design Optimization of Soft Robots: A Review of the State of the Art. IEEE Robotics & Automation Magazine **27**(4) (dec 2020) 27–43
4. Lee, C., Kim, M., Kim, Y.J., Hong, N., Ryu, S., Kim, H.J., Kim, S.: Soft robot review. International Journal of Control, Automation and Systems **15**(1) (feb 2017) 3–15
5. Stano, G., Percoco, G.: Additive manufacturing aimed to soft robots fabrication: A review. Extreme Mechanics Letters **42** (jan 2021) 101079

6. Pagoli, A.: Innovative Design of a Soft Robotic Gripper for In-hand Amir Pagoli. PhD thesis, Université Clermont Auvergne (2021)
7. Coevoet, E., Morales-Bieze, T., Largilliere, F., Zhang, Z., Thieffry, M., Sanz-Lopez, M., Carrez, B., Marchal, D., Goury, O., Dequidt, J., Duriez, C.: Software toolkit for modeling, simulation, and control of soft robots. Advanced Robotics **31**(22) (nov 2017) 1208–1224
8. Hiller, J., Lipson, H.: Automatic Design and Manufacture of Soft Robots. IEEE Transactions on Robotics **28**(2) (apr 2012) 457–466
9. Odot, A., Haferssas, R., Cotin, S.: DeepPhysics: a physics aware deep learning framework for real-time simulation. International Journal for Numerical Methods in Engineering **123**(10) (sep 2021) 2381–2398
10. Deshpande, S., Lengiewicz, J., Bordas, S.P.A.: Probabilistic Deep Learning for Real-Time Large Deformation Simulations. Computer Methods in Applied Mechanics and Engineering **398** (nov 2021) 115307
11. Mendizabal, A., Márquez-Neila, P., Cotin, S.: Simulation of hyperelastic materials in real-time using Deep Learning. Medical Image Analysis **59** (apr 2019) 101569
12. Kim, Y., Kim, Y., Yang, C., Park, K., Gu, G.X., Ryu, S.: Deep learning framework for material design space exploration using active transfer learning and data augmentation. npj Computational Materials **7**(1) (sep 2021) 140

Keynote 3 : You have to ask the right questions!

Gernot MEIER

Offenburg University of Applied Sciences

Fig. 1. Dr Gernot Meier (HS Offenburg)

"You have to ask the right questions" is a common, but not to be underestimated basic mindset that has played a central role from Aristotle, Isaac Asimov to beyond Michel Foucault. This basic mindset is familiar to many when making important strategic decisions in business. The basic attitude also has a crucial role in ethical discussions related to the digital future and its forecasting. This also from the knowledge that responsible action is always questioning action in the face of a future that can never be fully controlled.

Since all forms of forecasting are difficult when they concern the further, however, social and technical actors often take shelter in extreme positions or superlatives. Why might this be? Often the reason is that the problems are approached in an undercomplex way and that a technical solutionism believes that ethical questions can be solved and finalized in principle.

Machine ethics, information ethics, robot ethics, algorithms ethics also with all their differences, however, show the following: New technological possibilities give rise to new moral problems. For example, a questioning of the image of man, because digitalization challenges the previous humanistic image of man, or also: Who is responsible for what and why, and is responsibility divisible? Because technology and society are intensively interconnected, especially in the digital field. It is also very interesting here that the responsibility of a scientist or also of companies in the digital field is once again in focus. What will GPT7 say when she has been trained on ethical dilemmas at some point. "I'm sorry, my answers are limited. You need to ask the right questions."

That will always remain the tasks of humans.

Other and General Topics

From LMS to LXP: Extending Moodle with AI-based Recommendations for Learning

Saptadi Nugroho, Prabin Dahal, Gisela Hillenbrand, Katrin Bauer, Teresa Sedlmeier, Daniela Schlemmer, Claudia Schmidt, and Volker Sänger

Offenburg University of Applied Sciences, Badstraße 24, 77652 Offenburg, Germany
{saptadi.nugroho, prabin.dahal, gisela.hillenbrand, katrin.bauer, teresa.sedlmeier, daniela.schlemmer, c.schmidt, volker.saenger}@hs-offenburg.de

Abstract. We propose a Moodle-based LXP (Learning Experience Platform) architecture that extends the classical Moodle LMS (Learning Management System) into LXP. The extension of the Moodle LMS to an LXP is developed to improve the learner's motivation and to enable personalized learning. The first component in our architecture of the Moodle - based LXP is a recommender component based on Artificial Intelligence (AI). It helps learners by proposing appropriate learning resources based on the content they are currently studying. These recommendations are derived from metadata of the learning resources, such as predefined descriptions, number of views, ratings, and comments on the resources.

Keywords: Learning Management System; Learning Experience Platform; recommender component.

1 Introduction

In the last decades, Learning Management Systems (LMS) have been deployed in various educational institutions [1], [2]. An LMS is mainly used to organize and support online learning and includes learning content presentation, communication tools such as forums, tools for tests and exercises, and administrative functions [1], [3]. In practice, the teacher-centric LMS is mainly used for administrative purposes such as course announcements and content distribution, while the effective support of the learning process is not considered [1]. In addition, an LMS provides an identical service to all learners rather than a personalized learning, which would result in an environment that learners perceive as more relevant and motivating [4]. Advanced learning platforms, called Learning Experience Platforms (LXP), are designed to help learners to experience personalized learning by curating content from various sources and recommending it to other learners individually based on their current learning level and personal learning preferences [5]. The content of each particular course can be enhanced through Open Educational Resources (OER) created by other educators and shared by the public to support learning and knowledge sharing in society [6]. In addition to AI-supported content curation and AI-based recommendation, other features of an LXP include an attractive, social media like user interface (user experience), support for social interaction by integrating feedback and content rating features, and search for specific content [7]. Additionally, in many cases, gamification and reward systems are incorporated into LXPs to promote motivation.

In this paper, we present an architecture that extends Moodle [16], an open source LMS widely used in higher education [17], into an LXP. We identified the main components of our approach and designed the core architectures consisting of a recommender

component, recommender plugins, and a feature to rate and provide feedback for learning resources. Recommendation systems in the educational domain are approaches for retrieving and filtering learning resources and similar profiles to provide suggestions for learning resources that are most likely to be of interest to learners and thus support personal learning [18], [8]. The interaction between learners and learning resources can be analyzed and evaluated for aggregating learning resources into a list of personalized recommendations for the learners [4]. In general, recommendation systems can be categorized into content-based recommendations and collaborative recommendations [20], [9]. A hybrid approach can also be applied by combining both content-based and collaborative recommendations [9].

The proposed recommender component in this paper is based on course data and anonymous user data. It supports each learner with appropriate internal or OER learning resources to enable personalized learning. The rating system of a recommender component for learning resources within Moodle gives students the opportunity to rate and review all kinds of learning resources. In addition, the recommender component also contains content-based recommendations that recommend a list of learning resources that are similar to the queried learning resource. The description of the learning resource is compared with the description of other learning resources in the course. We have also designed a collaborative recommendation system component which implements trends in learning resources based on the number of views on the learning resource.

To the best of our knowledge there is an approach to extend Moodle with recommendations proposed by Vera et al. They proposed an educational resource recommendation system based on user preferences and needs that focuses on the knowledge level of students using Python and Moodle [15]. The input data for the recommendation system algorithm are survey data and students' academic grades [15]. In the paper [15], the education resource recommendation system is a collaborative recommendation system, which is different to the content-based recommendation system for similar learning resources and the collaborative recommendation for trending learning resources that we have designed. They used totally different data without an LXP.

2   Enhancing Moodle to an LXP

For many years, at Offenburg University of Applied Sciences the Moodle LMS has been used to provide students with learning resources for their courses. Teachers and students are used to the system. A lot of Moodle content was created, and many features were implemented in the courses, e.g. tests or gamification elements. Hence, replacing the Moodle LMS with a completely new LXP system was not an option. Instead, we decided to gradually expand Moodle into an LXP with AI-based recommendations for learning, so that the existing features of the original Moodle will remain available.

In the summer semester of 2023, we conducted an explorative survey at our faculty in various courses to understand which components of an LXP are important from students' point of view. Ninety-eight students provided their anonymous opinions on AI-based learning recommendations in an online questionnaire. By learning recommendations, we mean suggestions for a learner which learning content could or should be learned next. Examples could be: "Read chapter x in book y" or "Open the following link and attempt the corresponding quiz."

First, we wanted to know if and how students use external learning content. The survey results indicated that many students make use of external learning contents. For example, more than 80% of the participants often pursue external resources to clarify
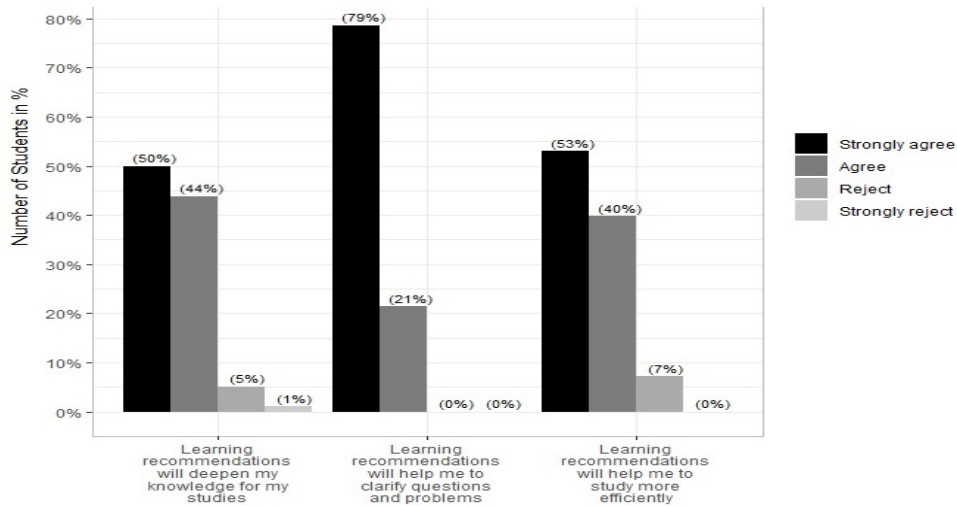
Fig. 1. Expected effect of learning recommendations (n=98).

open questions. And more than 60% utilize external learning contents to prepare for the examination. There are more options for using external content, such as repeating the learning content, exercising, or learning more about the topic. But here, the frequency of usage is lower, which means between 20% and 30% of the participants answered with "often." Regarding the learning recommendations (see Fig. 1), more than 90% of the participants stated that they would use them to deepen their knowledge for their studies (50% strongly agreed, 44% agreed). Similarly, more than 90% of the participants expected that learning recommendations will help them to study more efficiently (53% strongly agreed, 40% agreed). And 100% of the participants believed that learning recommendations will help them to clarify questions and problems (79% strongly agreed, 21% agreed). In addition, 80% (31% strongly agreed, 49% agreed) indicated that they agreed to make their anonymized data available for learning recommendations. However, only 40% of the participants would possibly provide non-anonymized data for personal learning recommendations (10% strongly agreed, 30% agreed). Many students are also willing to support a recommendation system through ratings and tags. 86% of participants would rate learning resources with 1 to 5 stars (agreed and strongly agreed), and 59% would tag learning resources to support content recommendations. In contrast, only 31% of the students are willing to write content summaries for learning resources. Although, more than 85% of the participating students prefer to select learning resources based on summaries. Although our study has limitations, especially due to the selection of students at the faculty of media, we consider for the Moodle LXP:

- In the group surveyed, the demand for learning recommendations is very high.
- Recommendations based only on anonymous data should be provided. Also, recommendations based on personal data could be available for those students who are willing to provide their personal data. Hence, we need one pool for personal and another for anonymous data.
- External learning content seems to be helpful for many students. As the collection of high-quality external learning resources is time-consuming, a component for content curation will be very important for Moodle LXP.
- It could be helpful if students evaluate and rate the contents when a new external content is integrated into a course, and best of all, provide short summaries for other students. In general, a component for ratings is mandatory so that rating data of external and internal learning resources for later recommendations can be collected.

3 Architecture of the Moodle-based LXP

According to the results of the survey, we designed a detailed architecture of the Moodle LMS and the recommender component. Fig. 2 depicts the relationship between the recommender component and the Moodle LMS. The recommender component is designed separately from the LMS, because hardware separation between the component and the original LMS can reduce the load of processing on the LMS. In general, loose coupling of recommender components and Moodle LMS helps to keep Moodle independent. The productive Moodle system (software and hardware) of the university is used for numerous courses from various teachers and students for the learning processes. Moodle is indispensable for the university. The proposed recommender component makes use of anonymized data such as descriptions and user-specific data such as ratings and number of views. The recommender component which is shown in Fig. 2, gathers and retains both anonymous and individual learner data.
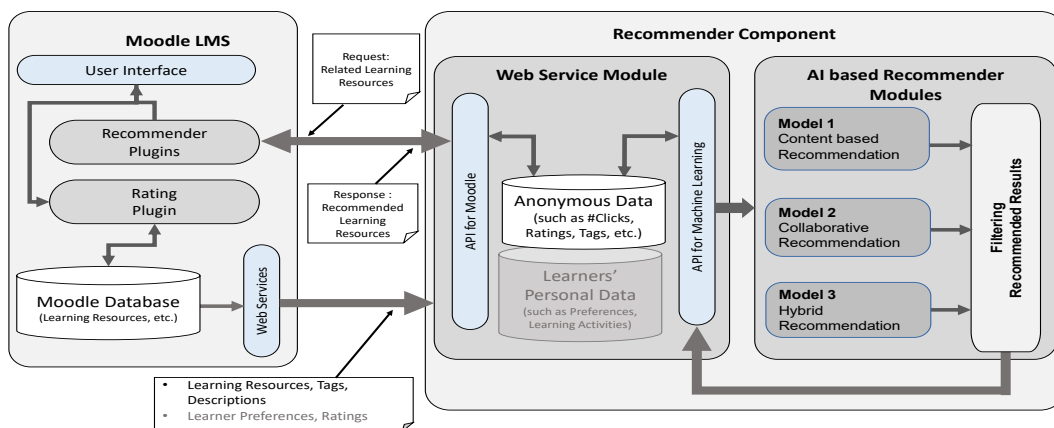
Fig. 2. The architecture of the Moodle LMS and the recommender component.

The main task of the recommender component is to use AI algorithms to examine the most appropriate learning resources and then to provide the results to the recommender plugin. In the current phase of the project only anonymous data is used for providing recommended resources. However, in later versions, we will also collect personal data from students, if they agree to provide their data. The personal data will include learning preferences and information about their learning activities. The data that describes the learning resources and learning behavior can be used for the recommendation process. When a learner in the Moodle system accesses a learning resource, the request to the recommender component is sent. The component then creates a corresponding list of recommended resources. The resulting list is sent to Moodle and presented to the learner in near real-time.

In Fig. 2 we can see that the recommender component is composed of web service module and AI-based recommender Module. Web service module's primary function is to store/transfer data between Moodle LMS and itself. There are two API definitions in the module: one which faces towards Moodle LMS (i.e. provides access to stored data to Moodle plugins) and the other which provides/stores data to/from various machine learning algorithm contained in the AI based recommender module.

4 Moodle LXP Implementation

For the specified architecture, a few Moodle plugins have already been implemented and integrated in our productive Moodle system. These include similar learning resources, rating systems, trending learning resources. In the current phase of the project, we have implemented several block plugins for Moodle LXP that display a list of relevant resources based on the selected content and the user interaction to the courses. When a learner chooses a learning resource (by clicking at it or by hovering the mouse to it for at least two seconds), the ID of the resource is sent to the recommender component, which in succession, computes recommendations by using TF-IDF and returns the recommended learning resources to the plugin, effectively a bidirectional communication. In addition, trending resources listed in Moodle are processed using Z-Score. Learners will now see a list of resources relevant to the resource in question and a list of trending resources as shown in Fig. 3.
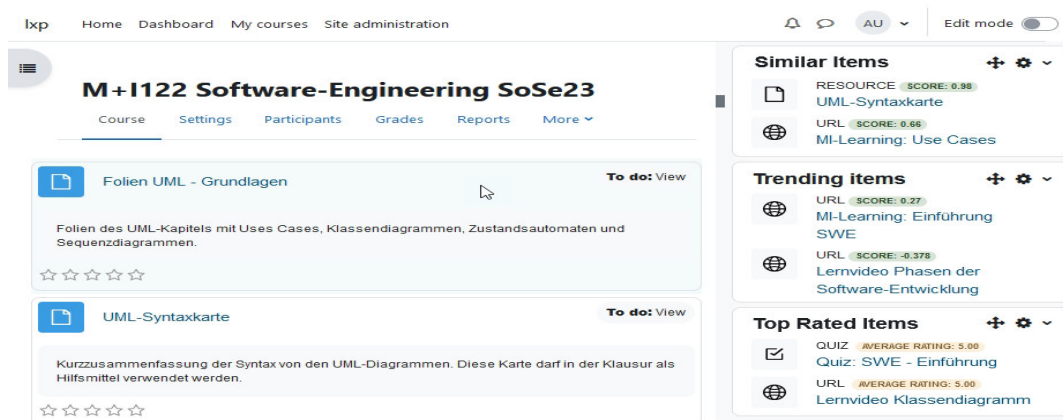
Fig. 3. The similar items, trending items and top-rated plugins for Moodle LXP.

4.1 Content based recommendation

In content based recommendations, the similarity between several content resources can be compared by using metadata that describes these resources. Metadata of resources can be tags, keywords, or descriptions. The content based recommendation uses various methods to analyze the similarity between metadata of learning resources. It uses Natural Language Processing (NLP) which is a subfield of AI that enables machines to understand information created by humans [11]. The Term Frequency and Inverse Document Frequency (TF-IDF) is one of the text processing approaches used in machine learning methods for NLP [12], [13]. TF-IDF is one of the most popular methods for measuring how important description words are to a learning resource document in information retrieval [9], [10], [11]. It is a weighting approach to describe resources in the vector space model so that resources with similar metadata will be rated similarly [23]. Metadata that describe resources can be created, assigned, and collected as input for the recommendation process to generate resource recommendations to users [22]. Metadata such as descriptions for resources or products can affect the quality of recommendations offered by recommendation systems [21]. TF-IDF implemented with SpringBoot framework has been used to recommend learning resources and courses [29]. In our project,

we use TF-IDF to model the similarity between different learning resources in a course module. We utilize the `sklearn` [25], a Python API with libraries, to extract TF-IDF features and calculate cosine similarity.

Information of a course such as course ID, resource ID, resource name, resource type, and description retrieved from the moodle database are stored in the LXP server database. Learning resource types can be URL, resource, folder, quiz, etc. In content based recommendation, resource descriptions, course IDs, and resource IDs are inputs for the recommendation algorithm that analyzes the similarity between learning resources in a course. The result of the algorithm is a list of learning resources that are similar and relevant to a queried learning resource as indicated by the Cosine similarity score. Fig. 4 shows the sequence diagram for the request and response process of Moodle's similar items plugin and web service module of the recommender component. The list of similar learning resources is sent from the AI-based recommender module to the web service module of the recommender component in the form of JavaScript Object Notation (JSON) with the format schema as shown in Fig. 4. In the JSON response, `coursemodules` are the list of similar resources returned from the recommender component. The `cmid` provides the ID of the learning resource and `courseid` is the course the learners are currently viewing.
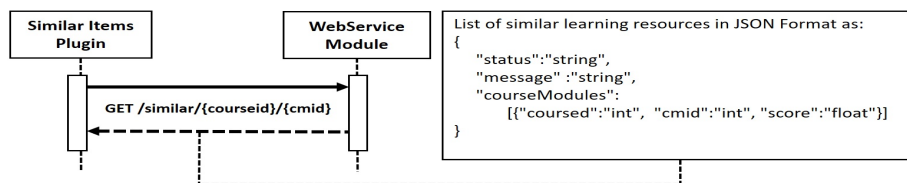
Fig. 4. The sequence diagram of similar items plugin and web service module of recommender component with response in JSON.

4.2 Collaborative recommendation

Collaborative recommendation systems, as the name suggests, takes into consideration what other users interacted with in the past and makes recommendations based on these conditions. Schafer, et. al. thus recommended this type of recommendation system as "people-to-people" [26]. It is also considered a popular and widely implemented system. The system evaluates learners' previous behavior, expressed in the form of ratings, feedback, number of clicks given by learners, to generate recommendations for other learners [24]. New learners can utilize this information to find resources that fit their needs. This process also facilitates finding a good learning path, as a new learner is following the successful steps performed by the other learners.

4.2.1 Rating system Collaborative systems make use of rating and comment features to know how likely a learning resource is recommended to other users. Originally, a rating system in the Moodle LMS is only available for forum topics, database items, and glossary items. Moodle does not provide an overall rating system for learning resources. In order to cover all learning resources in Moodle, we implemented a rating plugin. The new rating plugin allows learners to rate all kinds of learning resources in Moodle like book, wiki, quiz, page, file, URL, etc. Users can rate learning resources with one to five stars, and

they can assign a short review to the resource, such as "The video explains the topic excellently". We calculate the average rating given by all the learners and present it on the Moodle LXP. But the ratings are always presented anonymously.

The result of the assessment for each resource is displayed in the top-rated resources block. The top-rated and displayed resources become a recommendation for other learners to choose. Learners can easily visit resources based on ratings given by other learners.
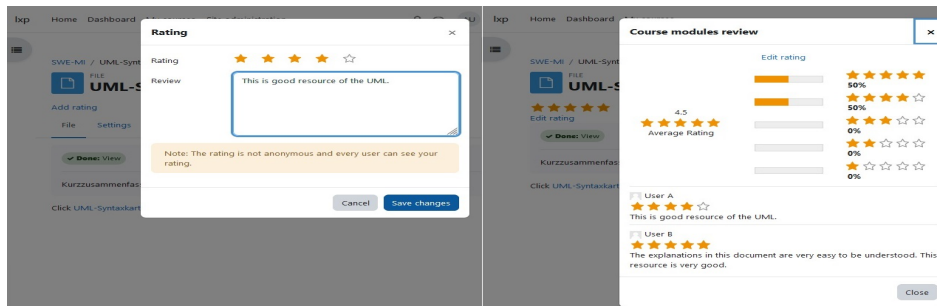
Fig. 5. The window popup of course modules for adding and reviewing ratings.

As a first step, the course Software Engineering was extended with various OER, including learning videos, chapters of e-books from our library, and websites from the internet. When learners select one of the internal or external resources, such as a file or a video, they will see the page of the resource they have selected. They can rate it in two different ways. Fig. 5 shows the user interface when they click the add rating link and the review rating link. The rating data will be stored in the Moodle LMS database. Learners and teachers can view the rating results of each resource.

4.2.2 Trending Items Trending (or Fad) generally refers to describing topics, hashtags, or keywords that are currently popular or gaining a lot of attention. Items that are trending follows a positively skewed, escalating slope, unsteady asymptote and a rapid declining slope [27]. Understanding popular resources can help us identify relevant or significant learning materials. With that knowledge it becomes easier to make effective decisions.

There are some things to consider while populating trending items such as number of clicks, rate of clicks (number of clicks per time period) and the baseline level of clicks. Since, the trending items will only be displayed in a few courses at the moment, the baseline level of clicks is considered zero. That means, there is no minimum number of clicks that an item should have before it is even considered for calculating the score. There are various methods which we can use to calculate trending items such as Slope analysis (Mann-Kendall and Sen's slope analysis), standard score (z-score) [14] and chi squared tests [28]. However, at the moment, the standard score is selected to calculate the trending items.

Trending items are displayed by a Moodle's block plug-in and currently all the data and calculations are stored in the Moodle's database. In the future, the recommender component will store the data and the calculations also will be performed by the recommender module. The plug-in's architecture is based on strategy pattern used in software development to make it flexible. Having this flexibility to Moodle's plug-in brings us ease in further research and testing of various algorithms in the future. Fig. 6 shows such a design.
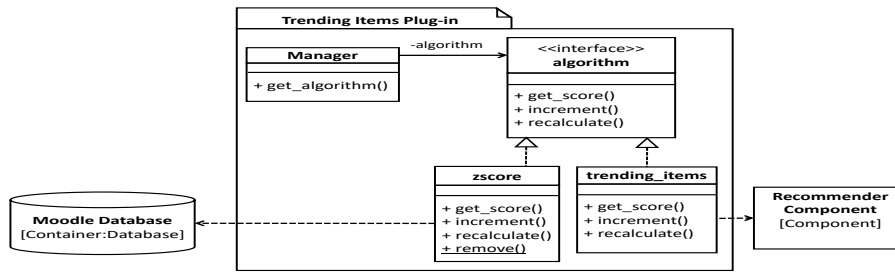
Fig. 6. Design of trending items plug-in and its interaction with recommender component.

The plug-in records the number of clicks for a certain time period. For instance, seven days. These seven days are called a window. The window is divided into seven panes each of width of one day. The window then slides each day, removing the last pane and adding a new pane corresponding to the current day, effectively becoming a sliding window.

Standard score (z-score) indicates how many standard deviations a datum (number of clicks) is above/below a standard deviation. So by definition, we calculate mean and standard deviation on the data. We calculate average number of clicks (mean) and standard deviation by using the data from history panes. Then we calculate the average z-score of the current panes. This is further illustrated in Fig. 7. Calculating average and z-scores for all activities/resources of a course takes time. So there is a scheduler task in the plug-in which is run daily.
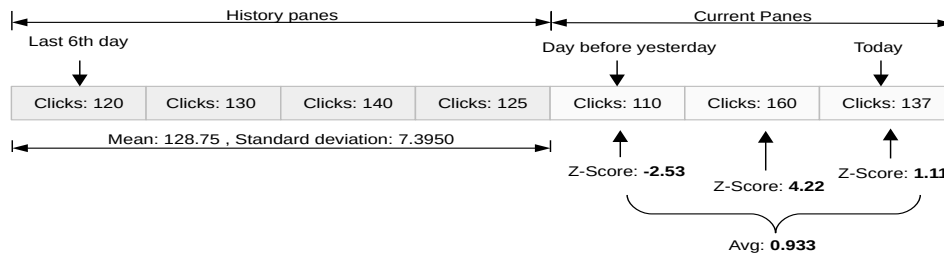
Fig. 7. Window panes with calculated z-score.

5 Conclusion and Future Work

In this paper we presented an architecture for an LXP built on top of a classic Moodle LMS. We designed and implemented initial parts of the recommender component and the initial plugins: a content based recommender plugin for similar learning resources, a collaborative recommender plugin for trending learning resources, and a rating plugin for all types of learning resources in the Moodle LMS. Based on the architecture presented, a separate server was set up for the recommender component and the interfaces to the productive Moodle server were defined.

For recommending learning materials, we have curated a variety of open educational resources and extended several courses in Moodle with these resources. We have already integrated the rating plugin in few courses in summer semester of 2023, and now we are in the process of collecting feedbacks from the students. Additionally, we are planning to

deploy similar items plugin, trending items plugin in the following winter semester and gather response and feedbacks from the students to further evaluate Moodle as an LXP platform.

In the next steps, we will implement of various components including hybrid recommendation systems and filtering modules. Furthermore we plan deeper investigations of various AI methods and different AI-based recommendation algorithms to support tailored and personalized learning.

6 Acknowledgement

References

1. Kipp, K. Exploring The Future Of The Learning Management System. *International Journal On Innovations In Online Education.* **2** (2018), `https://doi.org/10.1615/IntJInnovOnlineEdu.2018028353`.
2. Bersin, J. Insights on Corporate Talent, Learning, and HR Technology. *A New World Of Corporate Learning Arrives: And It Looks Like TV.* (2017,6), `https://joshbersin.com/2017/06/a-new-world-of-corporate-learning-arrives-and-it-looks-like-tv/`. Last accessed 01 Feb 2023.
3. Malikowski, S., Thompson, M. & Theis, J. A Model for Research into Course Management Systems: Bridging Technology and Learning Theory. *Journal Of Educational Computing Research.* **36**, 149-173 (2007,3), `https://doi.org/10.2190/1002-1T50-27G2-H3V7`
4. Meier, C. & Gori, S. Adaptive Lernumgebungen - Lernwirksamkeit und Umgang mit Daten im Blick behalten. *Schwerpunkt Talentmanagement / Personal Entwicklung.* pp. 28-33 (2019,10), `https://www.scil.ch/wp-content/uploads/2019/11/Meier-Gori-adaptive-Lernumgebungen-Personalfuehrung-2019-10-.pdf`. Last accessed 17 Oct 2022.
5. Valamis Group What is an LXP?. (2023,3), `https://www.valamis.com/hub/learning-experience-platform`. Last accessed 01 Apr 2023.
6. UNESCO The 2019 UNESCO Recommendation on Open Educational Resources (OER): supporting universal access to information through quality open learning materials. (2022), `https://unesdoc.unesco.org/ark:/48223/pf0000383205`. Last accessed 01 Apr 2023.
7. Stoller-Schai, D. Was machen wir mit "Learning Experience Platforms"?. (2020), `https://www.elearning-journal.com/2020/08/12/was-machen-wir-mit-lxp/`. Last accessed 12 Apr 2023.
8. Zhang, Q., Lu, J. & Zhang, G. Recommender Systems in E-learning. *Journal Of Smart Environments And Green Computing.* (2022), `http://dx.doi.org/10.20517/jsegc.2020.06`
9. Adomavicius, G. & Tuzhilin, A. Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. *IEEE Transactions On Knowledge And Data Engineering.* **17**, 734-749 (2005). `https://doi.org/10.1109/TKDE.2005.99`
10. Wang, X., Gülenman, T., Pinkwart, N., Witt, C., Gloerfeld, C. & Wrede, S. Automatic assessment of student homework and personalized recommendation. *2020 IEEE 20th International Conference On Advanced Learning Technologies (ICALT).* pp. 150-154 (2020). `https://doi.org/10.1109/ICALT49669.2020.00051`
11. Ohata, E., Mattos, C., Gomes, S., Reboucas, E. & Rego, P. A Text Classification Methodology to Assist a Large Technical Support System. *IEEE Access.* **10** pp. 108413-108421 (2022), `https://doi.org/10.1109/ACCESS.2022.3213033`

12. Daga, I., Gupta, A., Vardhan, R. & Mukherjee, P. Prediction of Likes and Retweets Using Text Information Retrieval. *Procedia Computer Science*. **168** pp. 123-128 (2020), `https://doi.org/10.1016/j.procs.2020.02.273`

13. Anila Sharon, J., Hepzibah Christinal, A., Abraham Chandy, D. & Bajaj, C. Application of intelligent edge computing and machine learning algorithms in MBTI personality prediction. *Intelligent Edge Computing For Cyber Physical Applications*. pp. 187-215 (2023), `https://doi.org/10.1016/B978-0-323-99412-5.00003-4`

14. Marrone, M. Application of entity linking to identify research fronts and trends. *Scientometrics*. **122**, 357-379 (2020,1), `https://doi.org/10.1007/s11192-019-03274-x`

15. Vera, A. & González, C. Educational Resource Recommender Systems Using Python and Moodle. *Computational Science And Its Applications – ICCSA 2022 Workshops*. **13380** pp. 15-30 (2022), `https://doi.org/10.1007/978-3-031-10542-5_2`, Series Title: Lecture Notes in Computer Science

16. Moodle Pty Ltd Welcome to the Moodle community. , `https://moodle.org/`. Last accessed 18 Oct 2022.

17. Hill, P. State of Higher Ed LMS Market for US and Canada: Year-End 2022 Edition. (2023,1), `https://philhillaa.com/onedtech/state-of-higher-ed-lms-market-for-us-and-canada-year-end-2022-edition/`. Last accessed 18 Oct 2022.

18. Ricci, F., Rokach, L. & Shapira, B. Recommender Systems: Techniques, Applications, and Challenges. *Recommender Systems Handbook*. pp. 1-35 (2022), `https://doi.org/10.1007/978-1-0716-2197-4_1`

19. Rivera, A., Tapia-Leon, M. & Lujan-Mora, S. Recommendation Systems in Education: A Systematic Mapping Study. *Proceedings Of The International Conference On Information Technology & Systems (ICITS 2018)*. **721** pp. 937-947 (2018), `https://doi.org/10.1007/978-3-319-73450-7_89`, Series Title: Advances in Intelligent Systems and Computing

20. Balabanović, M. & Shoham, Y. Fab: content-based, collaborative recommendation. *Communications Of The ACM*. **40**, 66-72 (1997,3), `https://doi.org/10.1145/245108.245124`

21. Belém, F., Silva, R., De Andrade, C., Person, G., Mingote, F., Ballet, R., Alponti, H., De Oliveira, H., Almeida, J. & Gonçalves, M. "Fixing the curse of the bad product descriptions" – Search-boosted tag recommendation for E-commerce products. *Information Processing & Management*. **57**, 102289 (2020,9), `https://doi.org/10.1016/j.ipm.2020.102289`

22. Bogers, T. Tag-Based Recommendation. *Social Information Access*. **10100** pp. 441-479 (2018), `https://doi.org/10.1007/978-3-319-90092-6_12`, Series Title: Lecture Notes in Computer Science

23. Wang, D., Liang, Y., Xu, D., Feng, X. & Guan, R. A content-based recommender system for computer science publications. *Knowledge-Based Systems*. **157** pp. 1-9 (2018,10), `https://doi.org/10.1016/j.knosys.2018.05.001`

24. Schafer, J., Frankowski, D., Herlocker, J. & Sen, S. Collaborative Filtering Recommender Systems. *The Adaptive Web*. **4321** pp. 291-324 (2007), `https://doi.org/10.1007/978-3-540-72079-9_9`, Series Title: Lecture Notes in Computer Science

25. Scikit-Learn Consortium at Inria Foundation scikit-learn Machine Learning in Python. , `https://scikit-learn.org/stable/`. Last accessed 18 Oct 2022.

26. Schafer, J., Konstan, J. & Riedl, J. E-Commerce Recommendation Applications. *Data Mining And Knowledge Discovery*. **5**, 115-153 (2001), `https://doi.org/10.1023/A:1009804230409`

27. Aguirre, B., Quarantelli, E. & Mendoza, J. The Collective Behavior of Fads: The Characteristics, Effects, and Career of Streaking. *American Sociological Review*. **53**, 569 (1988,8), `https://doi.org/10.2307/2095850`

28. Koletsi, D. & Pandis, N. The chi-square test for trend. *American Journal Of Orthodontics And Dentofacial Orthopedics*. **150**, 1066-1067 (2016,12), `https://doi.org/10.1016/j.ajodo.2016.10.001`

29. Anming, H. Research on Course Resource Recommendation System Based on Feature Selection. *2020 Asia-Pacific Conference On Engineering Education, Advanced Education And Training (EEAET 2020)*. (2020). Last accessed 01 Apr 2023.

Automated Machine Learning for Business Decision Simulation

Reinhard Bauer[1] and Tyler Marangi[2]

[1] Karlsruhe University of Applied Sciences
`reinhard.bauer@h-ka.de`
[2] SAP SE
`tyler.marangi@sap.com`

Abstract. In many industries it is common to support business decisions with the help of simulation models. Such models can be used to predict the outcome of a series of decisions. Often, the number of possible options is high, making it difficult to identify good candidates for simulation. Optimization models can help to find such candidates. Building optimization models can be an arduous task for a variety of reasons, such as computational complexity, algorithmic complexity, or required expertise. Many times, simulation models are readily available.

We propose to use black-box optimization techniques on simulation models to identify good business decisions. This allows generic optimization strategies to be applied to simulation models without additional effort or knowledge. There are numerous existing research papers on black box optimization. A traditional application of such routines is parameter optimization for algorithm tuning, e.g. solvers for mixed-integer linear programming. In recent years, a major focus has shifted to automated machine learning. The challenge here is to find good hyper-parameters for machine learning approaches or even to choose the best strategy automatically.

We sketch different approaches to automated machine learning and how to apply them on simulation models. To demonstrate the benefit of our approach, we create a use-case for energy optimization in a commercial facility, such as an office building. Finally, the performance of the different approaches is evaluated in an experimental study.

1 Introduction

In recent years, improved data availability resulted in demand for better use of the data collected. Business decision support systems have a long history, often originating in the field of operations research. Both simulation as well as optimization models are well-established tools to support business decisions. While most applications allow for custom-tailored, hand-made optimization models, this approach often is too costly to apply in commercial practice.

In case a simulation model already exists for a concrete application, we propose to use techniques from the field of black-box optimization / automated machine learning in order to automatically add optimization capabilities to this model. The general workflow of our approach can be seen in Figure 1.

The business decisions are included as parameters of the simulation model (visualized as red boxes) and have to be fixed before starting to simulate. Once those parameters are determined, the simulation phase starts. This involves one simulation run for a deterministic model and a multitude of runs for a stochastic model. After simulating, the

simulation outcome is evaluated. To that end, an objective function is specified which models targets like profit, cost or throughput.

The black-box optimization routine can interact with the simulation model only by setting the parameters (specifying the input of the function to be optimized) and reading the resulting objective value (evaluating the output of the function). In a loop, the black-box optimization routine iteratively identifies new promising candidate parameter sets, triggers the simulation run and reads back the objective value. New promising candidates are then identified and the loop starts all over. The actual way in which the next candidates are identified differs between the approaches to black-box optimization. To the best of our knowledge, this approach has not been applied before to business decision simulation.
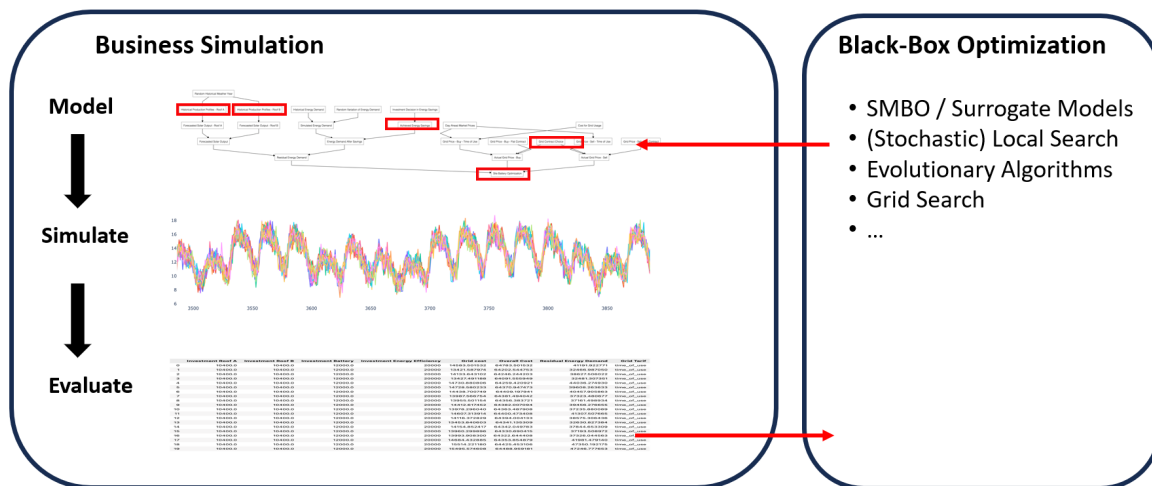
Fig. 1. Interaction between simulation and black-box optimization routine.

Related Work. *Black-box optimization* is a sub-field of mathematical optimization where a function $f : D \to \mathbb{R}$ is to be numerically optimized without further strong assumptions on properties of f, in particular without using derivatives. In a typical setup, the only way to obtain information about f is to evaluate f for a given input value, which in that context often is painstakingly time-consuming. There exist numerous methods for black box optimization, an overview can be found in [1]. An important distinction is between *model-free* and *model-based* methods. The first class encompasses most of the common (meta-)heuristics such as *random search*, *evolutionary algorithms*, *simulated annealing*, *hill-climbing*, *direct search* and many more. The common property of the second class is to build an approximation model of f, called the *surrogate*. The underlying assumption here is, that f is somehow smooth (which arguably is an assumption for most of black-box optimization). Good results are reported for surrogates based on gaussian processes and random forests.

Automated machine learning (AutoML) (see [2] for an overview) is the task to automatically identify good hyperparameters for machine learning methods. Many methods in AutoML do not exploit special knowledge on the technique to be trained and hence can be seen as a special case of black-box optimization. An important algorithm stemming from this community is *SMAC* [3]. Many AutoML-methods are readily available as open-source libraries such as *Ray* (`ray.io`), *Optuna* (`optuna.org`), *SMAC3* (`github.com/automl/SMAC3`), *Scikit-Learn* (`scikit-learn.org`).

It already is widespread practice to combine operations research with artificial intelligence. For a recent survey paper, see [4].

2   Black-Box Optimization for Simulation Models

We consider the following algorithm classes: Stochastic search, local search, grid search, a genetic algorithm, a simple variant of sequential model-based optimization using random forests as surrogates and some highly tuned model-based algorithms offered as part of the SMAC3-library (for details see github.com/automl/SMAC3). For each approach (except grid search) we allow a predefined budget of n simulation runs.

2.1   Grid search

For each parameter p_i, a set of possible sample points is selected. Grid search is an exhaustive search over all possible combinations of those sample points.

For discrete parameters, we select all possible values as sample points. For continuous parameters and a given number of samples per parameter, we select sample points equally spaced over the entire domain (starting and ending at the domain boundaries).

In many real-world settings, grid search is too slow to apply. However, we use it to approximate the optimal solution of our use case.

2.2   Stochastic (or random) search

Stochastic search evaluates n random instances for which each parameter has been chosen uniformly at random in the respective domain. This approach can be used as a baseline for comparison with other more sophisticated strategies.

2.3   Stochastic local search

Stochastic local search starts with a random instance (selected as in stochastic search) as an *active instance*. Then, new instances in a *neighborhood* around the active instance are sampled randomly and are subsequently evaluated. If an instance S is found to be better than the currently active instance, it becomes the new active instance. The approach stops after n instances have been evaluated.

We apply *restarts*: If, for some iterations, no improvement has been realized around the currently active instance, a new active instance is chosen uniformly at random.

We allow the target value to deteriorate: With a certain probability, a worse parameter set can also become the new active instance.

When sampling from the neighborhood, we proceed as follows: For a continuous parameter p_i with domain $[l_i, u_i]$ we sample from a normal distribution whose mean is the corresponding parameter value of the active solution and whose variance is $0.1(u_i - l_i)$. We clip values to the domain bounds if necessary. When sampling a discrete parameter, we keep the current value with probability $1 - c$ and choose a new value uniformly at random with probability c where c is a tuning parameter.

2.4 Evolutionary algorithms

The basics of evoluationary algorithms can be found in any introductionary textbook on the matter. We use the following specifics:

- Tournament selection with one elitist individuum in the next generation.
- Crossover: For each parameter p_i, the value is chosen uniformly at random from one of the two parents.
- Mutation uses the neighborhood definition of the local search approach: The routine to mutate an individuum is the same as to randomly identify a new instance in the neighborhood of the currently active solution.

2.5 Model-based approach using random-forests as surrogates

We use a minimalistic model-based approach:

- Exploration is done only in an introductionary phase. Here, for a number of samples k, a Sobol sequence [5] is constructed with dimensionality given by the number of parameters. Afterwards, the actual values of the discrete parameters are obtained by rounding. Each instance is evaluated by a simulation run.
- The following exploitation phase comprises $n - k$ iterations: At each step, first a random forest is trained on all instances evaluated so far. Features are the respective parameters. Targets are the corresponding objective values.
  Then, a Sobol sequence with 10.000 samples is created as described above. The random forest is used to estimate the objective value of each sample. The most promising candidate then is evaluated by a simulation run.

2.6 SMAC3

We test SMAC3 as a representative of a state-of-the-art framework for black-box optimization.

3 Application Scenario

3.1 Overview

As a use case to test the efficiency of our approach, we model the decisions involved in managing the energy aspects of a mid-sized office building. These comprise long- and short-term decisions. The long-term decisions are:

- The building has two roofs on which photovoltaic panels can be installed. Orientation and inclination of both roofs differ as does the maximum number of panels that can be installed. The decision to be taken is, for each roof, the photovoltaic capacity to build.
- It is possible to invest in energy efficiency measures, namely more efficient lighting and more efficient ventilation. Exactly one out of four options is to be chosen: Do not invest, invest only in lighting, invest only in ventilation, invest in lighting and ventilation.
- A battery energy storage system can be purchased. The power rating of the battery is fixed. The exact capacity of the system is to be decided.

- Two contract types are available for electricity procurement: A flat tariff as well as a tariff depending on the price of the day-ahead market.

The time-granularity of the model is one hour. From an operational point of view, the following decision must be taken for each hour of the model horizon:

- Should the battery charge, discharge or do nothing (and to which extent)?

Furthermore, the energy demand of the building is modeled. This constitutes the simulation part of the model:

- The actual power demand of the building is modeled based on historic energy consumption with volatility added.

Annualized cost is given for all investments. The objective is to minimize the expected energy cost over a given horizon including the annualized investment and the income of selling power to the grid.

3.2 Detailed Description

The model horizon consists of hourly periods $1, \ldots, T$. The model incorporates stochastic elements and hence is simulated n times for a given number n. The model outcome is the mean energy cost over all simulation runs. The overall model can be broken down into three parts: Residual load calculation, contract type/tariff choice and the battery model. All long-term decisions fixed, the former two parts can be calculated independently. Their outcome is an input to the battery model which decides the operational decisions.

Residual load. An estimate for the future hourly electrical energy demand is given as input data. We assume that investing in an energy efficiency measure reduces hourly demand by a fixed relative value. A jump diffusion process is added to the expected energy demand to account for the volatility of actual electricity consumption. Finally, expected solar production is subtracted. To that end, a historic year in the range 2005 to 2016 is chosen uniformly at random. The *expected solar power production* equals the power production that the considered number of pv-panels with same location and orientation would have produced in the same historic range. *Residual load* is the actual load after adjusting for the investments in energy efficiency and expected solar power production. Please note that residual load can be negative, resulting in *excess energy*. Residual load is handed over as input to the battery model.

Grid Tariff. There are two different contract types: Firstly, a *flat tariff* where the price k_t for buying energy from the grid is equal for all hours t, as is the price v_t for selling energy. Secondly, a *flexible tariff* where those prices are individual for each hour. We assume those prices are known for the entire model horizon.

Battery model. The battery is modelled using a mixed-integer linear program. Perfect foresight over the model horizon is assumed. Battery ageing is not taken into account (neither cyclic ageing, nor calendar degradation). Influence of temperature or other environmental factors are not taken into account. Input values are

- The prices k_t and v_t for buying/selling energy from/to the grid at hour t.
- The residual load d_t at hour t. Positive values mean demand, negative values mean excess energy.

- The battery efficiency η (we assume the same efficiency for charging and discharging).
- The battery capacity c_{max}.
- The battery power s_{max}.

Decision variables are

- Amount of energy p_t^+ and p_t^- bought and sold from and to the grid at hour t, respectively.
- Amount of energy s_t^+ charged into the battery (before considering efficiency) at hour t.
- Amount of energy s_t^- discharged from the battery (after considering efficiency) at hour t.

The resulting linear program is

$$\text{minimize} \sum_{t=1}^{T} k_t p_t^+ - v_t p_t^-$$

subject to

$$
\begin{aligned}
p_t^+ + s_t^- &= d_t + p_t^- + s_t^+ & &\text{for } t = 1, 2, \ldots, T & &(1)\\
s_{t+1} &= s_t + s_t^+ \eta - s_t^-/\eta & &\text{for } t = 1, 2, \ldots, T-1 & &(2)\\
s_t &\leq c_{max} & &\text{for } t = 1, 2 \ldots, T & &(3)\\
s_t^+ &\leq s_{max} \cdot charging_t & &\text{for } t = 1, 2 \ldots, T & &(4)\\
s_t^- &\leq s_{max} \cdot discharging_t & &\text{for } t = 1, 2 \ldots, T & &(5)\\
charging_t + discharging_t &\leq 1 & &\text{for } t = 1, 2 \ldots, T & &(6)\\
0 &\leq p_t^-, p_t^+, \; s_t, \; s_t^+, \; s_t^- & &\text{for } t = 1, 2 \ldots, T & &(7)\\
charging_t, \; discharging_t &\in \{0, 1\} & &\text{for } t = 1, 2 \ldots, T &&
\end{aligned}
$$

The modeled aspects of each equation are as follows: (1) Equality of supply and demand, (2) state of charge, (3) maximum capacity, (4) indicate charging, (5) indicate discharging, (Either charge or discharge), (7) non-negativity. Note, that the constraints to forbid simultaneous charging/discharging are only necessary when energy prices may be negative.

Potential Enhancements. It is worthwhile to add volatility to the flexible power tariff, incorporate battery ageing and model the energy demand more detailedly.

4 Experimental Evaluation

We conduct a computational study to assess the quality of the black-box optimization routines. The problem to be solved is the scenario explained in the last section. All experiments were performed on a MacBook Air with M1 Processor and 8GB RAM, running Python 3.8.12. Mixed-integer linear programs were solved with the COIN-OR Branch-and-Cut Solver (www.coin-or.org).

Grid Search. As the global optimum is unknown, we perform a grid search as a way to estimate the best possible objective value as well as the variability of the problem. Discrete decisions (efficiency investment and contract choice) are considered in their entirety. For each continuous parameter (installed photovoltaic capacity on each roof, battery sizing), ten equidistant values were chosen. The required run-time was 16h 26min,

the best solution found had a target value of 33303. The distribution of all evaluated parameter choices is as follows:
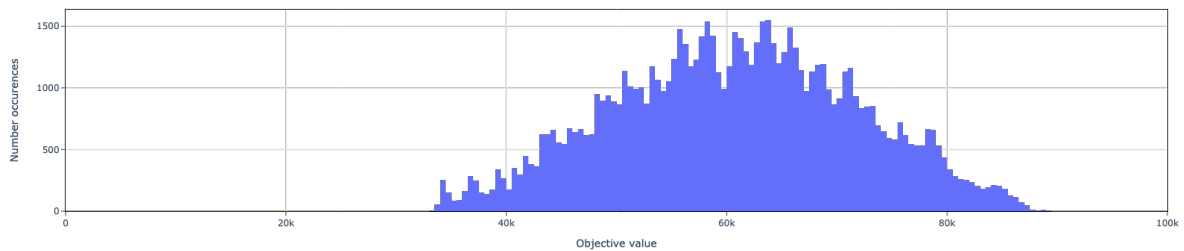
Performance of the black-box routines. The run-time of all approaches considered was dominated by the time required for simulation. The overhead for the optimization algorithm itself was negligible, even for the surrogate-based approaches. In this study we allow a time-budget of 50 simulation runs per approach. Some methods (e.g. the evolutionary strategy) allow for parallel runs, while others (e.g the surrogate-based algorithms) must be executed sequentially. Counted sequentially, the overall run-time of each strategy roughly is 1:40 min.

Solution quality of all black-box routines is near-optimal (assuming that the grid-search is dense enough to make that judgement). The individual best objective values found are

Grid Search	33303
Stochastic Local Search	33979
Evolutionary Algorithm	34184
SMBO - Random Forest	34217
SMAC3 - Hyperparameter Optimization Facade	34419
SMAC3 - Black Box Facade	34608
Stochastic Search (for comparison)	43037

We interpret the remaining difference in the solution quality as random noise rather than a systematic advantage of one strategy over the other on this problem. All approaches seem robust enough to "solve" the use case. However, we want to stress that the performance can be quite application specific and may depend on the choice of tuning-parameters of the algorithm.

Progress Charts. In the following, we show a *progress chart* for each algorithm: Each simulation outcome is visualized by a blue dot in the respective graph. The red line shows the progression of the best objective value found so far during iteration.

Stochastic Search. This algorithm acts as a baseline to compare the other algorithms with. 50 instances are chosen uniformly at random, simulated and evaluated. The approach ends with a target value of 43037 which is far away from the best known solution.
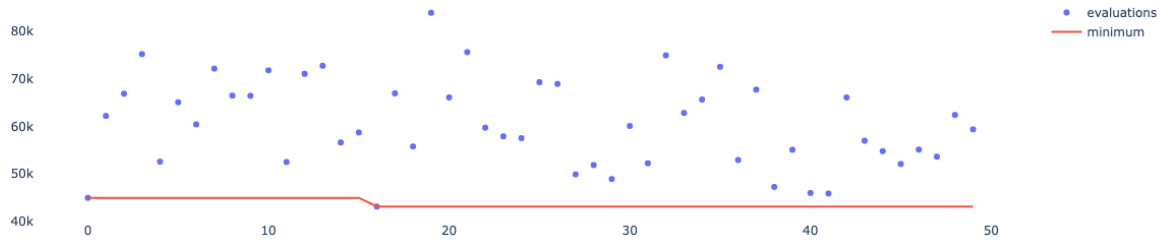
Evolutionary Algorithm. Five generations with population size ten. Tournament selection with one elitist individuum in the next generation. Discrete decisions are mutated uniformly at random with probability 0.1. Continuous decisions are mutated according to a normal distribution for which the mean is the current value and the variance is 0.1 times the difference between maximum and minimum possible value. Mutated decisions that lie outside the decisions domain are clipped to the next feasible value.
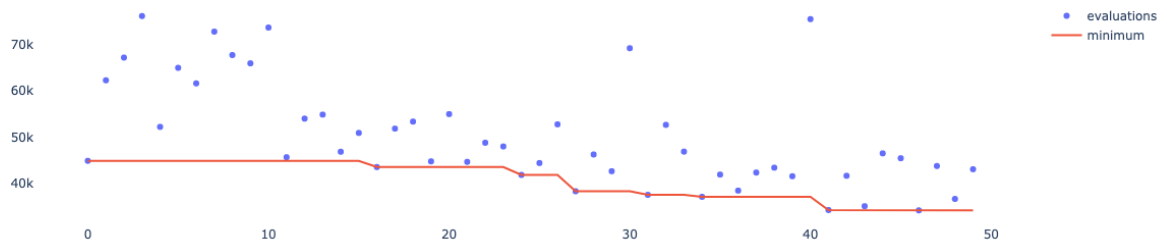
SMAC3 - Black Box Facade. Default parameters are applied.
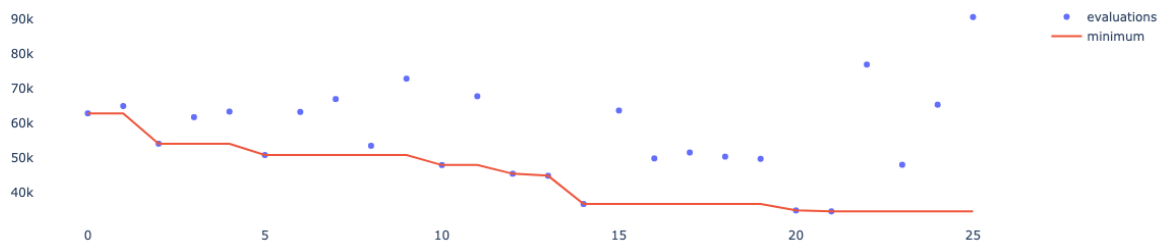
SMAC3 - Hyperparameter Optimization Facade. Default parameters are applied.
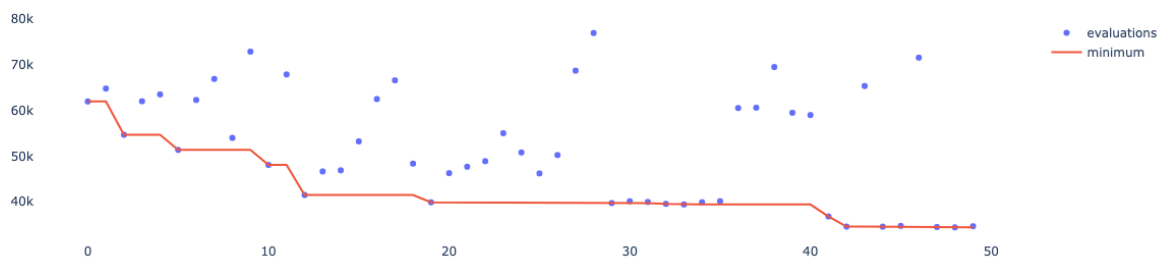
SMBO - Random Forest. The initial exploration phase uses 15 simulation runs using a Sobol-design. Each surrogate random forest is evaluated 10.000 times using a Sobol-design.
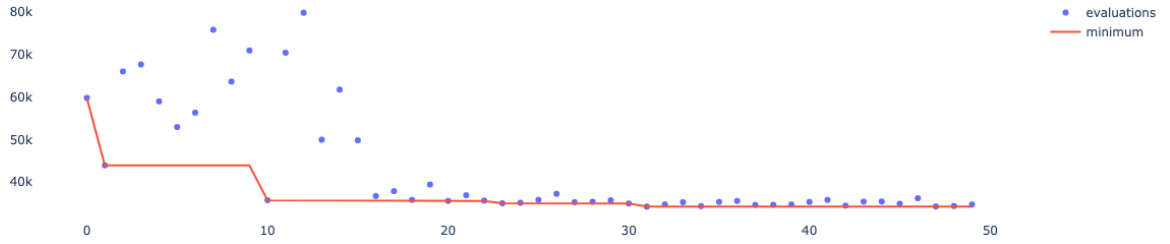
Stochastic Local Search. The probability to accept a worse solution as new incumbent is 0.1. A random restart is performed after 10 consecutive iterations that do not improve the best solution. The neighborhood is similar to the mutation of the evolutionary algorithm.
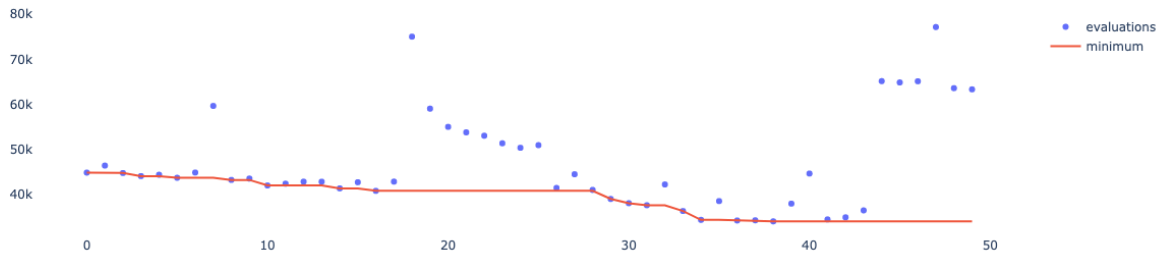
5 Conclusion

We proposed to use black-box optimization techniques on simulation models to identify good business decisions. This is useful in situations where building an optimization model is too time-consuming or costly, in particular when a simulation model already exists. To assess the quality of the approach, we conducted a small computational case study. We modeled the energy procurement of a mid-sized office building, making strategic decisions such as investment choices and simulating operations over time. The black-box routines were able to find near-optimal solutions demonstrating the efficiency of the approach. To the best of our knowledge, the method of automatically adding optimization capability to existing business decision simulations by using surrogate-based black-box optimization routines is new.

The obvious drackbacks of this approach are the lack of dual bounds on the cost function and the limited experimental experience. While the first aspect is inherent to the applied techniques, the experimental knowledge should be extended by adding more use-cases which we propose as further work on the topic.

Bibliography

[1] Audet, C., Hare, W.: Derivative-free and blackbox optimization. Springer (2017)

[2] Hutter, F., Kotthoff, L., Vanschoren, J.: Automated machine learning: methods, systems, challenges. Springer Nature (2019)

[3] Hutter, F., Hoos, H.H., Leyton-Brown, K.: Sequential model-based optimization for general algorithm configuration. In: Learning and Intelligent Optimization: 5th International Conference, Rome, Springer (2011)

[4] Gupta, S., Modgil, S., Bhattacharyya, S., Bose, I.: Artificial intelligence for decision support systems in the field of operations research: review and future scope of research. Annals of Operations Research (2022)

[5] Joe, S., Kuo, F.Y.: Constructing sobol sequences with better two-dimensional projections. SIAM Journal on Scientific Computing **30**(5) (2008)

Investigating Learning Transferability and Deployment for Neural NILM Strategies

Yacine Belguermi[1], Gilles Hermann[1], and Patrice Wira[1]

IRIMAS – Institut de Recherche en Informatique, Mathématiques, Automatique et Signal
IUT de Mulhouse, Université de Haute-Alsace
`yacine.belguermi@uha.fr, gilles.hermann@uha.fr, patrice.wira@uha.fr`

Abstract. This research focuses on Non-Intrusive Load Monitoring (NILM), a crucial component of energy management, enabling users to effectively monitor and reduce energy consumption. In a previous work, we developed a hybrid nerual network with combining 1D Convolutional Neural Networks (1D CNN) and Long Short-Term Memory networks (LSTM) for disaggregating 11 appliances using the AMPds2 dataset. Based on this work, we aim to create a generalized model capable of transferring knowledge gained from one building to others. Promising results have been achieved through fine-tuning techniques, indicating the model's adaptability and effectiveness in diverse settings. Notably, our research breaks new ground by employing transfer learning for the disaggregation of 10 appliances, surpassing previous work while maintaining a lower complexity. This study underscores the potential of NILM techniques in energy conservation and establishes a foundation for scalable, transferable models that can contribute to sustainable energy.

Keywords: Non-Intrusive Load Monitoring, energy disaggregation, Neural Networks, Transfer learning.

1 Introduction

Today's technologies are playing a pivotal role in addressing the global energy crisis by reshaping the way we design, construct, and manage buildings, giving rise to the concept of "smart buildings". In response to increasing energy demands, climate change concerns, and the need for sustainable practices, smart buildings leverage advanced technologies to optimize energy efficiency and resource utilization. These technologies enable buildings to dynamically adapt and monitor their heating, cooling, lighting, and power usage, thereby reducing energy consumption. Smart building technology has emerged as a key concept in modern urban infrastructure, prioritizing improvements in energy efficiency and sustainability [1]. A key facet of smart building operation involves the ongoing monitoring of electrical consumption by individual appliances, an aspect that has gained significant attention in recent years [2]. Through meticulous tracking and analysis of energy usage patterns within smart buildings, opportunities to address inefficiencies and implement precise energy-saving measures become evident. Authors in [3] reviewed existing research about the outcome of continuous and real-time monitoring of contained appliances within multiple buildings over multiple geographical areas. The results of the mentioned study demonstrate that appliance monitoring can yield energy savings of up to 23%. Nevertheless, it is imperative to acknowledge that implementing load monitoring in buildings necessitates the incorporation of additional hardware and resources [2, 4].

Actually, we can avoid the issue of installing more hardware to monitor appliances by using more sophisticated approaches. These methods involve enhancing the capabilities

of the primary meter, enabling it to intelligently process total building energy consumption data for the purpose of monitoring individual appliances [2, 4, 5]. This methodology is commonly referred to as "Non-Intrusive Load Monitoring" (NILM) which has gained prominence in the field of energy management and smart building technology [2]. Recent research in NILM primarily focuses on using Machine Learning (ML) algorithms to address the challenge posed by the unpredictable nature of electrical load behavior [2]. In our previous study [6], we introduced a new neural network design to separate individual appliance signals from a main meter reading using a publicly available dataset. However, there is a common limitation with these solutions, generalizability was not verified across various buildings [5]. This issue arises because training ML models demands extensive datasets [5], which often means installing sub-meters for an extended period. Consequently, the main goal of our research is to test the applicability of our previous findings on a different dataset.

2 Non-Intrusive Load Monitoring (NILM) with Machine Learning

2.1 Overview and motivation

Non-Intrusive Load Monitoring (NILM), also known as energy disaggregation, is a technique used in the field of energy monitoring and management [4]. It involves the process of extracting detailed information about the individual appliances and devices within a building when there is only one meter measuring the total energy consumption for the entire building. This is modeled using equation (1) [4, 5]. In this scenario, NILM relies solely on the total power consumption signal, which is the aggregate power consumption of all appliances and devices operating within the building. By analyzing the unique electrical signatures and patterns associated with various appliances, NILM algorithms can identify when specific appliances are in use and estimate their power consumption. This enables users to gain insights into the energy usage of individual devices without the need for additional meters or sensors on each appliance, making it a non-intrusive and cost-effective method for monitoring and managing energy consumption in buildings. The total power consumption $P(t)$ is the sum of the unknown individual appliances' consumption $p_i(t)$ among N appliances and a measurement error $\varepsilon(t)$ [7]:

$$P(t) = \sum_{i=1}^{N} p_i(t) + \varepsilon(t) \tag{1}$$

Due to the inherent complexity of the problem, the inverse aggregation operation, particularly in cases where individual appliance consumptions are unknown and appliance behaviour is non-deterministic, poses a significant challenge. Traditional and straightforward algorithms may struggle to effectively disaggregate the aggregate signal under these conditions. Consequently, there has been a growing interest among scientists in employing ML algorithms to tackle this demanding task. ML offers a promising approach because it can adapt and learn from data, making it well-suited for capturing various appliances' diverse and often non-linear behaviours [2].

ML approaches for NILM are typically categorized into two primary domains: On/off detection (classification) and instantaneous consumption estimation (regression) [2, 8]. Our particular focus lies within the regression perspective. Among the array of ML algorithms proposed for this purpose, Markov models stand out as a noteworthy choice [2, 9].

Markov models possess the capability to autonomously discern and adapt to temporal dependencies within the data. Consequently, they hold significant promise in effectively addressing the challenges associated with load disaggregation [2, 9]. On the other hand, Artificial Neural Networks (ANNs) play a prominent role in NILM disaggregation [2, 5]. This involvement spans various types of ANNs, ranging from the foundational feedforward ANN to more advanced architectures like Convolutional Neural Networks (CNNs) and context-aware models such as Long Short-Term Memory (LSTM) networks [2]. We
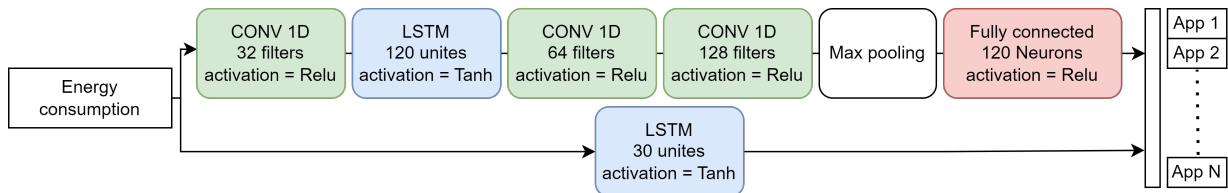
Fig. 1. Proposed multi-target disaggregation neural network [6]

have recently developed a new training scheme for disaggregating energy signals [6] using neural networks. Our approach consists of using a hybrid dual-channel deep neural network. We used 1D CNN along with LSTM layers to disaggregate 11 appliances in a Multi-Target Regression (MTR) framework with a sampling period of 60 seconds. The learning architecture is represented by Figure 1 [6]. Our method resulted in a disaggregation accuracy of 93.27% and a Normalized Root Mean Squared Error (NRMSE) of 0.19 [6]. Furthermore, our model performed better than previous works [6] using the AMPds2 dataset [10] which a reference low-frequency dataset. Nonetheless, a significant limitation of the majority of existing studies is their absence to assess the transferability and generalizability of the developed models to different households [5, 11]. In simpler terms, these studies often train and test their models using data from a single specific building, without considering how well the models might perform in other buildings.

2.2 NILM and transfer learning

In the context of this study, the authors in [5] introduced two distinct methodologies for addressing the concept of transfer learning on a mono-target regression basis for 5 appliances: Model-Agnostic Meta-Learning (MAML) and ensemble learning. In the MAML approach, a singular neural network undergoes pretraining on a meta-dataset and subsequently experiences fine-tuning when exposed to a new dataset. Conversely, in the ensemble approach detailed by the authors in [5], a network is structured as an ensemble of numerous sub-networks, each trained on distinct datasets. The work presented in [11] introduced a fine-tuning procedure using a pure 1D CNN within the framework of multi-target regression, utilizing four publicly available datasets. In these experiments, the number of disaggregated appliances is between 3 and 6. In their study detailed in [12], researchers examined the direct application of pre-trained models to alternate datasets without the need for fine-tuning or additional learning. This evaluation was conducted on three distinct appliances: A microwave, a dishwasher, and a fridge. The study introduced two distinct neural architectures for this purpose, a 1D CNN based architecture and another one consisting of a hybrid network incorporating both 1D CNN and bidirectional Gated Recurrent Units (GRU). Commonly, the mentioned studies [5, 11, 12] have presented intriguing results while encountering certain challenges during the

transfer process. Additionally, the sampling period is the same (around 8 seconds). Lastly, a significant commonality among these studies is the transfer of knowledge to identical appliances, with each output corresponding to the same set of appliances.

Drawing inspiration from prior research and the concept of developing a versatile model, our objectives, using the best model obtained from [6], are as follows:

- To assess the adaptability of our trained model, as documented in [6], when applied to alternative datasets characterized by a notably low sampling frequency of 60 seconds.
- To expand the scope of appliance disaggregation to include a larger number of appliances, exceeding six in total.
- To introduce previously unconsidered (new) appliances into our model.

3   Investigation learning transferability for NILM

The intriguing aspect of transfer learning lies in its ability to enable a pre-trained model to generalize and perform a similar task on different data sources, requiring significantly less data and time [13]. Nevertheless, this necessitates pre-trained models that have been trained on very large multi-source datasets to ensure their optimal performance. To apply this concept to our specific task, we can easily fine-tune a pre-trained model to disaggregate appliance electrical consumption with just a few days of monitoring using plug meters.

In this section, we conduct our experiments and perform fine-tuning on the model originally obtained in [6], but this time we apply it to a different dataset. We utilize the same development environment as described in [6]. We recreate the identical network architecture from the previous work and import the parameters obtained in [6], subsequently initiating the fine-tuning process. Notably, we have switched the optimization algorithm from Adamax to Root Mean Square propagation (RMSprop) [14] which is an optimization algorithm that adapts the learning rates for each parameter during training to improve convergence.

3.1   DB description

Our initial research was conducted using the AMPds2 dataset, officially known as 'The Almanac of Minutely Power dataset 2 (AMPds2) [6, 10]. This dataset is publicly available and comprises two years of aggregated power consumption and load monitoring data collected from a household located in Canada [10]. It includes data from 20 different appliances, all sampled at a 60-second interval. We find this dataset particularly noteworthy due to its unique characteristic of low-frequency sampling and large number of monitored appliances [10]. In the other hand, the Electricity Consumption and Occupancy (ECO) dataset comprises appliance load measurements from six buildings over an eight-month period in Switzerland [15]. These measurements encompass current, voltage, and phase shift data from the three phases, sampled at a one-second frequency [15]. We preprocessed the dataset and calculated the real and reactive power, and then sub-sampled the whole set from 1 second to 60 seconds. Among the six houses we selected the second house because it contains more monitored loads than the others. Table 1 represents the most important characteristics of the used datasets. Note that we didn't consider the stove appliance from ECO dataset because of short monitoring duration. Figure 2 shows a heat map of appliances utilization over 244 days.

Table 1. Description and comparaison of major features for AMPds2 and ECO datasets

	AMPds2	ECO (house2)
Electrical distribution system	Single phase	Three Phases
Sampling period	T=60s	T=1s
Duration	2 years	244 days
Number of selected appliances	11/20	10/11
List of selected appliances	Basement, clothes washer, clothes dryer, dishwasher, fridge, furnace, heat pump, home office, wall oven, television, and hot water	Dishwasher (DWR), air exhaust (AXT), fridge (FRG), kettle (KTL), freezer (FRZ), television (TLV), tablet (TBL), entertainment (ENT), and lamp (LMP)
Selected features	Active and reactive powers	Active and reactive powers
Maximum active power (P)	11,706.0 watts	5,968.6 watts
Minimum active power (P)	0.0 Watts	0.0 Watts
Mean active power (P)	860.3 Watts	210.4 Watts
Standard deviation of active power (P)	826.7 watts	330.3 watts

3.2   Tests and results

We retrained the original model on the original dataset AMPds2 without considering the current (I) because ECO consists of a three-phase system. The results are almost the same with a tiny degradation of around 1%. Then we cloned the resulted model and fine tuned it using a learning rate of 5×10^{-5} using 50 epochs. We considered randomly 20% of our data as training set and another 20% for testing. The Mean Absolute Error (MAE) in equation (2) was chosen to be both a metric to calculate the error and a loss function for the RMSprop optimizer. Along with MAE, we use the Desegregation Accuracy (DA) which indicates how well the model fit the ground truth (equation (3)). Note that N refers to the number of appliances, L is the length of the signal in samples, y_i^j represents the jth sample that belongs to the ith appliance for the real values. $\hat{y}$ refers to the estimated consumption with the same logic as for y.

$$MAE = \frac{1}{N \times L} \sum_{i=1}^{N} \sum_{j=0}^{L-1} \left| y_i^j - \hat{y}_i^j \right| \tag{2}$$

$$DA(\%) = 100 \times \left(1 - \frac{\sum_{i=1}^{N} \sum_{j=0}^{L-1} \left| y_i^j - \hat{y}_i^j \right|}{\sum_{i=1}^{N} \sum_{j=0}^{L-1} y_i^j} \right) \tag{3}$$

In order to investigate our results, we trained another model which has the same architecture but with randomly initialized parameters (weights and biases). Table 2 represents the results for the training of both models: Pre-trained model on AMPds2 (Fine-tuned) and the randomly initialized model (Trained). The columns titles represents the abbreviations of the chosen appliances and the last column is the overall performance. In Figure 3, we can visualize different disaggregation scenarios for three appliances from the test set: fridge, dishwasher, television, entertainment, lamp, and kettle. The area filled in grey
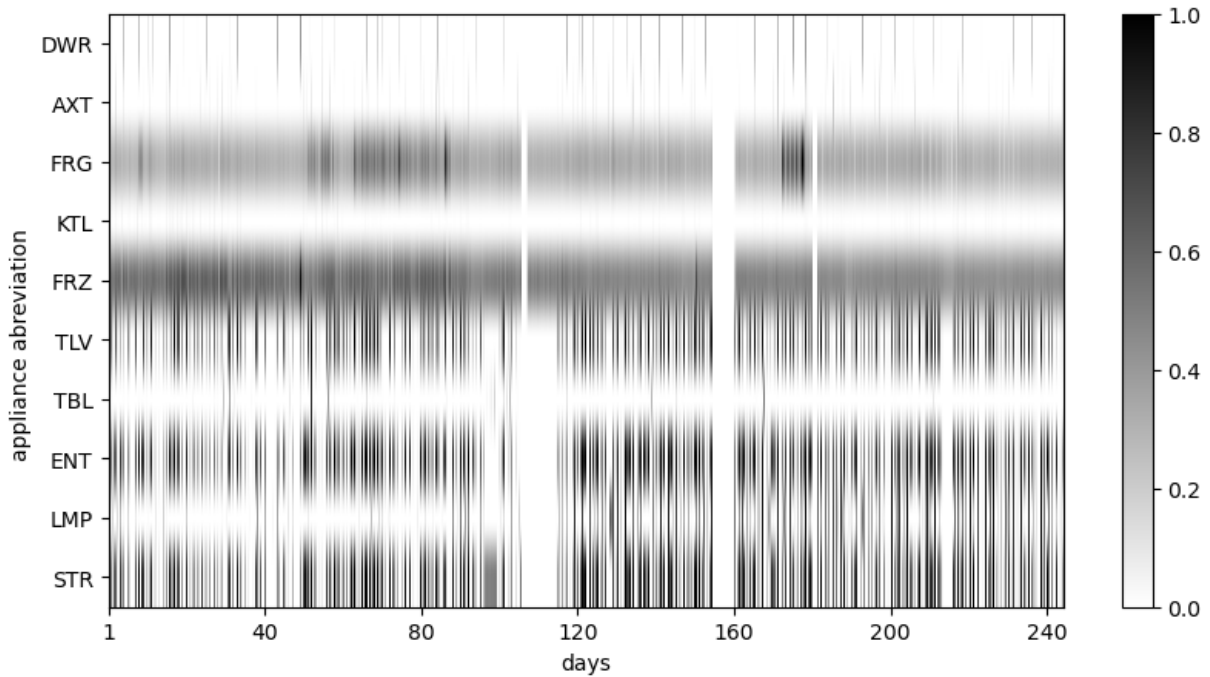
Fig. 2. Appliances' use rate over all monitored days

represents the total household consumption (main meter), the blue dashed line shows the actual consumption for these appliances (ground truth), the green line represents the disaggregation results from the fine-tuned model, and the red line shows the outcomes from the trained model.

4 Discussion and comparison with related works

In general, when we look at the results presented in Table 2 and Figure 3, we can see that the fine-tuned model performs much better than the model trained with random parameters. The fine-tuned model has an overall DA of 87.02% and a mean MAE of 5.56 Watts, whereas the randomly trained model has a DA of 82.58%. The most significant improvement is seen in the dishwasher appliance category. For the fine-tuned model, the DA for dishwashers is 79.67%, while the trained model only achieves a DA of 49.77%. This difference can be explained by the fact that the pre-trained model was exposed to the AMPds2 dataset, which contains more information about dishwasher usage compared

Table 2. Results of both models on the test set from ECO (fine-tuned trained on AMPds2 and fine tuned on ECO)

	model	DWR	AXT	FRG	KTL	FRZ	TLV	TBL	ENT	LMP	STR	ALL
MAE	Fine-tuned	**6.55**	**1.01**	**7.70**	6.25	**5.95**	**5.40**	1.32	**8.39**	**9.23**	**3.83**	**5.56**
	Trained	16.1	1.43	8.34	**6.18**	7.67	8.25	**1.13**	11.60	9.96	3.95	7.47
DA	Fine-tuned	**79.67**	**43.54**	**82.91**	47.09	**87.78**	**93.40**	40.35	**92.68**	**83.88**	**88.37**	**87.02**
	Trained	49.77	20.08	81.48	**48.48**	84.27	89.83	**49.20**	89.88	82.6	87.99	82.58

104
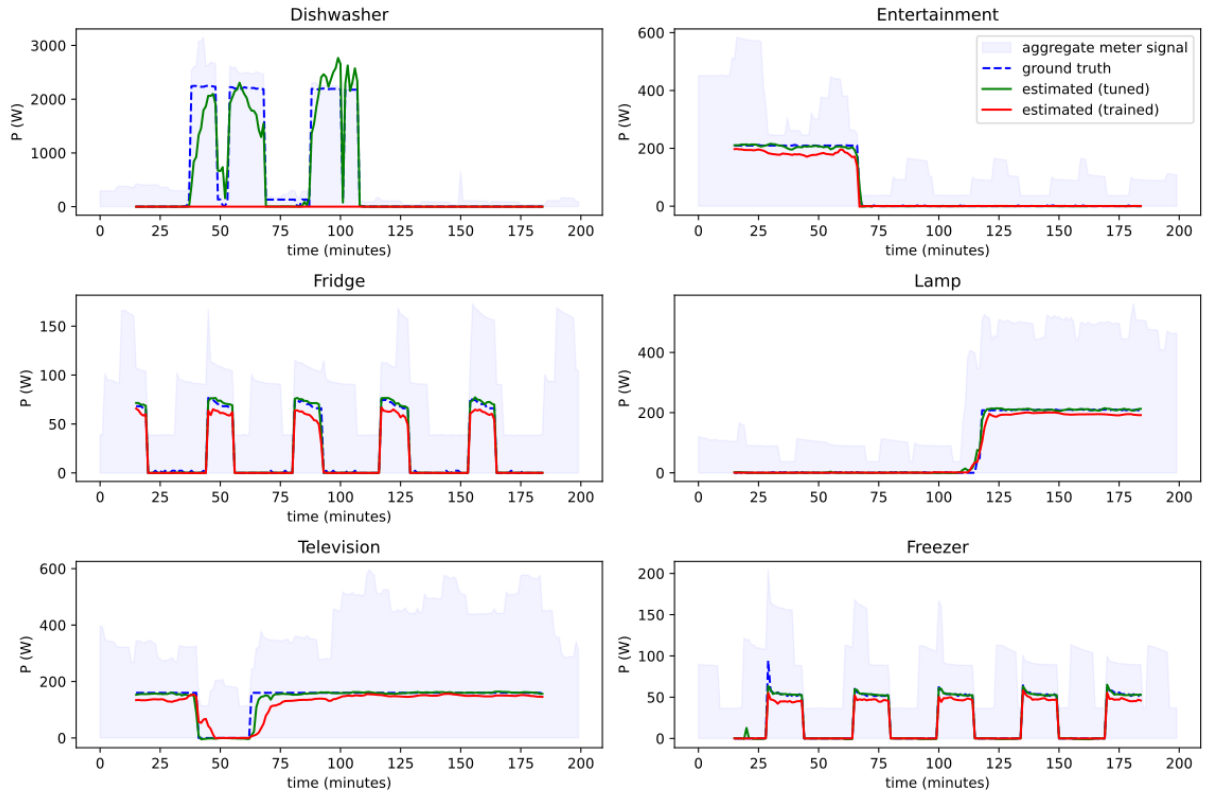
Fig. 3. Disaggregation performance of both models on the test set with respect to the aggregated signal and the real values (refereed to as ground truth)

to the ECO dataset. This exposure allowed the pre-trained model to generalize better, while the trained model struggled to accurately separate dishwasher usage from other appliances. Figure 2 proves our point of view on how this appliance was rarely used.

The datasets have other common appliances, namely the fridge and television. When comparing accuracy, the fine-tuned model performed better, especially for the television appliance, showing a notable increase of approximately 4% in accuracy. However, the fridge appliance's performance was quite similar for both models, with a slight advantage of 1.5% in favour of the fine-tuned model. This result can be attributed to the fact that the Fridge appliance is typically running continuously as we can see in Figure 2, allowing the trained model, even with limited data, to effectively disaggregate this appliance. We have observed intriguing outcomes regarding the entertainment appliance, where the fine-tuned model demonstrated superior disaggregation performance compared to the trained model. Interestingly, even though the exact appliance was absent in the AMPds2 dataset, the home office appliance consumption is much similar to the entertainment appliance. The freezer consumption signature is very similar to the fridge appliance which explains the good performance.

For the remaining appliances, the fine-tuned model consistently outperformed the trained model, except for two appliances: tablet and kettle. Regrettably, both models struggled to effectively disaggregate these two appliances. It is noteworthy that the tablet and kettle appliances were not included in the AMPds2 dataset and their use was rare (Figure 2), which consequently led to the pre-trained model's failure in accurately disaggregating them.

Table 3. Comparison of our work with related works

Work	Sampling period	Target	Number of appliances	Input width	Number of datasets	Transfer method
[5]	8s	mono-target	5	99	2	fine-tuning
[11]	6s	**multi-target**	3-7	599	**3**	fine-tuning
[12]	8s	mono-target	3	variant	**3**	**direct-test**
Our	**60s**	**multi-target**	**10**	**30**	2	fine-tuning

Let us now examine the output performances of both models, as depicted in Figure 3. In the left column, we observe the appliances common to both datasets. Notably, for the dishwasher appliance, the trained model exhibited complete inability to detect it, while the fine-tuned model achieved a partial fit to the actual consumption pattern, as reflected in the accuracy values reported in Table 2. Across the remaining examples, it becomes evident that the fine-tuned model consistently outperforms the trained version by providing a closer match to the ground truth a lower error. Finally, a comparative analysis is conducted to assess the intricacy of our research, the extent of disaggregated appliances considered, and whether the transfer necessitates fine-tuning or can be directly tested. The evaluation of complexity encompasses several critical facets, namely the width of the input window, the choice between multi or mono-target disaggregation, and the sampling period. Based on the information on Table 3, we notice that our method is less complex with a very high sampling period (60s) instead of 8s or 6s [5, 11, 12] with a window of 30 samples only. Our model is a multi-target meaning that we need only one model to disaggregate all appliances simultaneously. The limitation of our work is the number of used datasets where we trained our model on AMPds2 and fine-tuned it on ECO, the final limitation is that we still need fine-tuning meaning we still need some supervised data to apply our task.

5   Conclusion

In conclusion, our research has demonstrated the potential and effectiveness of transfer learning in the context of Non-Intrusive Load Monitoring (NILM). By training a model on the AMPds2 dataset and then transferring this knowledge to the ECO dataset, we achieved remarkable results. Notably, we successfully disaggregated 10 appliances more than any previous multi-target work, achieving an impressive 87.02% accuracy using just 20% of the dataset for training. Comparing our transfer learning approach to a model with randomly initialized parameters, which achieved an accuracy of 82.53%, underscores the significance of pre-trained knowledge in NILM tasks. While our results are promising, there remains room for improvement in the original model to enhance accuracy even further. Our overarching goal is to create a highly generalized model that requires minimal fine-tuning, making it adaptable to a wide range of scenarios. This entails harnessing larger and more diverse datasets to further advance the field of NILM and contribute to energy conservation efforts.

References

1. Alanne, K., Sierla, S.: An overview of machine learning applications for smart buildings. Sustainable Cities and Society **76** (2022) 103445

2. Kaselimi, M., Protopapadakis, E., Voulodimos, A., Doulamis, N., Doulamis, A.: Towards trustworthy energy disaggregation: A review of challenges, methods, and perspectives for non-intrusive load monitoring. Sensors **22**(15) (2022)
3. Zangheri, P., Serrenho, T., Bertoldi, P.: Energy savings from feedback systems: A meta-studies' review. Energies **12**(19) (2019)
4. Weißhaar, D., Held, P., Benyoucef, D., Ould Abdeslam, D., Wira, P., Mercklé, J. In: Smart Meters Improved by NILM. Springer International Publishing (2023) 29–54
5. Wang, L., Mao, S., Wilamowski, B.M., Nelms, R.M.: Pre-trained models for non-intrusive appliance load monitoring. IEEE Transactions on Green Communications and Networking **6**(1) (2022) 56–68
6. Belguermi, Y., Wira, P., Hermann, G.: A multi-output lstm-cnn learning scheme for power disaggregation within a nilm framework. In: 2023 IEEE 21st International Conference on Industrial Informatics (INDIN), Lemgo, Germany (2023) 1–6
7. Hart, G.W.: Nonintrusive appliance load monitoring. Proceedings of the IEEE **80**(12) (1992) 1870–1891
8. Cimen, H., Palacios-Garcia, E.J., Kolaek, M., Cetinkaya, N., Vasquez, J.C., Guerrero, J.M.: Smart-building applications: Deep learning-based, real-time load monitoring. IEEE Industrial Electronics Magazine **15**(2) (2021) 4–15
9. Wu, Z., Wang, C., Peng, W., Liu, W., Zhang, H.: Non-intrusive load monitoring using factorial hidden markov model based on adaptive density peak clustering. Energy and Buildings **244** (2021) 111025
10. Makonin, S.: AMPds2: The Almanac of Minutely Power dataset (Version 2) (2016)
11. Li, D., Li, J., Zeng, X., Stankovic, V., Stankovic, L., Xiao, C., Shi, Q.: Transfer learning for multi-objective non-intrusive load monitoring in smart building. Applied Energy **329** (2023) 120223
12. Murray, D., Stankovic, L., Stankovic, V., Lulic, S., Sladojevic, S.: Transferability of neural network approaches for low-rate energy disaggregation. In: ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Brighton, UK (2019) 8330–8334
13. Vrbančič, G., Podgorelec, V.: Transfer learning with adaptive fine-tuning. IEEE Access **8** (2020) 196197–196211
14. Murphy, K.P.: Probabilistic Machine Learning: An introduction. MIT Press (2023)
15. Beckel, C., Kleiminger, W., Cicchetti, R., Staake, T., Santini, S.: The eco data set and the performance of non-intrusive load monitoring algorithms. In: Proceedings of the 1st ACM Conference on Embedded Systems for Energy-Efficient Buildings (BuildSys '14), New York, NY, USA (2014) 80–89

Applying Machine Learning Algorithms for Classifying Time-Frequency Failures in Power Grid Systems

Gandorj Darambazar[1], Ali Moukadem[1], Bruno Colicchio[1], and Patrice Wira[1]

IRIMAS – Institut de Recherche en Informatique, Mathématiques, Automatique et Signal
IUT de Mulhouse, Université de Haute-Alsace
gandorj.Darambazar@uha.fr, ali.moukadem@uha.fr, bruno.Colicchio@uha.fr,
patrice.wira@uha.fr

Abstract. In power grid systems, Power Quality (PQ) disturbances affect manufacturing process, cause malfunction of equipment and induce economic losses. This paper presents ten new features to identify PQ disturbances such as voltage sag, swell, interruption, harmonics and combined defaults. At first, Hilbert Transform (HT) and Phase Locked Loops (PLL) techniques are applied to estimate the frequency and phase of harmonic components of voltage signals in real time. Then new descriptors, i.e., features, based on Time-Frequency (TF) representations are used. These TF features are obtained from the Rényi and Shannon entropy obtained with the Short-Time Fourier Transform (STFT), the Stockwell Transform (ST) and the Optimized Stockwell Transform (OST). In order to evaluate the proposed TF descriptors, machine learning algorithms are applied to effectively discriminate the different types of disturbances. Classification results show an accuracy of more than 99.4% even in 5 dB SNR high-noise condition.

Keywords: Power quality, power grid failure, time-frequency feature, classification, machine learning.

1 Introduction

In power distribution grids non-linear loads introduce current harmonics and voltage disturbances that affect the performance of other loads [1]. On another side, large penetration of renewable power generation into the existing power grid injects the inevitable issues related to the Power Quality (PQ) [2]. Therefore, it is very important to detect and eliminate the power quality disturbances and failures in order to reach and maintain quality power. So, one of the important issues in PQ problems is to detect and identify disturbance waveforms automatically in an efficient and fast manner. PQ monitoring considers four calculations as it is represented by Figure 1: The phase angle detection, fundamental frequency estimation, amplitude estimation, and frequency content analysis.

There are several techniques for detecting or identifying PQ disturbances [3]: Methods for analysing PQ parameters, filtering methods and Fourier analysis techniques. The most widely adopted approach in signal processing is the spectral analysis using Fourier analysis. This technique is tremendous for analysing stationary signal because the characteristics of the signal do not change with time, but it is unfavourable for non-stationary signals because of its inadequacy in tracking the changes in the magnitude, frequency or phase. The Time-Frequency representation (TF) can be provided by methods like the Stockwell Transform (ST), the Hilbert-Haung Transform (HHT) or either the Wavelet

Transforms [4]. The TF representation gives the energy distribution of a signal according to two variables, the time and the frequency.

   This paper uses the TF analysis for characterizing non-stationary electrical signals and detecting PQ disturbances. This means that PQ disturbances are characterized by ten features in the TF domain. The new relevant parameters allows then to apply Machine Learning (ML) algorithms for classifying PQ failures. PQ disturbance classification is achieved by using ML algorithms such as the K-Nearest Neighbors (K-NN), the Random Forest (RF), and the Support Vector Machine (SVM). These algorithms are trained on the ten selected TF-features extracted from the grid voltage system.
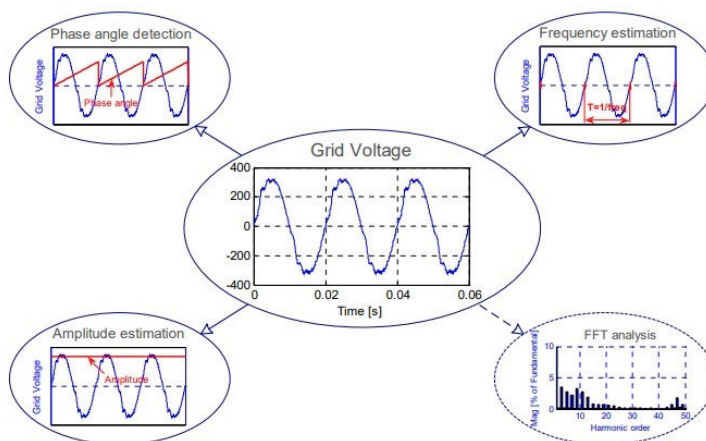
Fig. 1. General overview of power quality monitoring and analysis

2 Time-frequency domain for the evaluation of the power quality

2.1 Context of power quality

The term power quality may be defined as a wide variety of electromagnetic phenomena that characterize the voltage and current at a given time and location in the power system. Voltage swell, sag, flicker, notch, harmonics, interruptions, oscillatory transient and their combinations are some of the common PQ events [1]. Traditionally, the conventional methods to address PQ analysis issues lead to the main descriptors that are the Power Factor (PF), the Total Harmonic Distortion (THD) and the unbalance:

- The PF is the ratio between the active power and the reactive power and is expressed by the $\cos \phi$. This means that if $\cos \phi = 1$, then there is no reactive power flow and the phase angle between the voltage and current is zero.
- The THD is often used to define the degree of harmonic content in an alternating signal. The THD is the ratio of the square root of the sum of all harmonic components except fundamental to the fundamental component. For a signal, the term THD represents the percentage of distortion from its fundamental wave shape. Harmonics for example are more present on the current.
- In a three phase power grid, the voltage unbalance is a condition in which the three-phase voltages differ in amplitude or are displaced from their normal 120° phase

relationship, or both. The degree of unbalance is usually defined by the ratio of the negative sequence voltage component to the positive sequence component. In power grids with high penetration of solar PV generators, they are typically not allocated equally between the three phases.

Theses indicators are simple PQ descriptors and are not representative of the quality of the waveforms of a power supply system, i.e., the voltage and the current. In other words, they are not able to distinguish the previously listed common PQ events, i.e., they are not able to calculate the parameters of the waveforms.

These events occur on the voltage or on the current signals and affect their waveforms. In this study, the objective is not to calculate the parameters of the waveforms but to identify the PQ events. Our choice consists in using TF-descriptors because they are able to handle non-stationary signals which are more common rather than stationary signals and to take into account frequencies variations over time which is very relevant to discriminate PQ faults [5].

2.2 System modelling and fault description

In this study, the power grid is defined by the two following signals:

$$u(kT_s) = U \sin(2\pi f kT_s + p) \tag{1}$$

$$i(kT_s) = I \sin(2\pi f kT_s + p + \theta) \tag{2}$$

where $u(t)$ and $i(t)$ are respectively the voltage and the current of amplitude U and I. f is the fundamental frequency ($\omega = 2\pi f$), p is the phase angle and θ is the phase angle between the voltage and the current. T_s represents the sampling period and k is the time index or the iteration because the power system is considered as a full discrete system.

Voltage and current waveforms which are supposed to be a periodic (sinusoidal) waveforms may contain PQ event/events. This paper classifies 10 different situations of the PQ. These are the standard and pure waveform (C_0) corresponding to (1) or (2), and 9 conditions with the voltage with PQ disturbances like sag (C_1), swell (C_2), interruption (C_3), sag + harmonic 3 (C_4), sag + harmonic 3 + harmonic 5 (C_5), swell + harmonic 3 (C_6), swell + harmonic 3 + harmonic 5 (C_7), harmonic 3 (C_8), and harmonic 3 + harmonic 5 (C_9). The PQ events are fully formalized by mathematical expressions in Table 1. These situations of the PQ are refered as classes C_i with $i = 0...9$ and where the considered signal is $x_i(kT_s)$ with $h(kT_s, \tau)$ the Heaviside step function defined as (T is an instant):

$$h(kT_s, \tau) = \begin{cases} 1 & \text{if } kT_s \geq \tau \\ 0 & \text{if } kT_s < \tau \end{cases} \tag{3}$$

In the following, all the developments and proposed methods can indifferently handle the voltage or the current measured on the power grid.

2.3 Proposed TF features

A complete detection and classification scheme of the PQ disturbances is proposed. An overview of this scheme is represented by Figure 2. The following three main steps are considered: A - the detection processing, B - the TF-feature extraction processing, and C - the classification processing.

Table 1. Description of the PQ events

PQ disturbance	signal	parameters values
C_0: Pure sine wave	$x_0(kT_s) = A\sin(\omega kT_s)$	$A = 1, f = 60\text{Hz}, \omega = 2\pi f$
C_1: Sag	$x_1(kT_s) = A\big(1 - \alpha h(kT_s, T_1) + \alpha h(kT_s, T_2)\big)\sin(\omega kT_s)$	$0.1 \le \alpha \le 0.9$
C_2: Swell	$x_1(kT_s) = A\big(1 + \alpha h(kT_s, T_1) - \alpha h(kT_s, T_2)\big)\sin(\omega kT_s)$	$0.1 \le \alpha \le 0.8$
C_3 : Interruption	$x_3(kT_s) = A\big(1 - \alpha h(kT_s, T_1) + \alpha h(kT_s, T_2)\big)\sin(\omega kT_s)$	$0.9 \le \alpha \le 1,$ $T \le t_2 - t_1 \le 9T$
C_4: Sag + Harmonic 3	$x_4(kT_s) = A\Big(1 - \alpha h(kT_s, T_1) + \alpha h(kT_s, T_s)\big)\sin(\omega kT_s) + \alpha_3 \sin(3\omega kT_s)\Big)$	$0.1 \le \alpha \le 0.9,\ 0.05 \le \alpha_3 \le 0.15,\ \sum \alpha_i^2 = 1$
C_5: Sag5 + Harmonics 3 &	$x_5(kT_s) = A\Big(1 - \alpha h(kT_s, T_1) + \alpha h(kT_s, T_2)\big)\sin(\omega kT_s) + \alpha_3 \sin(3\omega kT_s) + \alpha_5 \sin(5\omega kT_s)\Big)$	$0.1 \le \alpha \le 0.9,\ 0.05 \le \alpha_3, \alpha_5 \le 0.15,\ \sum \alpha_i^2 = 1$
C_6: Swell + Harmonic 3	$x_6(kT_s) = A\Big(1 + \alpha h(kT_s, T_1) - \alpha h(kT_s, T_2)\big)\sin(\omega kT_s) + \alpha_3 \sin(3\omega kT_s)\Big)$	$0.1 \le \alpha \le 0.8,\ 0.05 \le \alpha_3 \le 0.15,\ \sum \alpha_i^2 = 1$
C_7: Swell + Harmonics 3 & 5	$x_7(kT_s) = A\Big(1 + \alpha h(kT_s, T_1) - \alpha h(kT_s, T_2)\big)\sin(\omega kT_s) + \alpha_3 \sin(3\omega kT_s) + \alpha_5 \sin(5\omega kT_s)\Big)$	$0.1 \le \alpha \le 0.8,\ 0.05 \le \alpha_3, \alpha_5 \le 0.15,\ \sum \alpha_i^2 = 1$
C_8: Harmonic 3	$x_8(kT_s) = A\big(\sin(\omega kT_s) + \alpha_3 \sin(3\omega kT_s)\big)$	$0.05 \le \alpha_3 \le 0.15,$ $\sum \alpha_i^2 = 1$
C_9: Harmonics 3 & 5	$x_9(kT_s) = A\big(\sin(\omega kT_s) + \alpha_3 \sin(3\omega kT_s) + \alpha_5 \sin(5\omega kT_s)\big)$	$0.05 \le \alpha_3, \alpha_5 \le 0.15,$ $\sum \alpha_i^2 = 1$

111
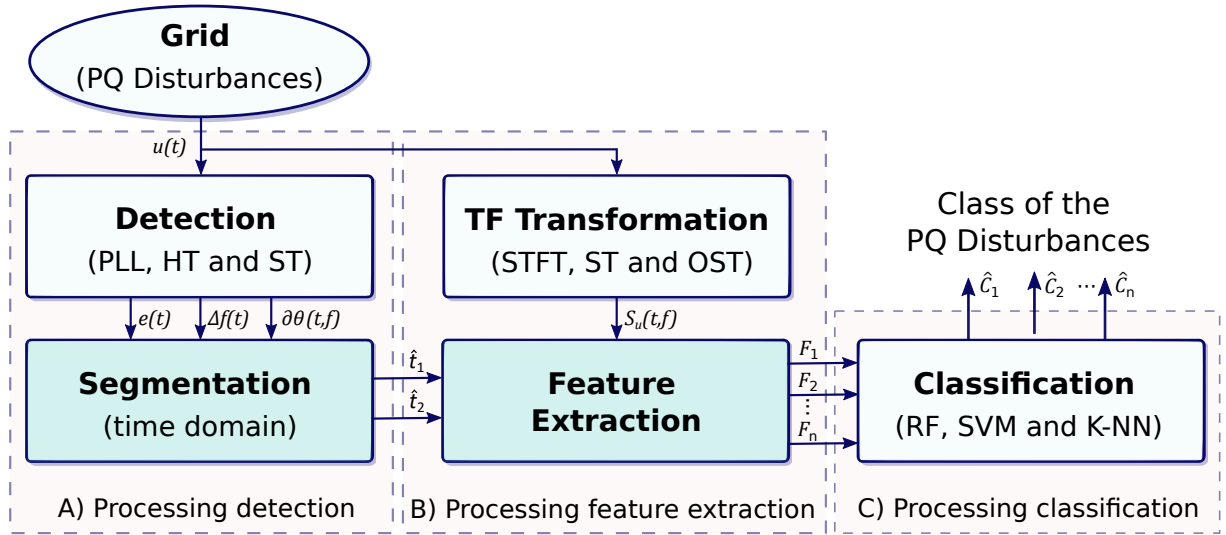
Fig. 2. Proposed workflow for the detection and classification of power quality disturbances

A) In the detection processing, with a signal measure x from the grid, the following values are calculated by the detection block: e, Δf and $\partial \theta$ depend on the time. They are:

$$e = sin(\theta - \hat{\theta}) \approx \theta - \hat{\theta}, \tag{4}$$

$$\Delta f = f - \hat{f}, \tag{5}$$

$$\partial \theta(kT_s, \hat{f}) = \frac{\partial \Phi_x^w(kT_s, \hat{f})}{\partial kT_s}. \tag{6}$$

where

- e which represents the phase angle deviation;
- Δf which is the instantaneous frequency deviation compared to the constant frequency;
- $\hat{f}$ is the estimated frequency and $\hat{\theta}$ the estimated phase angle at instant k; This can be achieved in real-time by PLL techniques [6] or more sophisticated techniques or signal processing tools like the Hilbert Transform (HT) and the Stockwell Transform (ST);
- $\partial \hat{\theta}(kT_s, \hat{f})$ is the estimated value of the instantanneous phase with respect to time;
- $w(kT_s)$ is the sliding Gaussian widow.

The segmentation steps consists in estimating instants t_1 and t_2, i.e., in calculating $\hat{t}_1 = k_1 T_s$ and $\hat{t}_2 = k_2 T_s$ in order to robustify the TF features. Indeed, $\hat{t}_1$ represents the start time and $\hat{t}_2$ represents the end time of the event. This allows to handle high quality descriptors in the TF-frame. They are thus relevant and significant in the TF frame even in the case of non-stationary signals measured from the power grid. New and recent techniques have been developed to exceed system performance traditional, in terms of image quality, size, weight, energy consumption [7].

B) The "TF Transformation" uses TF-methods like the ST, the Short-Time Fourier Transform (STFT), or the Optimized Stockwell Transform (OST). These methods outputs $S_u(kT_s, \hat{f})$ which is the TF representation of the signal u at a given instant [8]. It is a $n \times m$ matrix that contains the frequency content of the signal over the last period of time.

The "Feature Extraction" block consists in using $S_u(t, f)$ from whom $\hat{t}_1$ and $\hat{t}_2$ serves to trigger the calculation of ten TF-descriptors. These ten TF-descriptors are called features F_i (with $i = 1...10$) and are:

- $F_1 = Nf(t)$ is the number of components as a function of time. Unlike conventional features designed to evaluate harmonic distortion such as the Total Harmonic Distortion (THD), F_1 aims to track the number of harmonics and allows to know precisely when any harmonic component may appear or disappear.
- $F_2 = SER$ is the Shannon entropy ratio obtained by the STFT.
- $F_3 = CM$ is the energy concentration obtained by the STFT, it is a statistical measurement to assess the quality of time-frequency representation. It is often used to evaluate the degree of energy dispersion around the instantaneous signal frequency in the time-frequency plane.
- F_4, F_5 and F_6 are transient features based on the Shannon energy. They characterize the shape of the transient signals at specific frequencies, i.e., 60, 180 and 300 Hz.
- F_7, F_8, F_9 and F_{10} are the parameters that control the OST's Gaussian window. They are obtained by maximizing the energy concentration F_3 [7].

Obviously, some of these TF-descriptors have already been used in the literature to describe the PQ disturbances. The new TF-descriptors that have been introduced in this study to enhance the significativity of the PQ under non-stationary conditions are F_1 and F_2 [9].

C) The classification processing will be detailed in the next section. It is based on three ML techniques that only take into account the 10 features.

3 TF-failures classification with machine learning techniques

The sources of distortion in waveforms are present at generation, transmission and distribution parts of the power grid. For better processing and identification of the PQ events, features are extracted from the raw signal. The feature set extracted is used as an input to the classifier system which classifies the disturbance. Traditional PQ disturbance classification methods are susceptible to noise interference, transients, time-varying events, feature selection and feature quality. The quality of the feature, also referred as veracity, is related to accuracy, biases, noise, and abnormality in signals. Therefore classification of the PQ disturbances requires advanced algorithms such as Artificial Neural Networks (ANN) and other ML techniques [14–18].

This paper proposes a PQ disturbance classification based on algorithms such as the K-Nearest Neighbors (K-NN), the Random Forest (RF), and the Support Vector Machine (SVM). They have been chosen because of their simplicity. If the features are relevant, the classification problem can be simplified. These algorithms are trained on the ten TF-features F_i previously defined. The PQ classification consists in recovering the class $\hat{C}_i$ that characterises waveform distortions and deviations from the ideal sine wave. The waveform patterns are the ones expressed by signals $x_i(kT_s)$ corresponding to a class C_i of a PQ event or a specific distortion of the pure sine waveform.

3.1 Dataset description

The sampling rate is $T_s = 1$ ms ($f_s = 1$ kHz) and without any loss of generality $f = 1/T = 60$ Hz. The amplitude of the pure sine wave is chosen as $A = 1$. A total of 1000 PQ events have been generated with 100 different signals for each of the 10 C_i classes.

For each class, the numerical values of parameter α have been uniformly chosen between the limits given in Table 1. The duration of the signals is 1 second and the duration of the failure (PQ event) within a signal has a duration of 80 μs. For each situations $T \leq T_1 \leq T_2 \leq 9T$ and $T \leq T_2 - T_1 \leq 9T$ i.e., $0.016667 \leq T_2 - T_1 \leq 0.1500$ seconds. Each PQ situation related to the classes C_i has been generated 100 times with different levels of Gaussian noise: No noise, 20 dB and 5 dB noises.

Then, the features are calculated for all the signals with the tree methods (STFT, ST and OSTO). These data represents a database composed of 300 000 data. for the learning process of the ML algorithms and 75% of the data has been randomly selected and used for training the algorithms and 25% for the tests.

3.2 Design of the ML classifiers

The K-NN is one of the simplest of classification algorithms that is often used as a benchmark for more complex classifiers. It is a supervised learning that does not make any assumptions on the underlying data distribution.The K-NN has been implemented with the Euclidean distance as the distance metric in the multidimensional feature space and a neighbourhood size $K = 1$. The numerical value of K has been obtained by applying a cross validation over the data.

The RF is made up of multiple decision trees which are common supervised learning algorithms. Their predictions are aggregated to identify the most popular result. It is an ensemble method. The implementation of the RF uses the following configuration: It is composed of 30 Bagged Trees with a maximum number of splits of 1025.

SVM works by mapping data to a high-dimensional feature space so that data points can be categorized, even when the data are not otherwise linearly separable. A separator between the categories is found, then the data are transformed in such a way that the separator could be drawn as a hyperplane. The mathematical function used for the transformation is known as the kernel function. The SVM has been designed with the followings: The Kernel function is Gaussian with a scale of 0.79, a box constraint level of 1 and all the data are normalized.

4 Evaluation of the PQ event classification and results

All the calculation and simulation have been conducted by using the Matlab environment. A primary result is presented in Figure 3 where two signals are mapped in the two dimensional TF-space by using the three methods previously described: STFT, ST and OST. It can be seen that for the disturbed signals, with only a sag, and with a sag with additional harmonic components of rank 3 and 5, that it relatively easy to detect the start and end times of the fault. From here, $\hat{t}_1$ and $\hat{t}_2$ are estimated from the image, i.e., energy represented in the TF-space. Once these instants have been determined, the 10 TF-descriptors F_i can be calculated. One can also see for example, that during the period (between $\hat{t}_1$ and $\hat{t}_2$) the presence of 2 higher-order harmonics in the second signal (i.e., $\hat{F}_1 = 3$).

Figure 4 shows the resulting classification in the 2-dimensional frame of $\hat{F}_1$ vs $\hat{F}_2$ for the three types of signals: Without noise, with a 20 dB noise and with a 5 dB noise. $\hat{F}_1$ and $\hat{F}_2$ are obtained with the OST method. Each dot represents a PQ event and its class is color-coded. The extracted features separate all the classes successfully. It can be seen that it is more difficult to separates the classes of PQ events in noisy signals. Without noise, classes can be well separated event with simple ML techniques. However, one must
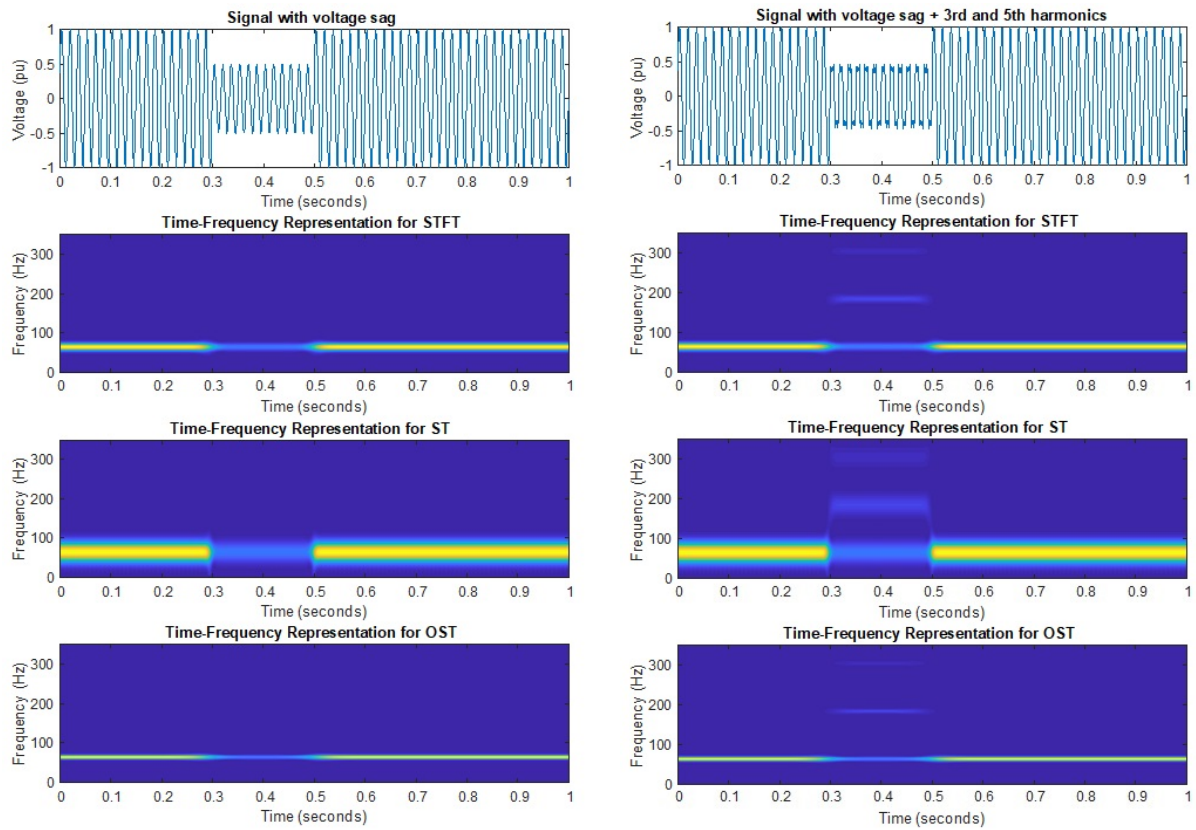
Fig. 3. TF-analysis of a signal with sag (left) and a signal with sag but also disturbed by harmonic components of rank 3 and 5 (right)
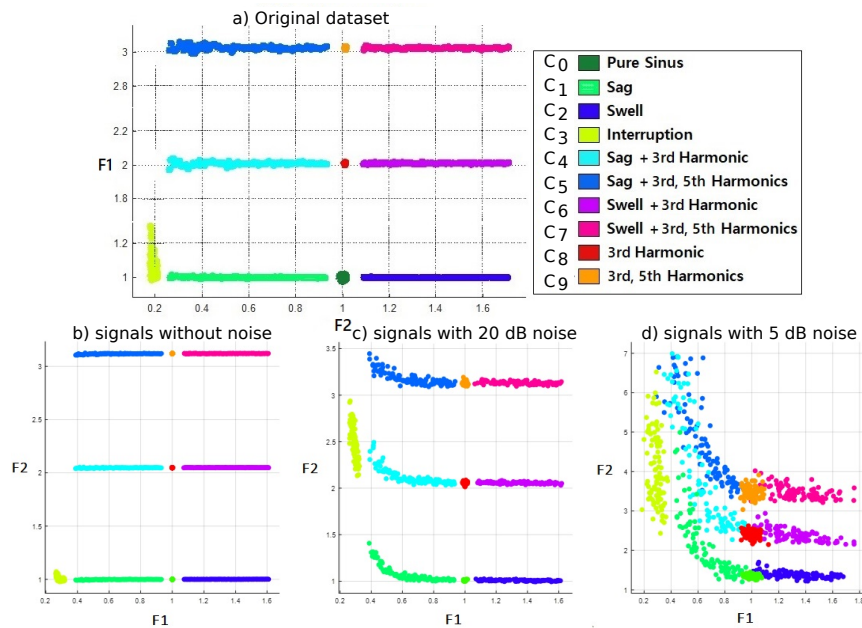
Fig. 4. Representation of the classified PQ events in the $\hat{F}_1\hat{F}_2$-frame with different levels of noise by using the OST method
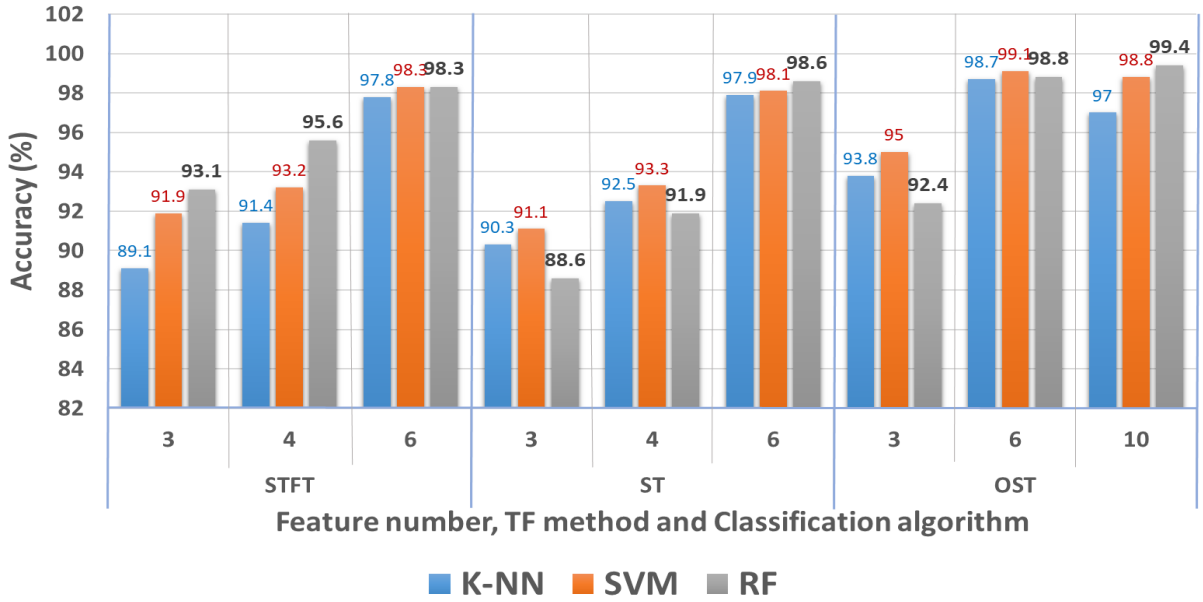
Fig. 5. Overview of the PQ events classification performance

remain that the classification was achieved in a 10-dimensional space, it is thus not easy to see the decision boundaries of all the classes.

Finally, a complete analysis of the PQ events classification performance is presented by Figure 5. Results show the ML-classification performances by taking into account only the 3 first, 4 first, 6 first or all the 10 $\hat{F}_i$ features by using the 3 different TF methods. The RF algorithm leads to a classification rate higher than 99,4% in each situations of PQ disturbances and noise. K-NN and SVM algorithms reaches a classification rate higher than 97% under the same test conditions with 10 features. In some situation, the classification accuracy is 100%. This is the case when the SNR is high. It can be concluded that the proposed descriptors are able for effectively discriminate the different types of disturbances even in the presence of noise.

5   Conclusions

The issue of power quality in the power system is of great importance for the smooth and long-lasting operation of the electrical devices. It is very important to detect and eliminate the power quality disturbances in order to obtain quality power. This paper presents a work to evaluate the classification of PQ events in realistic power distribution systems. The goal consists in handling non-stationary disturbed signals which are ubiquitous in a large number of power grids. Ten time-frequency descriptors have been proposed by using three different time-frequency methods: The Short-Time Fourier Transform (STFT), the Stockwell Transform (ST) and the Optimized Stockwell Transform (OST). Once calculated, the ten are gathered to serves as inputs to different machine learning techniques in order to identify and classify the PQ events even under noise conditions. For this, simple machine learning techniques have been used and evaluated. Indeed, three classifiers, such as K-Nearest Neighbors (K-NN), Support Vector Machine (SVM), and Random Forest (RF) are utilized to recognize the PQ events categories. The results showed that the time-frequency features are relevant to estimate physics-related meaningful parameters of the noisy and non-stationary signals and thus to better understand the origin of the

failures. Furthermore, the complete approach is new and effective in classifying with a high accuracy different varieties of PQ events occurring in power grids.

References

1. Bollen, M.H.J.: Understanding Power Quality Problems. John Wiley and Sons Inc. (1999)
2. Kumar, D., Padhy, B.P.: Probabilistic approach to investigate the impact of distributed generation on voltage deviation in distribution system. Electrical Engineering **105**(5) (2023) 2621–2636
3. Abidullah, N.A., Abdullah, A.R., Sha'ameri, A., Shamsudin, N., Ahmad, N., Jopri, M.: Real-Time Power Quality Disturbances Detection and Classification System. World Applied Sciences Journal **32**(8) (jan 2014) 1637–1651
4. Hlawatsch, F., Auger, F.: Time-Frequency Analysis, Concepts and Methods. ISTE and Wiley (2008)
5. Moukadem, A., Bouguila, Z., Abdeslam, D.O., Dieterlen, A.: A new optimized Stockwell transform applied on synthetic and real non-stationary signals. Digital Signal Processing **46** (nov 2015) 226–238
6. Darambazar, G., Colicchio, B., Moukadem, A., Wira, P.: A comparison of pll for online frequency tracking in power grids. In: 30th IEEE International Symposium on Industrial Electronics (ISIE 2021). DOI: 10.1109/ISIE45552.2021.9576189.
7. Drouaz, M., Colicchio, B., Moukadem, A., Dieterlen, A., Ould-Abdeslam, D.: New Time-Frequency Transient Features for Nonintrusive Load Monitoring. Energies **14**(5) (2021) 1437
8. Darambazar, G., Moukadem, A., Colicchio, B., Wira, P.: Entropy measures applied on time-frequency domain for detection and identification of power quality disturbances. In: IEEE 20th International Conference on Harmonics and Quality of Power (ICHQP 2022)
9. Darambazar, G., Moukadem, A., Colicchio, B., Wira, P.: Entropy measures applied on time-frequency domain for detection and identification of power quality disturbances. Electrical Engineering (submitted, 2023)
10. Sejdic, E., Djurovic, I., Jiang, J.: Time–frequency feature representation using energy concentration: An overview of recent advances. Digital Signal Processing **19**(1) (2009) 153–183
11. Sucic, V., Saulig, N., Boashash, B.: Estimating the number of components of a multicomponent nonstationary signal using the short-term time-frequency Rényi entropy. EURASIP Journal on Advances in Signal Processing **2011**(1) (dec 2011) 125
12. Darambazar, G., Moukadem, A., Colicchio, B., Wira, P.: Entropy measures applied on time-frequency domain for detection and identification of power quality disturbances. In: 20th International Conference on Harmonics and Quality of Power (ICHQP), Naples, Italy (jun 2022)
13. Stockwell, R., Mansinha, L., Lowe, R.: Localisation of the complex spectrum: The S Transform. IEEE Transaction on Signal Processing **44**(4) (apr 1996) 998–1001
14. Biswal, M., Dash, P.K.: Detection and characterization of multiple power quality disturbances with a fast S-transform and decision tree based classifier. Digital Signal Processing **23**(4) (jul 2013) 1071–1083
15. Valtierra-Rodriguez, M., de Jesus Romero-Troncoso, R., Osornio-Rios, R., Garcia-Perez, A.: Detection and classification of single and combined power quality disturbances using neural networks. IEEE Transactions on Industrial Electronics **61**(5) (2014)
16. Panigrahi, B., Dash, P., Reddy, J.: Hybrid signal processing and machine intelligence techniques for detection, quantification and classification of power quality disturbances. Engineering Applications of Artificial Intelligence **22**(3) (2009) 442–454
17. Eristi, B., Eristi, H.: A new deep learning method for the classification of power quality disturbances in hybrid power system. Electrical Engineering **2022** (jun 2022)
18. Oliveira, R.A.d., Bollen, M.H.J.: Deep learning for power quality. Electric Power Systems Research **214** (2023) 108887

AI-Based CT Data Pipeline

Robin Tenscher-Philipp[1], Tim Schanz[2], Martin Simon[3]

[1] Robin Tenscher-Phlipp (M.Sc)
robin.tenscher-philipp@h-ka.de
[2] Tim Schanz (M.Sc)
tim.schanz@h-ka.de
[3] Martin Simon (Prof. Dr.-Ing.)
martin.simon@h-ka.de

Hochschule Karlsruhe - Technik und Wirtschaft
University of Applied Sciences
Fakultät für Maschinenbau und Mechatronik
Moltkestr. 30
76133 Karlsruhe

Abstract. Data generation plays an increasingly important and crucial role in training artificial intelligence (AI) models. Focusing on the field of 3D computed tomography, for example, there is often a lack of available data. To address this problem, we propose an AI-based solution to generate training data by adding artificial defects to CT data. Our method enables the generation of large amounts of realistic defect-laden parts that can be used as training data for AI applications. By automating the process and adjusting the parameters, we can generate different defect types and distributions. To evaluate the generated results, this work trains a segmentation AI and applies it to unseen real-world data. This approach closes the gap in the availability of training data and enables the industry to use AI technology effectively.

Keywords: Data Generation, Sparse Data, Artificial Intelligence, Deep Learning, Industrial Computed Tomography, Defect Analysis, Autoencoder, 3D Segmentation

1 Introduction

In the realm of technological innovation, Artificial Intelligence (AI) and Deep Learning (DL) have emerged as transformative forces. However, their voracious appetite for data presents a formidable challenge that spans across diverse applications. Our focus centers on addressing this challenge within the domain of industrial computed tomography (CT) data analysis, where data scarcity has long hindered progress. This paper introduces an AI data-driven pipeline that leverages a single CT scan of an industrial additively manufactured metal component containing internal defects as its foundational data source. This pipeline comprises a sequence of steps, each crucial for reshaping the landscape of data-driven AI applications.

This research represents an important contribution in the domain of data provision for AI and DL applications, offering a novel solution to the persistent challenge of data scarcity. Our ability to generate diverse data from minimal inputs has the potential to improve significantly industrial CT data analysis and numerous other fields reliant on data-intensive AI models. One of the most challenging one is 3D-CT-Data

2 Related work

The current state of synthetic data generation research encompasses three major approaches: algorithmic data generation, simulation, and AI-based methods. It involves a diverse array of techniques and algorithms. Classical algorithmic data generation relies on randomness as its core principle. In this case, the data can be generated directly by algorithmic rules that are fed with random numbers to increase variation [1]. The ranges of random number generation must be parameterized for this purpose. For this, a deep understanding of the characteristics of the target data is necessary to set up and parameterize the corresponding rules. Depending on the complexity of the data to be generated, in this work industrial CT data, an algorithmic generation can become very complex, which can affect the realism as well as the variance of the data. Another way of generating data is simulation [2, 3]. Here, the real generation process is digitally recreated. Depending on the effort and the application, a simulation can provide real data close to reality. In combination with a controllable random generator, a lot of variant data can be generated. This generation can be used for industrial CT data. With the use of deep learning (DL) techniques, in particular autoencoders (AE) [4] and generative adversarial networks (GAN) [5], which can be used in many different applications. Generative networks are mainly used to generate data [6–11]. Compared to previous methods, DL has the advantage that artificial neural networks (ANN) can learn what the data should look like by using training data. The choice of training data is important. They should be sufficiently numerous and reflect all necessary real features as well as possible through their variance. In this way, good training conditions can be created in which the ANN learns the patterns and structures of the training data and can thus generate new data. Our approach of an AI-based generation of industrial CT data including defects is described below in this paper. The state of the art in data generation is an ever-evolving field of research. The field is constantly evolving and expanding, particularly with the advent of artificial intelligence methods and the data they require. The choice of a suitable data generation method depends on the application, the required know-how and the available hardware.

The landscape of segmentation in recent years has witnessed a significant transformation. Traditional methods, while still relevant in many applications, are progressively giving way to AI-driven approaches. Among these classic techniques is thresholding, which relies on variations in image intensity to isolate features within a specified intensity range. Its effectiveness hinges on features having distinct intensity values. When other features share the same intensity range, thresholding becomes incapable of distinguishing between them. For scenarios where features vary over time and require contextual information for recognition, conventional algorithmic techniques, such as edge detection, exhibit limitations due to their rigid rules. Deep Learning (DL) emerges as a game-changer in such situations. Through training ANNs acquire the ability to discern features within their natural context. When the circumstances change, the ANNs benefit from their ability to abstract and are still able to recognize the desired features. Important ANN architectures for segmentation are UNet [12, 13] based architectures. The basis of these architectures is their U-shaped structure based on encoders and decoders. The encoder part consists of convolutional layers that are used to extract the features. The decoder recombines the information in the ascending path and converts it into a representation. In addition to the AE, the UNet architecture adds skip links that connect the encoder and decoder in different layers. This links information in different layers to give a more accurate result. The UNet architecture is the basis for many other derivatives such as VNet [14] or PCUNet [15]. All these derivatives have different advantages depending on the data. Additionally, these architectures nowadays get enhanced using attention mechanisms [16] build in Transformer models like UNETR [17, 18].

3 Methodology

For an AI-based pipeline to generate data in the case of 3D industrial CT data of components with internal defects, the following pipeline Fig. 1 was developed.

Fig. 1. Pipeline from CT scan to image segmentation

(1) CT scan: The first step is to extract the internal defects from a real scanned component. The scanned part is available in voxel format. To extract the defects, the part is converted to STL format, which represents the surface geometry of the part. This step allows the extraction of the internal defects which are saved as individual separate STL files.

(2) Train Autoencoder: The second step involves creating the core of the defect generation process, an AE architecture as shown in Fig. 2. To work with the AE, the separate STL defects from the first step are transformed back into a constant-sized 25^3 voxel format. With this input data, the AE learns by encoding defects, mapping them to a compressed representation, and then decoding them to recreate defects in the voxel format.
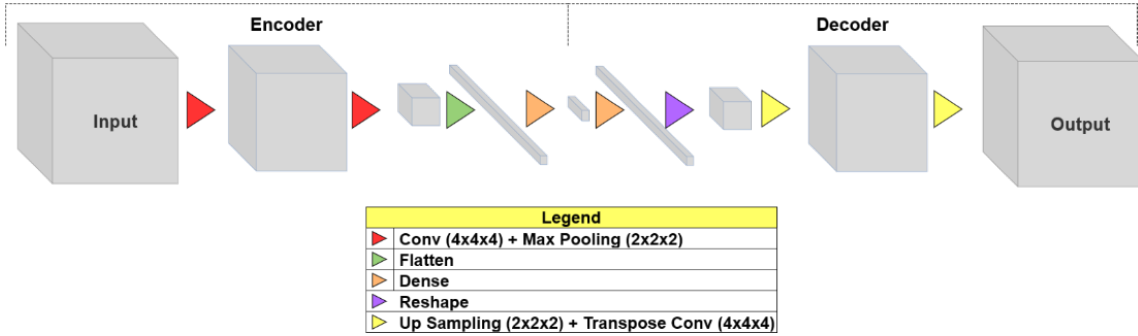
Fig. 2. Schematic of an AE architecture.

(3) Defect Generation: In the third stage, the trained AE is used to generate defects. This is done by separating the encoder part from the AE. By specifically varying the code layer, individually defined defects with desired geometric properties can be generated. To feed the correct activations into the code layer, individual properties are specifically analyzed, resulting in a parameter space that produces possible real generations. The larger the code layer, the more precisely individual properties can be mapped and set in the generation. Depending on the training quality of the auto-encoder, an infinite number of defects can be generated.

(4) Defect Placement: In the fourth step, the generated data is converted to STL format and placed into a desired STL object. The placement of defects in objects or components is done using algorithms based on properties such as number of defects, defect size and intersection. In this way, many variations of a component with different defect characteristics, numbers and distributions are generated and processed in the next step.

(5) CT-Simulation: In this work, the application and plausibility of the generation pipeline is evaluated using industrial CT data of components with inner defects. The CT simulation software ARTIST is used to generate the artificial CT data. Realistic simulation setups were created for this purpose. The software was automated using a script. The challenge lies in the parameterization of the software. Realistic setups require knowledge of X-ray physics. In a further step, the CT projections are reconstructed into a 3D image using the CT reconstruction

software Siemens Cera [19]. By varying the simulation and reconstruction parameters, the diversity of the resulting images increases. As a result of step 5 of the step chain, artificially realistic CT images with internal defects are created. Additionally, ground truth is needed for the next phase, and to obtain this, the scans and reconstructions are idealized. This involves simulating a defect-free and defective component to generate binary ground truth through subtraction and thresholding. These label maps serve as training data for all variations of artificial CT data since defect positions remain consistent between ideal and realistic simulations.

(6) AI Segmentation: In the last step of the data generation pipeline, artificial neural network models, such as convolutional neural networks (CNNs), are trained with the synthetic 3D data. The 3D voxel data from (5) were sliced into smaller chunks with a resolution of 128^3 voxels to avoid hardware limitations. In this work a UNet architecture is used. Since this is a binary segmentation, the sigmoid function is used as the output function. The binary cross entropy is used as the loss. The performance of the segmentation is monitored using the metric Binary Intersection over Union (BIoU). After training, the model is used to segment real CT scans of components, e.g. for quality control in manufacturing. Manual annotation can then be omitted.
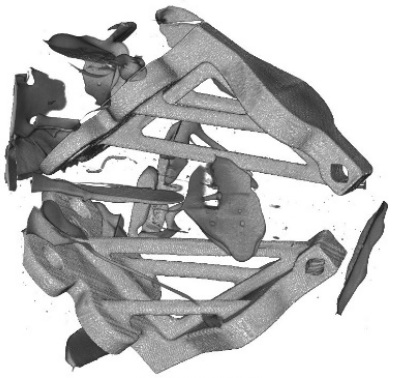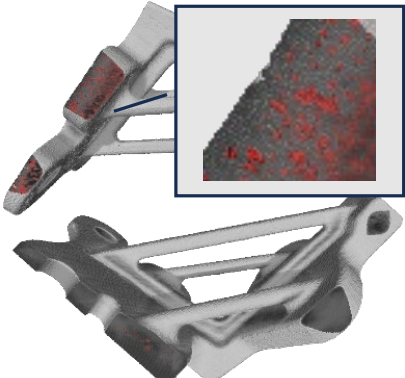
4 Results

In this paper we propose a pipeline that can synthetically generate single part features or whole objects. In our use case, we specialize the pipeline on defective industrial CT data. This pipeline can extract the information for the complete workflow from a single real image, train the AI and generate as much data as desired. The following are results of the pipeline and individual sub-points that are important and necessary to generate the desired synthetic data. The pipeline is developed as python modules for each step except for the simulation and reconstruction which makes it easy to automate the processes.

4.1 Initial CT Data, STL Component and Defect extraction

The first step in the Pipeline is to create a CT scan of the component (Table 1, left) as baseline. Only one scan is required for the whole pipeline which is done on a CT machine. The projection data is then reconstructed to a voxel 3D volume (Table 1, center). In this case two hock parts were scanned together using additional support material to fix the parts during the scan procedure. Marching cube [20] algorithm techniques allow to extract the material and defects surfaces depending on a threshold and a voxel resolution (Table 1, right). Additionally, the support material could be eliminated on STL level. By separating the hull as STL component it can be later used to place synthetic defect variations in it. The real defects itself are separated and stored one by one in an STL file as well. Those are the baseline for synthetic defect generation. Around 40,000 defects could be extracted from the component. By applying a filter to retrieve only watertight structures of a certain size around 25,000 defects are useful for further processing.
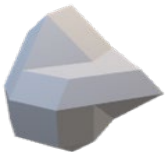
Table 1. (right) A photo of the 3D printed metal component. (center) Voxel rendering of 3D reconstruction after CT scan. (right) STL rendering of component with internal defects.

Real Component	3D CT Reconstruction	STL Component with Defects

4.2 Autoencoder Training and Defect Generation

Synthetic defect generation employs an AE. The initial step involves creating the AE architecture. To address the issue of varying vertex and face counts in STL defects, they are converted into binary voxel volumes at a consistent 25^3 resolution for easier data management (Table 2, 1st row). Approximately 25,000 voxelized defects serve as training data, with the encoder learning to create a simplified representation of the input in the code layer and then reconstruct it using the decoder. Post-training, the encoder and decoder can function as separate models. When the same data is applied to the encoder for prediction, it produces code layer representations. The code layer values for all 25,000 samples are recorded, and mean and standard deviation calculations are performed. These values are crucial for generating meaningful variations with the decoder. Values outside the mean and standard deviation ranges yield unrealistic results. The decoder can produce synthetic defects (Table 2, 2nd row) with higher resolution compared to the extracted defects, enhancing geometric details. This demonstrates the decoder's ability to generate geometries closely resembling real data. The rational number space offers an infinite number of variations, even when constraining values to the mean and standard deviation ranges for each code layer node. The binary voxel count of each generated defect is measured and stored, along with their corresponding STL versions, in a size-oriented database.
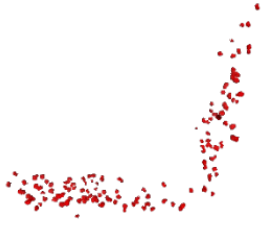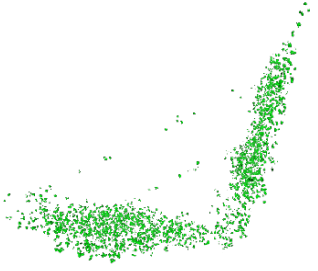
Table 2. (1st row) Extracted defects from the original component as single STL files. (2nd row) Synthetic defects generated with the encoder of the AE brought to application.

Extracted					
Generated					

4.3 Defect Placement Results

In 4.1 we preserved the STL surface hull of the component to place defect variations inside. To achieve this, an algorithm first loads the STL hull, which is the 3D model of the component. It then proceeds to calculate the specific defects to place within the component based on two key parameters: the desired average defect size and the standard deviation. These parameters guide the algorithm in selecting which folders within the defect database to load the defects from. In other words, the algorithm uses these parameters to determine the appropriate sources for defects that will be placed within the STL component. Also, depending on the inner volume in unit^3 and a defined defect density factor the algorithm calculates the number of defects to be placed. Afterwards it searches coordinates inside the STL hull by picking random coordinates, then checking if the coordinates are within the surface using the normal vector of the faces. If a coordinate is valid, it places a defect inside chosen randomly from the database within the defect size spread. The algorithm also allows to enable and disable intersections by using the empty hull or the defect updated hull for valid coordinate search. Additionally, multiprocessing could be activated to decrease the time for processing significantly depending on the available resources. Using our placement algorithm 40 variations of the component with defects are created. In Table 3, we compare the placement of defects. The first row shows large defects with a low density factor, while the second row shows small defects with a high density factor. With the first four steps it is possible to generate a large amount of STL Data with inner defects for further applications. Additionally, is it possible to place the generated defects in any STL.

Table 3. Comparison of defect placement using different parameters.
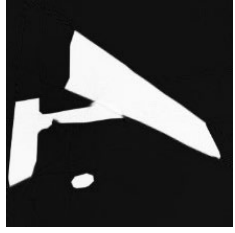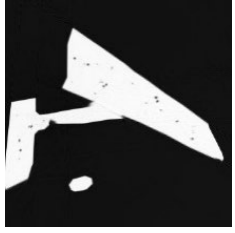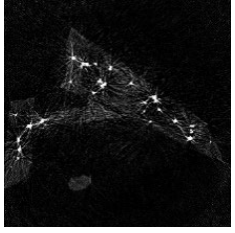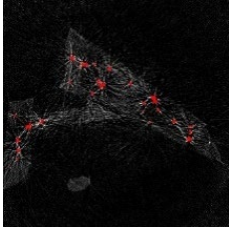
	Cross section of component with internal defects	Internal defects
Defects: 161 Density factor: 0.001 Avg. size: 5000vx		
Defects: 1700 Density factor: 0.01 Avg. size: 367vx		

4.4 CT Scan and Reconstruction Results

Creating useful training data for segmentation we need the samples and the corresponding ground truth. For this purpose, first a single ideal scan and reconstruction is done of the hull without defects and of each of the 40 variations of the defective components. By subtracting the reconstruction of the defective components from the hull only the defects and some random scan and reconstruction artifacts remain. Then a binarization is done to create the ground truth voxel

volume files. The process is illustrated in Table 4. Afterwards, the 40 component variations with defects are simulated and reconstructed more realistically using advanced parameters and functions of the simulation and reconstruction. Afterwards an algorithm is used to chop the volume samples and their ground truth into chunks, therefore it iterates through the sample. If the actual chunk has a certain amount of voxel above a material threshold and this amount is more than a certain percentage of the whole chunk, then this chunk is stored with the corresponding chunk of the ground truth. The selected chunk size is 128³. Using an overlap of 50% when windowing around 8000 samples for training could be retrieved.
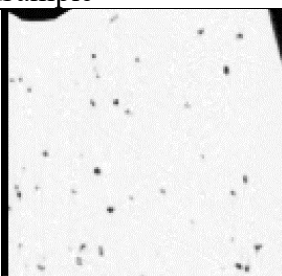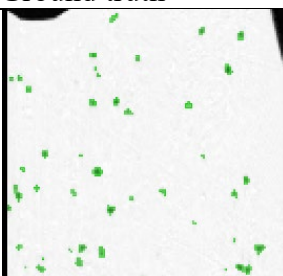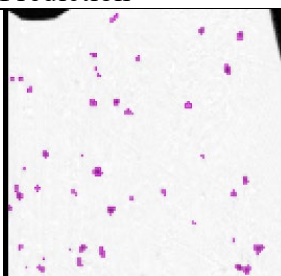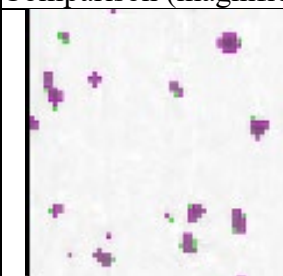
Table 3. Step by step illustration of ground truth preparation.

Hull without defects	Component with defects	Subtraction result	Thresholding	Binary segmentation

4.5 Segmentation Model and Training Results

The segmentation is done using a neural network model with the UNet architecture. Instead of building a model with full size CT volume resolution, the model is trained on the 8000 chunks with a resolution of 128³, split into train-, test- and validation data (80/10/10). This reduces the computational effort and training time. Since the model should focus on the features of the defects and their close surroundings of material to air transition the outer geometry is less relevant and therefore it is common to train on chunks. The model has around 1.5 million parameters. As layer activation the ELU function is used, as output activation sigmoid function. We use binary cross entropy as loss function. To track the progress of the training we use the metric BIoU. The model achieved remarkable 96% BIoU on the validation data. In the following table (Table 5) a sample slice of the synthetic test data is shown together with its corresponding ground truth, the prediction and a comparison between ground truth and prediction. We can see that the model was able to find all defects. The segmentation is almost complete only a very few voxel do not match. The comparison is magnified to increase the visibility of the differences.
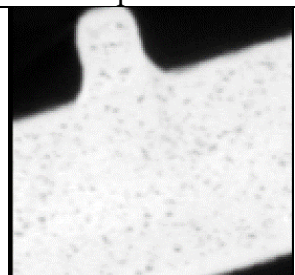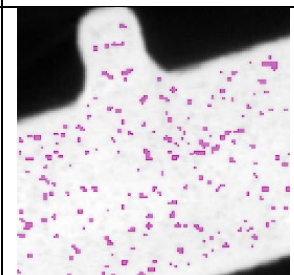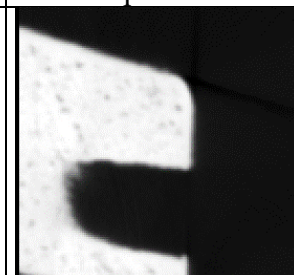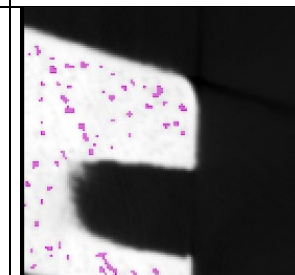
Table 4. Illustration of example sample slice, ground truth, prediction and a comparison overlay of ground truth and prediction

Sample	Ground truth	Prediction	Comparison (magnified)

4.6 Prediction on Real CT Data

Finally, the capabilities of the model trained on purely synthetic CT data and prediction on real data is shown. To predict on the real CT data, it was also chopped into chunks, fitting to the model architecture. In Table 6, we present the outcomes of our research, focusing on the final stage of our process. Our objective in this stage is to generate synthetic data with significant variability. This diverse dataset is intended for training a segmentation model used in quality analysis. Notably, we achieve this using only a single real CT scan as the basis for our synthetic data generation. We can see that the component has in some areas a very high porosity. The darker gray spots in the material are all defects. The trained defects segmentation model can detect most of the defects with good voxel accuracy in respect to the defect geometries. Thus, we show that the entire pipeline works. The synthetic data generated are sufficiently realistic to train segmentation AI models and apply them to real data.

Table 5. Prediction result on 2 real CT data sample slices.

Real sample 1	Prediction 1	Real sample 2	Prediction 2

5   Conclusion

The developed data generation pipeline provides a solution to fill any STL file with artificial defects (pores) that can be used as training data for various AI applications and also other applications which make use of STL data like finite element simulation. This pipeline provides new opportunities for the creation of numerous digital components with specific defect sets. In this work, the pipeline is used to generate data for defect segmentation in CT data. For conversion from STL to CT data we use the simulation tool aRTist and the reconstruction software CERA. By scripting and automating the pipelining steps, our method provides an efficient approach to generating CT data. The segmentation results with synthetic defects are used to evaluate the performance of our pipeline. The successful segmentation of defects in real data proves the effectiveness and stability of our method. This approach enables the generation of an infinite amount of data for various applications that require image data. This is especially valuable in industry, where data for quality assurance and other use cases is often sparse. Without sufficient training data, AI algorithms cannot learn effectively. Our pipeline addresses this problem by providing a wide range of defects with high variation, achieved by adjusting the code layer for pore generation. By combining simulation and reconstruction techniques, we also have control over the complexity and appearance of the defects. In summary, we have succeeded in developing a process that uses CT scans to generate countless realistic replicas from a single component and have demonstrated its quality in comparison with real data.

However, there are some steps in the simulation and reconstruction that require a lot of time and know-how for a good parameterization and thus for the generated results. To optimize, these tools need to be handled more efficiently. One of our next targets is to eliminate the need for external tools and to integrate the whole process into our AI-based pipeline. In addition, we are continuously working on extending the capabilities of our pipeline by adding new defect classes

such as cracks as well as creating a graphical user interface for this pipeline. This will greatly expand the range of applications and take them to a new level of versatility.

References

[1] T. Schanz, R. Tenscher-Philipp, F. Marschall, and M. Simon, "Deep Learning Approach for Multi-Class Segmentation in Industrial CT-Data," *ReJNDT*, vol. 1, no. 1, 2023, doi: 10.58286/28077.

[2] Fuchs *et al.,* "Generating Meaningful Synthetic Ground Truth for Pore Detection in Cast Aluminum Parts," *e-Journal of Nondestructive Testing (eJNDT) 1435-4934*, vol. 9, 2019. [Online]. Available: https://www.ndt.net/article/ctc2019/papers/iCT2019_Full_paper_106.pdf

[3] *aRTist - Analytical RT Inspection Simulation Tool.* [Online]. Available: https://www.artist.bam.de/ (accessed: Jan. 17 2023).

[4] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science (New York, N.Y.)*, vol. 313, no. 5786, pp. 504–507, 2006, doi: 10.1126/science.1127647.

[5] I. J. Goodfellow *et al.,* "Generative Adversarial Networks," Jun. 2014. [Online]. Available: https://arxiv.org/pdf/1406.2661

[6] A. Sharma, O. Grau, and M. Fritz, "VConv-DAE: Deep Volumetric Shape Learning Without Object Labels," Apr. 2016. [Online]. Available: https://arxiv.org/pdf/1604.03755

[7] J. Wu, C. Zhang, T. Xue, W. T. Freeman, and J. B. Tenenbaum, "Learning a Probabilistic Latent Space of Object Shapes via 3D Generative-Adversarial Modeling," Oct. 2016. [Online]. Available: https://arxiv.org/pdf/1610.07584

[8] E. Kalogerakis, S. Chaudhuri, D. Koller, and V. Koltun, "A probabilistic model for component-based shape synthesis," *ACM Trans. Graph.*, vol. 31, no. 4, pp. 1–11, 2012, doi: 10.1145/2185520.2185551.

[9] H. Huang, E. Kalogerakis, and B. Marlin, *Analysis and synthesis of 3D shape families via deep-learned generative models of surfaces*: The Eurographics Association and John Wiley & Sons Ltd, 2015. [Online]. Available: https://diglib.eg.org/handle/10.1111/cgf12694

[10] C. B. Choy, D. Xu, J. Gwak, K. Chen, and S. Savarese, "3D-R2N2: A Unified Approach for Single and Multi-view 3D Object Reconstruction," Apr. 2016. [Online]. Available: https://arxiv.org/pdf/1604.00449

[11] A. Dosovitskiy, J. T. Springenberg, M. Tatarchenko, and T. Brox, "Learning to Generate Chairs, Tables and Cars with Convolutional Networks," Nov. 2014. [Online]. Available: https://arxiv.org/pdf/1411.5928

[12] Özgün Çiçek, Ahmed Abdulkadir, Soeren S. Lienkamp, Thomas Brox Olaf Ronneberger, *3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation.* [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-319-46723-8_49#copyright-information (accessed: Dec. 6 2022).

[13] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," in *Image Processing, Computer Vision, Pattern Recognition, and Graphics*, vol. 9351, *Medical Image Computing and Computer-Assisted Intervention - MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III*, N. Navab, J. Hornegger, W. M. Wells, and A. Frangi,

Eds., Cham: Springer International Publishing, 2015, pp. 234–241. Accessed: Apr. 27 2022.

[14] F. Milletari, N. Navab, and S.-A. Ahmadi, "V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation," pp. 565–571, doi: 10.1109/3DV.2016.79.

[15] T. Schanz, R. Tenscher-Philipp, F. Marschall, and M. Simon, "AI-Powered Defect Segmentation in Industrial CT Data," *The Upper-Rhine Artificial Intelligence Symposium*, vol. 2022, pp. 2–11. [Online]. Available: https://d-nb.info/1272864154/34 #page=10

[16] A. Vaswani *et al.,* "Attention Is All You Need," 2017. [Online]. Available: https://arxiv.org/pdf/1706.03762.pdf

[17] A. Hatamizadeh and H. Roth, "UNETR: Transformers for 3D Medical Image Segmentation," 2022.

[18] Ali Hatamizadeh, Dong Yang, Holger Roth, and Daguang Xu, *UNETR: Transformers for 3D Medical Image Segmentation*, 2021. [Online]. Available: https://www.researchgate.net/profile/ali-hatamizadeh/publication/350253170_unetr_transformers_for_3d_medical_image_segmentation

[19] *CERA – Software for High-quality CT Imaging.* [Online]. Available: https://www.oem-products.siemens-healthineers.com/software-components (accessed: Oct. 15 2023).

[20] W. E. Lorensen and H. E. Cline, "Marching cubes: A high resolution 3D surface construction algorithm," *SIGGRAPH Comput. Graph.*, vol. 21, no. 4, pp. 163–169, 1987, doi: 10.1145/37402.37422.

Road Extraction and Routing from Satellite Imagery by Image Segmentation using Deep Learning

Mohammed Arebi[1]

[1] Furtwangen University of Applied Science
`mohammed.fateh.arebi@hs-furtwangen.de`

Abstract. In this thesis, we address the challenging task of interpreting large-scale satellite imagery by developing an automated system for generating semantic road maps and road graphs with speed limit predictions to enable efficient routing. We explore various convolutional deep neural networks, such as ResNet34, ResNet50, SeResNetX50, and InceptionV3, and conduct extensive studies on hyperparameters and loss functions to optimize the road extraction process. Our pipeline includes image pre-processing algorithms to handle varying image qualities, a model for road segment prediction, and post-processing techniques for graph extraction while retaining geographic information. The results demonstrate the effectiveness of our approach, showcasing the importance of appropriate model selection and optimization. The integration of graph extraction and geographic information enhances the routing process. Overall, this research contributes valuable insights into road extraction and routing from satellite imagery using deep learning, laying the groundwork for future advancements in this field.

Keywords: Satellite Imagery, Road Extraction, Deep Learning, Graph Extraction, Routing

1 Introduction

A stark increase in the amount of satellite imagery available in recent years has made the interpretation of this data a challenging problem at scale. Such images require a deep comprehension of the information contained in them to yield helpful insights. By creating an automated system for generating semantic maps of roads and highways and then further to road graphs with speed limit prediction, allows for routing in satellite images, this thesis investigates the aforementioned issue. Various convolutional deep neural networks were constructed, put into practice, and experimentally tested in order to solve the problem as a supervised machine learning task. We investigate the use of machine learning methods trained on aligned satellite images and possibly outdated maps for labelling the pixels of a satellite image with semantic labels as presented in Figure 1. For this, publicly accessible datasets and frameworks are employed. The resulting pipeline includes image pre-processing algorithms that allows it to cope with input images of varying quality, resolution, and channels, a model that predicts road segments from a satellite image and image post-processing algorithms to extract the road graph while retaining geographic information for efficient routing.

Figure 1. An example of the segmentation mask generating script's output. The raw GeoJSON label is shown in the top left, while the equivalent 8-bit RGB picture is shown in the upper right. The lower-left illustrates the output of the script: the pixel mask inferred from the GeoJSON label. The road mask is overlaid over the RGB picture in the lower right.

The study explores the efficacy of various U-Net variants with different backbone architectures, including ResNet34, ResNet50, SeResNetX50, and InceptionV3, among others to optimize the road extraction process. Additionally, a range of ablations studies and hyperparameter tests were conducted, examining for example different loss functions such as combinations of Binary Cross Entropy (BCE) loss and Dice coefficient, Focal loss, and Dice coefficient, as well as Weighted Tversky loss, mask road buffers, learning rate scheduling and network architectures. The application of graph extraction from the predicted segmentation map, along with the retention of geographic information, proved beneficial in the subsequent routing process. To achieve this, skeletonization and graph cleaning techniques were employed as shown in Figure 2. Speed prediction was also further developed on top of the road segmentation and graph extraction shown in Figure 3.

Figure 2. Prediction for a single mask channel before enhancement (left) and after enhancement (right) in the top row. Transformation of the enhanced prediction mask (left) to a skeleton (right) in the middle row. The skeleton (left) used to generate the graph structure (right) in the final row.
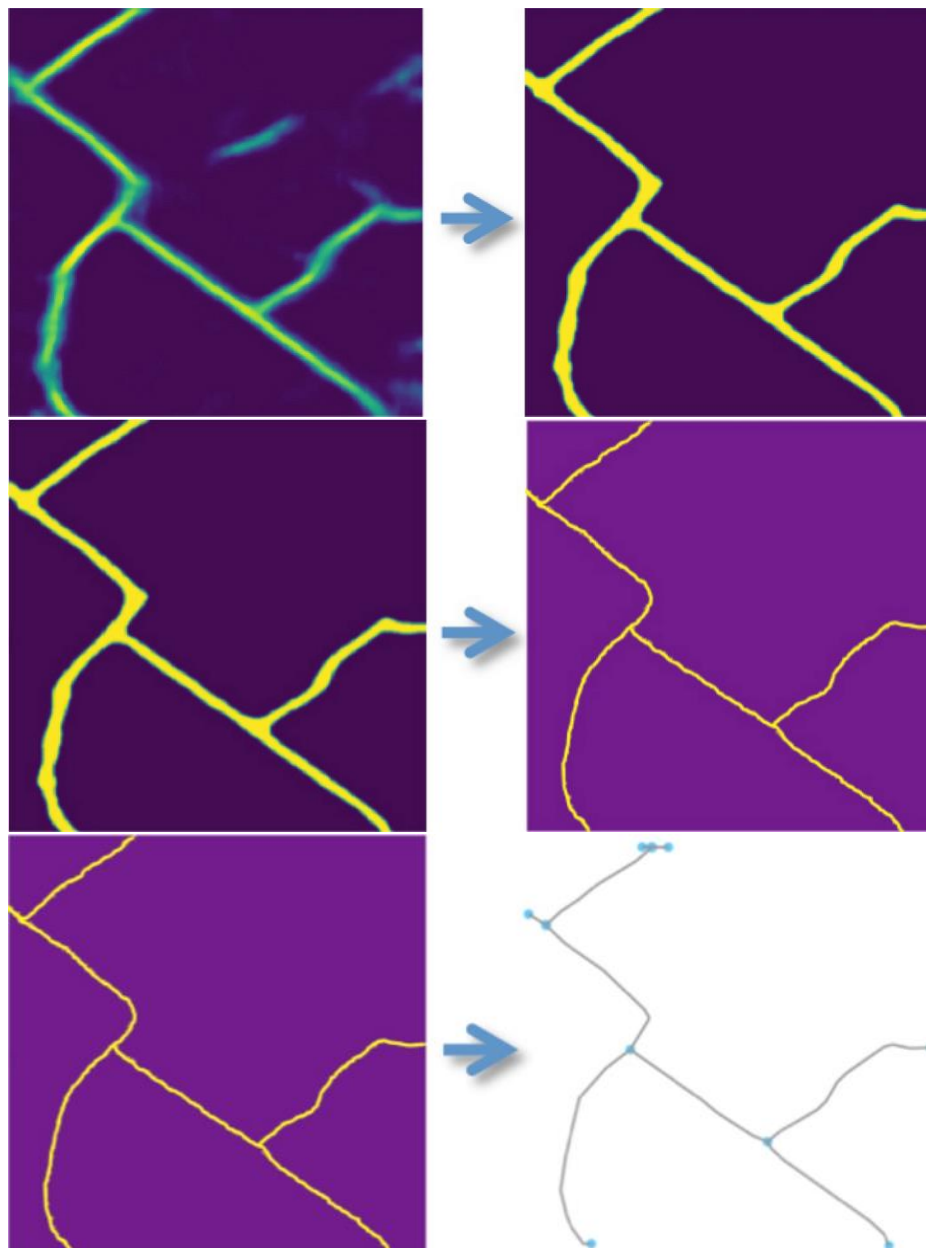
The study commenced with the collection and preprocessing of satellite imagery data. The U-Net model architecture was then implemented, serving as the backbone for road extraction. Various training procedures and hyperparameter optimization techniques were applied to enhance model performance. Evaluation metrics were carefully selected to assess the accuracy and efficiency of the road extraction process.

Figure 3. Left: sample training image in Vegas. Middle: typical binary training mask. Right: continuous mask where the mask value is proportional to the value of interest (in this case, speed limit).

Furthermore, the extracted road network underwent graph extraction, ensuring the preservation of critical geographic information. This facilitated efficient routing capabilities, empowering users to navigate the road network effectively. Skeletonization techniques were utilized to simplify the road network representation, while graph cleaning methods were employed to remove artifacts and enhance the accuracy of the road network.

The experimental results demonstrate the efficacy of the proposed approach. Comparative analysis of different backbone architectures revealed variations in performance, highlighting the importance of selecting appropriate models for road extraction. The hyperparameter tests and evaluation of loss functions, learning scheduling, classification thresholding and ensembling among others, provided insights into their impact on the quality of road extraction results. The utilization of graph extraction and geographic information retention significantly improved the routing process, demonstrating the practical applicability of the developed system. The skeletonization and graph cleaning techniques further enhanced the accuracy and reliability of the road network representation.

In conclusion, this research contributes to the field of road extraction and routing from satellite imagery by utilizing deep learning and image segmentation approaches. The findings underscore the importance of selecting suitable U-Net variants and backbone architectures, as well as optimizing hyperparameters and loss functions for accurate road extraction. The integration of graph extraction, geographic information retention, skeletonization, and graph cleaning techniques plays a pivotal role in achieving reliable road networks and efficient routing. Future research directions may involve exploring additional deep learning models and further refining the proposed methods to enhance performance and address emerging challenges. Figure 4 shows an example for the final pipeline for a sophisticated example image.

Figure 4. Example speed limit prediction incorporated in the predicted graph for an image in the test set, which is then overlaid on the test image.
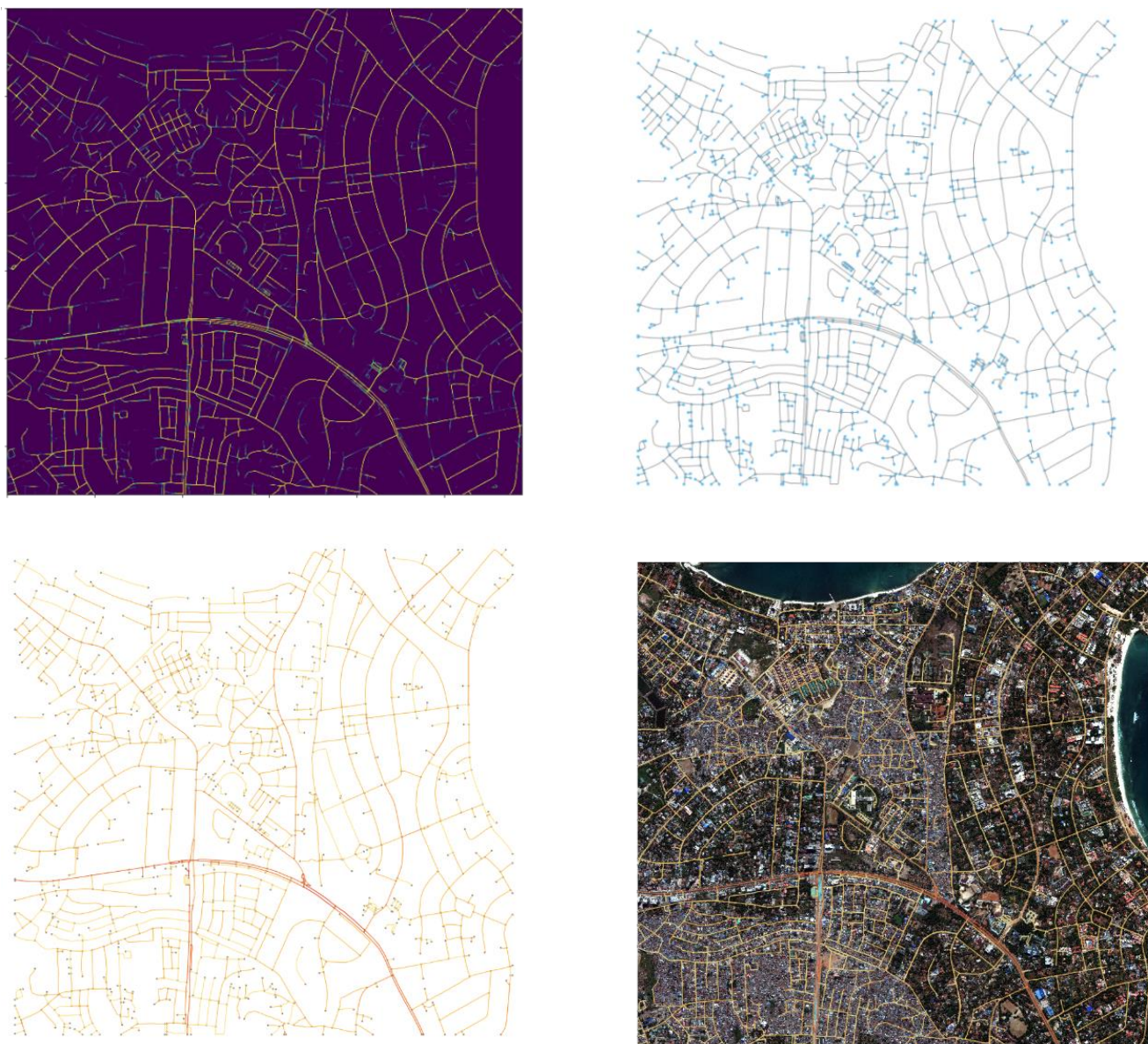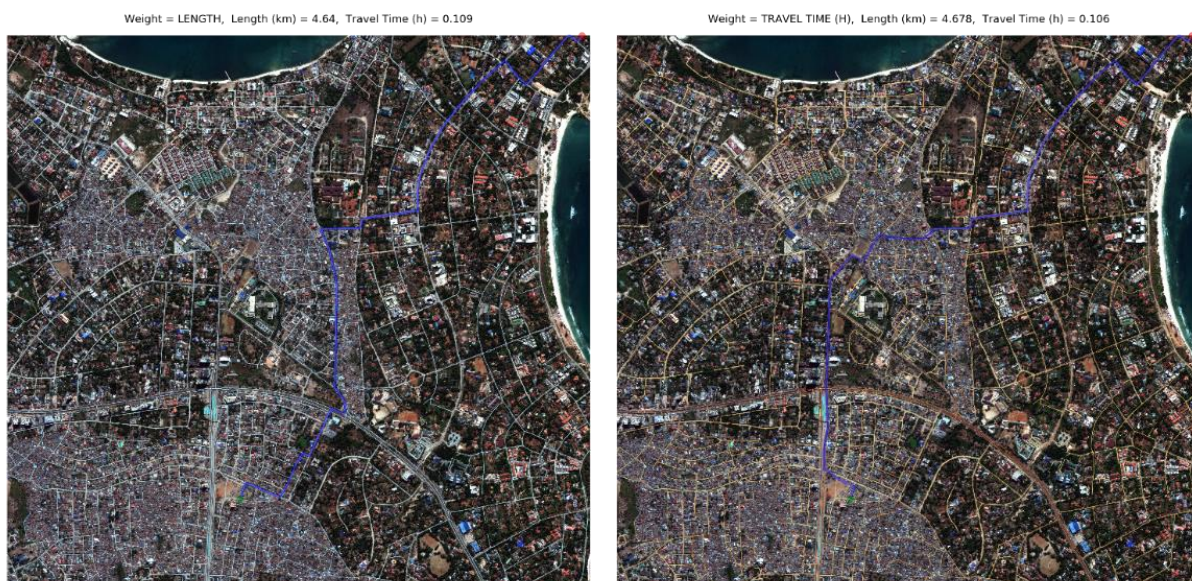
Figure 5. Optimal route based on either distance or travel time using the final proposed speed graph for the test image.

Weight = LENGTH, Length (km) = 4.64, Travel Time (h) = 0.109

Weight = TRAVEL TIME (H), Length (km) = 4.678, Travel Time (h) = 0.106

Discovery: an Explainable AI Powered Academic Search Engine for Knowledge Workers

Robin Weitemeyer[1], Jun Ma[2], Yu Cao[2], Sinan Sen[2], Jens Beyer[3], and Lena Kölmel[4]

[1] Institut für Lernen und Innovation in Netzwerken, Karlsruhe University of Applied Sciences
robin.weitemeyer@h-ka.de
[2] Datalyxt GmbH
{jun.ma, yu.cao, sinan.sen}@datalyxt.com
[3] LAVRIO.solutions GmbH
jens@lavrio.solutions
[4] Institut für Arbeitswissenschaft und Betriebsorganisation, Karlsruher Institute for Technology
lena.koelmel@kit.edu

Abstract. In recent years, transformer models were able to achieve astonishing results in various natural language processing (NLP) tasks. Especially with the rise of ChatGPT and the countless alternatives following its triumphal march, using large language models (LLM) for information retrieval has established itself in both the private, work and research context. For scientific search however, semantic analysis with LLMs is an underutilized tool for enhancing the work process of literature research. We therefore propose an academic search engine called Discovery, which uses BERT to semantically analyze arbitrary text queries in order to recommend fitting search results for scientific publications. Through explainable artificial intelligence (XAI), additional information about the AI output is provided to the user with the goal to decrease the time needed for evaluating the suitability of a recommended paper.

Keywords: Large Language Model, Transformer, Knowledge Work, Academic Search Engine, Semantic Search, Explainable Artificial Intelligence

1 Introduction

Knowledge workers became valuable assets for today's organizations as forecast by Drucker [1]. By applying their expert knowledge to tasks such as general problem solving and developing of new products, they accelerate progress in both science and technology [2]. It is thereby in the interest of every company to maximize the productivity of its knowledge workers.

A knowledge task where we see major potential for efficiency improvement is the acquirement of new knowledge to keep up with new technical and scientific advances. The International Data Corporation (IDC) estimated in 2001 that a knowledge worker spends "roughly 30% of the workday searching for information" [3]. In 2012, Borlund et al. stated that "prior knowledge or no knowledge of the topic in question do also influence searching time" [4]. In addition, Toms et al. found out in 2013 that two-third of the time needed for finding relevant information "was mainly spent reviewing documents that had already been retrieved" [5]. We investigated those insights further and developed a tool, which aims to significantly decrease the time spend on searching for information by providing meaningful recommendations for relevant literature. Due to their state-of-the-art performance in natural language processing (NLP) tasks [6], we believed that a

search engine utilizing transformer-based AI, i.e. large language models (LLM), would be suitable for accomplishing this. Therefore, we introduce our novel explainable AI powered academic search engine called Discovery. However, our goal was not solely to map a search query to helpful recommendations, but also to provide knowledge workers with additional support for finding information on a topic of which the user has little to no prior knowledge and for quickly evaluating the usefulness of the recommended literature. Both of this present additional difficulties during the search for information as stated by Borlund et al. [4] and Toms et al. [5] and would be crucial for increasing the efficiency of the task of literature research. Especially regarding recommendation evaluation, we argue that explaining the decisions of an AI application with methods of explainable artificial intelligence (XAI) is a critical addition.

In the following we give an overview on related work in Section 2, looking at technologies we deployed and introducing existing LLMs-based search engines. In Section 3, we describe the methodology with which we performed the requirements analysis and implemented our search engine approach. Concluding, we discuss our achievements and give a look into future works.

2 Related Work

Transformer-based AI models have been successfully deployed on a wide range of NLP tasks like machine translation, sentiment analysis, question answering, language modelling and named entity recognition [7]. This is also a result of efforts on pre-training language models on large text datasets. Two of the most popular transformer model families are Bidirectional Encoder Representations from Transformers (BERT) from Google [8] and OpenAI's Generative Pre-trained Transformers (GPT) [9]. Although both achieve outstanding results on NLP benchmarks, there are significant differences between those two model types. BERT as well as GPT are based on the original transformer architecture [10] which follows an encoder-decoder structure. However, BERT is an bidirectional encoder that processes the input in both directions and therefore considers the left and right context of each word. In contrast, GPT is an autoregressive decoder, which predicts the next word only depending on its left context. It is thus arguable that BERT tends to be better in understanding the overall context of a text and therefore learn meaningful representations. GPT on the other hand, excels at generating text to a given input [11]. Thou there have been introduced several different variants for BERT and GPT with their own advances [12–14], traditionally speaking BERT performs well for natural language understanding (NLU) and GPT for natural language generation (NLG) [15].

Since 2001, a significant rise can be observed in the number of releases of new LLM models [15] indicating an increased interest and usage of such AI models. Thus, it is unsurprising that there exist several commercial solutions for literature research utilizing LLMs. The academic search engines Consensus[5], Elicit[6], Scispace[7] and Zeta Alpha[8] use transformers for question answering and summarization of abstracts or insights of several scientific publications. In addition, they also list fitting literature for a search query. However, not all of those search engines leverage the full potential of LLMs by not at all or not entirely replacing a keyword-based search with a vector search, i.e. a semantic search. For example, Consensus performs their search in multiple steps. First, they run

---

a simple keyword-based search on the paper database. Only afterwards, a vector search is run on the reduced dataset. This artificially narrows down the extent of the semantic search.

One feature that all of these search engines lack is the explanation of their AI recommendations. By summarizing the content of relevant papers, they potentially reduce the amount of information that has to be review by the user. However, it provides no insight into why the AI model estimates the recommended papers to be relevant for a search query. XAI can be utilized to generate such insight into the AI models behaviour and logic [16]. Danilevsky et al. [17] give an overview of XAI methods and visualization techniques for NLP. However, in the case of transformer models, generating good explanations still poses a challenging task due to their increasing size and complex architecture [18].

3 Methodology

This section will give an overview of the methods used for requirements analysis, computing the literature recommendations and explaining the AI output for better understanding why a certain scientific publication is recommended.

3.1 Requirements Analysis

For requirements analysis, we conducted interviews with research associates of different experience and disciplines to gain more insight in their search strategies as well as most common pain points while conduction literature research. We interviewed six knowledge workers with an average age of 27 years and of whom three were female (standard deviation = 4.6). By selecting the interview partners, we aimed for a diverse distribution in terms of research type (basic research (n = 3) or applied research (n = 3)), scientific working methods (empirical (n = 4) or non-empirical (n = 2)), professional orientation (human (n = 1), technical (n = 2) or organizational (n = 3)) and current career stage (graduate (n = 1), PhD student (n = 4) or post-doctoral (n = 1)). Participants had only superficial knowledge about AI-related topics and did not work in an AI-related field. On average, the interviews took 71 minutes and were conducted as semi-structured interviews divided into four segments. First, we asked about the procedure of a prototypical literature research including details about the used literature databases, the formation of search words and the documentation of results. In the next segment, we asked about other search strategies aside from the prototypical literature research. We wanted to know which kind of different research scenarios exist and what changes regarding the prototypical procedure those entail. Next, we asked about pain points while conducting literature research and their causes. Lastly, we asked the participants to describe the perfect literature research support system, without any barriers due to technical feasibility. We asked about useful features and what kind of pain points could be solved by implementing those.

To analyse the data we transcribed the interviews from speech to text via f4x. Due to the diverse answers between the different interview partners, we decided against a standardized qualitative analysis and instead isolated the most relevant text passages, summarized them into key aspects and finally grouped them into semantically similar clusters. This procedure resulted in a suitable database from which we derived empirically substantiated system requirements. Participants reported that they usually follow an unstructured procedure while conducting a literature research. They adapt their strategies

along the way, depending on the success rate of e.g. the prior search words. They could differentiate between different literature research scenarios (e.g. planning of an experiment, preparation of a manuscript, identifying interesting research gaps, informing oneself about the status quo regarding a topic, learning about new methods or theories, identifying hot topics or looking for literature on behalf of third parties). None of our interview partners identified conscious adaptions to the prototypical procedure depending on the specific research scenario. Most participants reported that during the search process, they mostly only screen title and abstract, full text reading is a subsequent process. Further, the analysis of the interviews led to the following key insights:

- Searching for literature on an interdisciplinary topic or topics in general about which the interviewee has little prior knowledge is stated as one of the biggest pain point.
- Additional information about the search results is requested in order to speed up the process of the immediate evaluation whether or not a recommended paper should be considered for a more thorough inspection.

We then derived three key requirements from the interview results for the listed insights:

1. The search engine should be able to perform a semantic search on continuous text written in natural language so that the user has the possibility to only describe what to search for instead of knowing the exact terminology.
2. The search engine should be able to recommend terms which are related to the search query but possibly unknown to the user since they may be primarily used in different disciplines.
3. The search engine should provide additional information to the user about why the deployed AI considers the recommendations fitting to the search query.

Based on these requirements, the first prototype of our novel AI powered academic search engine was developed. The use of LLMs to compute recommendations addresses the first two requirements while the application of XAI aims to generate the desired additional information of the third.

3.2 Semantic Sentence Similarity

To meet the key requirement 1 we decided to utilize LLMs for replacing a keyword-based search with a semantic search [19]. For this, a transformer model should be deployed to learn meaningful sentence embeddings for a semantic search on research literature by computing the semantic similarity between those vector representations. Recommending publications based on this semantic sentence similarity approach allows for a more intuitive use of our search engine by knowledge workers of different experience. It is no longer required to carefully construct an effective search string with precise combinations of correct terms and their synonyms in order to receive helpful recommendations. Instead, our AI powered academic search engine Discovery can process arbitrary text written in natural language and automatically consider related terms. Thus, making it possible to provide a rather descriptive search query. In Discovery, the literature recommendations are determined by calculating the cosine similarity between the sentence embeddings of the search text and the titles and abstracts of the papers.

Due to BERT's suitability for NLU tasks, we implemented Discovery using the SentenceTransformer[9] Python framework which originates from the Sentence-BERT (sBERT)

---

[9] SentenceTransformer, https://www.sbert.net/, October 2023

network [20]. sBERT builds on the base BERT model and fine-tunes it in a Siamese network architecture. This extension allows for better performances when combined with similarity measures such as the cosine similarity. For our approach, DistilBERT [13] was chosen as the pre-trained BERT model, which was then fine-tuned in a Siamese network on the online contrastive loss with the cosine distance as the similarity measure. Our own database for the search engine and model training consists of over 350.000 publications from IEEE, Springer and Scopus containing publicly available metadata, e.g. title, abstract and keywords. To start with, we confined the database on the topic of AI to make the required data size more manageable. Since there was no ground truth for the similarity between the data points and no reference search queries, we used the self-supervised training method contrastive learning [21]. The training task was to learn sentence embeddings such that the title and abstract of a paper have a high similarity. For positive samples, we kept the original title and abstract pair. For negative samples, we shuffled the titles and abstracts to receive not only incorrect title-abstract pairs but also random title-title and abstract-abstract combinations. The resulting positive and negative samples had a ratio of 1:3. Although, in the face of computation constraints, we were forced to limit our training data to a subset of 150,000 data points. As expected of the contrastive loss, the positive samples were assigned a label of 1, which reduces the distance between the two text inputs, and 0 for the negative samples, increasing the distance. The network used for fine-tuning is shown in Figure 1. For dimensionality reduction, we introduced two fully connected layers with dimensions of 256 and 128 along with Tanh activation functions. This reduces computational complexity while still providing a high enough dimensional vector for our model to effectively capture semantic nuances in a more compact space. We fine-tuned our entire model over five training epochs utilizing the AdamW optimizer with a learning rate of 2e-05 and weight decay of 0.01 to ensure model stability and convergence. The training took a total of 7 hours and 31 minutes on Google Colab using the T4 GPU, highlighting the efficiency and effectiveness of our training pipeline. This efficient use of computational resources underscores the practicality and scalability of our approach.

In order to fulfil the key requirement 2, we introduced an additional functionality to Discovery: term recommendation. The recommended terms are selected from a dataset, which we create by aggregating the keywords over all papers in our database. Since related terms occur in semantically similar texts, the same model can be used which we fine-tuned for the task of paper recommendation. The difference is that instead of searching over the titles and abstracts, we compute the sentence embeddings of the keywords with our BERT-based model and compare them to the input string. The user may submit a search request for either a single term or a descriptive text. Recommending related terms can aid users with limited or no prior knowledge in their search while also introducing them to the appropriate terminology, thereby creating new knowledge.

3.3 Explainable Artificial Intelligence

Key requirement 3 has the intent to decrease the time a knowledge worker needs to evaluate the literature recommendations and to select the set of suitable papers for further consideration. We argue that providing additional explanations about why the AI model recommended the exact papers can improve the user's decision making [22]. However, comprehending the intrinsic logic of LLMs presents multiple challenges. First and foremost is their black box characteristic [23] due to their complexity. With hundreds of millions to billions of parameters, understanding the intricate relationships and dependencies within the model becomes an impossible task without additional explanations.
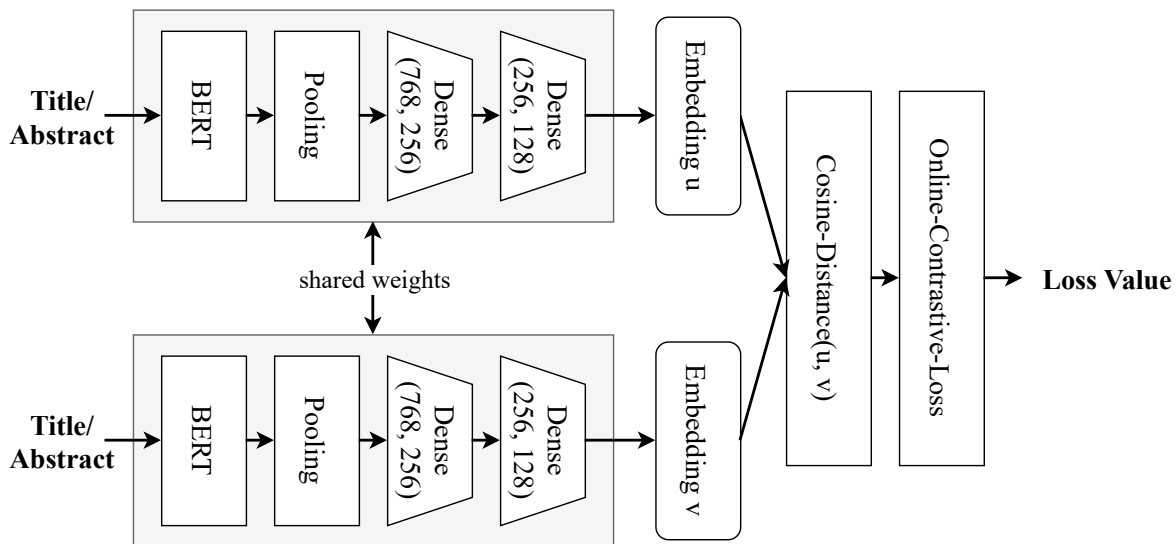
Fig. 1. Visualization of the model architecture used for computing the embeddings between which the consine similarity is calcutated. The model was trained in a Siamese network architecture using constrastive learning. The original title and abstract pairs were used as positive samples, while titles and abstracts were shuffled to synthesise negative samples.

The non-linearity introduced by deep architectures with non-linear activation functions further obscures straightforward interpretations, making it difficult to discern any direct input-output relationships. Additionally, these models are inherently data-driven, meaning their behaviour is shaped by vast and varied training data. This data-driven nature can embed subtle nuances, biases, and patterns from the training data into the model, many of which may not be immediately apparent or interpretable.

In contrast to other use cases of XAI, we do not perform a classification task on the scientific papers but instead measure the semantic similarity between two text fragments. Python packages implementing the XAI methods like LIME[10] [24] or SHAP[11] [25] are highly focused on these more usual use cases of XAI. Thus, we implemented our own XAI algorithm, which was inspired by the visualization of LIME explanations on text data.

For the purpose of our academic search engine, we want to explain to the user how the words in the search string influenced the similarity measurement between the user input and the recommended papers. Hence, we do not want to globally explain the LLM, but instead generate local explanations based on word importance. The main idea of our algorithm follows a similar concept as the one proposed by Landthaler et al. [26] for semantic text matching. The XAI algorithm leaves one word out of the input and then recalculates the similarity score for the top N search results. For each paper, the magnitude of the change in similarity correlates to the importance attributed to the left out word. This step is repeated for each word in the user input. We visualize the word importance by colouring the words of the search query according to their impact on the similarity score. An example for the visualization of our XAI method is shown in Figure 2.

To begin with, we wanted to keep the explanation as simple as possible to not overwhelm the user. Hence, the word importance is only calculated for the user input. How-

---

[10] LIME Python Package, https://github.com/marcotcr/lime, October 2023
[11] SHAP Python Package, https://github.com/shap/shap, October 2023

Fig. 2. Visualization of the custom XAI method for a search query example. Red depicts positive and blue negative word importance.

ever, our algorithm can easily be modified to also consider the title and abstract of the recommended papers. Our intention is to gradually increase the complexity of the explanations guided by user feedback.

4 Conclusions

We presented the working prototype of our novel explainable AI powered academic search engine Discovery for recommending literature and related terms to knowledge workers based on semantic sentence similarity. It was implemented using state-of-the-art LLM and XAI techniques by successfully deploying a pre-trained BERT-based model, which was fine-tuned on our own database with contrastive learning in a Siamese network architecture, for the task of measuring sentence similarity. Through the integration of our own XAI method, Discovery is further capable of providing an explanation for every recommended paper to the users based on the word importance of their search queries. Our search engine was developed for three key requirements, which were derived from the results of our requirements analysis consisting of a literature research and interviews with probands of our target group. During so, we could identify significant potential for improving the efficiency of the task of acquiring new knowledge through literature research.

Our approach fully replaces the traditional keyword-based search with a vector search thereby setting Discovery apart from established academic search engines like Google Scholar or Semantic Scholar. Further, even in comparison to other transformer-based search engines, as those discussed in Section 2, we introduce additional features by recommending related terms and explaining the model output through XAI. Other search engines utilize LLMs to summarize abstracts or extract insights from scientific papers. We see this generative approach critical in the context of literature research. For example, ChatGPT has been proven to lack accuracy providing scientific facts [27]. In addition,

it is stated in the FAQ of Elicit that the users should assume that only 90% of the information they see is accurate. Further, summarizing and extracting the insights could potentially mislead the knowledge worker into relying only on the generated information, rather than considering the recommended papers in more detail. Our XAI approach is designed to accelerate the immediate assessment of the recommendations, but not to make the more thorough review of the papers obsolete.

The eligibility of the recommendations provided by Discovery has been initially tested and shows promising results, however, more extensive evaluation of the system is required through further human feedback and comparison with suitable benchmarks [28]. Furthermore, first feedback indicates that the word importance is not satisfying the need for explanation and that a topic-based explanation is assumed to be more effective. Thus, additional XAI approaches have to be implemented and evaluated on the target group.

Acknowledgements

References

1. Drucker, P.F.: Landmarks of Tomorrow: A Report on the New "Post-Modern" World. Routledge (1996)
2. Powell, Walter W.; Snellman, K.: The knowledge economy. Annu. Rev. Sociol. (Annual Review of Sociology) (2004) 199–220
3. Feldman, S., Sherman, C.: The high cost of not finding information: An idc white paper. KMWorld Magazin (2001) 1–10
4. Borlund, P., Deier, S., Bystroem, K.: What does time spent on searching indicate? In: Proceedings of the 4th information interaction in context symposium. (2012) 184–193
5. Toms, E.G., Villa, R., McCay-Peet, L.: How is a search system used in work task completion? Journal of Information Science **39**(1) (2013) 15–25
6. Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Le Scao, T., Gugger, S., Drame, M., Lhoest, Q., Rush, A.: Transformers: State-of-the-art natural language processing. Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations (2020) 28–45
7. Lin, T., Wang, Y., Liu, X., Qiu, X.: A survey of transformers. AI Open **3** (2022) 111–132
8. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of naacL-HLT. Volume 1. (2019) 4171–4186
9. Radford, A., Narasimhan, K.: Improving language understanding by generative pre-training (2018)
10. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. In: Advances in Neural Information Processing Systems 30 (NIPS 2017). (2017)
11. Bosley, M., Jacobs-Harukawa, M., Licht, H., Hoyle, A.: Bosley, mitchell, et al. "do we still need bert in the age of gpt? comparing the benefits of domain-adaptation and in-context-learning approaches to using llms for political science research (2023)

12. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V.: Roberta: A robustly optimized bert pretraining approach (2019)

13. Sanh, V., Debut, L., Chaumond, J., Wolf, T.: Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter (2020)

14. Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., Amodei, D.: Language models are few-shot learners. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., Lin, H., eds.: Advances in Neural Information Processing Systems. Volume 33., Curran Associates, Inc. (2020) 1877–1901

15. Naveed, H., Khan, A.U., Qiu, S., Saqib, M., Anwar, S., Usman, M., Akhtar, N., Barnes, N., Mian, A.: A comprehensive overview of large language models (2023)

16. Gunning, D., Aha, D.: Darpa's explainable artificial intelligence (xai) program. AIMag (AI Magazine) **40**(2) (2019) 44–58

17. Danilevsky, M., Qian, K., Aharonov, R., Katsis, Y., Kawas, B., Sen, P.: A survey of the state of explainable ai for natural language processing. In: Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing. (2020) 447–459

18. Deb, M., Deiseroth, B., Weinbach, S., Schramowski, P., Kersting, K.: Atman: Understanding transformer predictions through memory efficient attention manipulation (2023)

19. Jimmy Lin, Rodrigo Nogueira, A.Y.: Pretrained Transformers for Text Ranking: BERT and Beyond. Synthesis Lectures on Human Language Technologies. Springer Cham (2022)

20. Reimers, N., Gurevych, I.: Sentence-bert: Sentence embeddings using siamese bert-networks. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processingand the 9th International Joint Conference on Natural Language Processing. (2019) 3982–3992

21. Hadsell, R., Chopra, S., LeCun, Y.: Dimensionality reduction by learning an invariant mapping. In: 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06). Volume 2. (2006) 1735–1742

22. Dwivedi, R., Dave, D., Naik, H., Singhal, S., Omer, R., Patel, P., Qian, B., Wen, Z., Shah, T., Morgan, G., Ranjan, R.: Explainable ai (xai): Core ideas, techniques, and solutions. ACM Computing Surveys **55**(9) (2023) 1–33

23. Adadi, A., Berrada, M.: Peeking inside the black-box: A survey on explainable artificial intelligence (xai). IEEE Access **6** (2018) 52138–52160

24. Ribeiro, M.T., Singh, S., Guestrin, C.: Why should i trust you? In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016. (2016) 1135–1144

25. Lundberg, S.M., Lee, S.I.: A unified approach to interpreting model predictions. In Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R., eds.: Advances in Neural Information Processing Systems. Volume 30., Curran Associates, Inc. (2017)

26. Landthaler, J., Glaser, I., Matthes, F.: Towards explainable semantic text matching. Proceedings of Jurix 2018: International Conference on Legal Knowledge and Information Systems (2018)

27. Borji, A.: A categorical archive of chatgpt failures (2023)

28. Kalyan, K.S., Rajasekharan, A., Sangeetha, S.: Ammus : A survey of transformer-based pretrained models in natural language processing. (2021)

Posters and Coffee Break

Fine-Grained Product Classification on Leaflet Advertisements

Daniel Ladwig[1], Bianca Lamm[1,2], and Janis Keuper[1]

[1] IMLA, Offenburg University
dladwig@stud.hs-offenburg.de, keuper@imla.ai
[2] Markant Services International GmbH
bianca.lamm@de.markant.com

Abstract. In this paper, we describe a first publicly available fine-grained product recognition dataset based on leaflet images. Using advertisement leaflets, collected over several years from different European retailers, we provide a total of 41.6k manually annotated product images in 832 classes. Further, we investigate three different approaches for this fine-grained product classification task, *Classification by Image*, *Classification by Text*, as well as *Classification by Image and Text*. The last both approaches use the text extracted directly from the leaflet product images. We show, that the combination of image and text as input improves the classification of visual difficult to distinguish products. The final model leads to an accuracy of 96.4% with a Top-3 score of 99.2%.
https://github.com/ladwigd/Leaflet-Product-Classification

1 Introduction

The monitoring of product prices is an important data analysis task for retailers as their own price strategy heavily depends on the prices set by competitors. In this context, the monitoring of product advertisements in printed or online leaflets are the predominant source to obtain pricing and promotion activities from competitors. However, the highly unstructured and multi-modal (image + text information) nature of leaflets and the large number of often very similar products makes the underling product identification and matching task quite challenging. Figure 1 depicts an example of the promotions of the same product in the leaflets of two different retailers.

From a Computer Vision perspective, the retail product price monitoring task resolves into several objectives, from product detection to fine-grained classification (FC). Due to its overall complexity, these tasks are currently mostly solved manually, requiring vast resources. In this work, we focus on the FC task and provide a first dataset with 41.6k manually annotated product images in 832 classes manually obtained from leaflets, alongside first baseline solutions.

2 Related Work

We focus our literature review on publicly available and annotated image collections for the application of *fine-grained product classification* on images in a retail context. The data sources of the datasets vary strongly. First, images from web stores are used
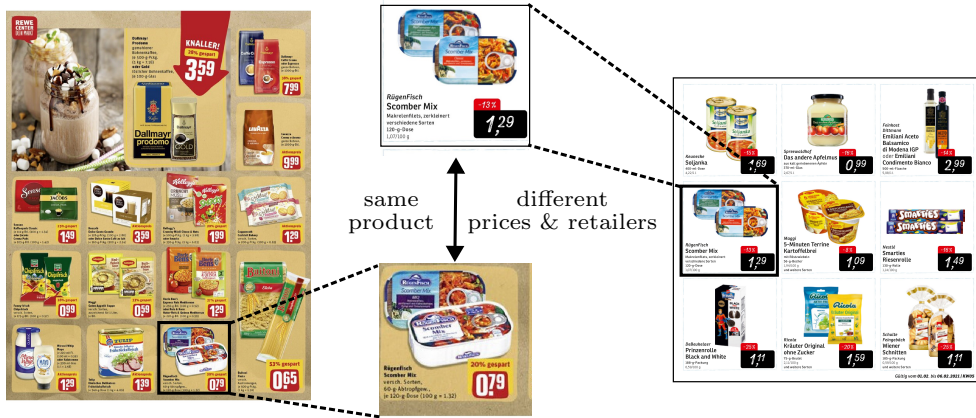
Fig. 1. Price monitoring based on printed leaflets is a key data analysis task in retail, which technically can be defined as a fine-grained, multi-modal classification problem. We provide a first public dataset with 41.6k annotated samples for this task.

as a data source. [1] introduced a *Retail Product Categorisation Dataset* which covers about 48k products with staged "studio" product images in 21 categories. These images are recorded in controlled environments. The authors developed a concatenation of a Convolutional Neural Network (CNN) and a Long Short-Term Network for the FC task [1]. For more realistic real-life scenarios product images "in the wild" are used. The *Products-10k* [2] collection comprises about 10k product classes for about 150k "studio" images and "in the wild" images recorded by customers. The backbone of their approach is the model EfficientNet-B3 [3]. Also, images of supermarket shelves serve as a data source. [4] provides the *Retail-121* dataset consisting of 121 fine-grained retail product categories. For solving the FC task, the authors supply an own approach that is based on a non-parametric probabilistic model and a CNN [4]. To the best of our knowledge, there are no datasets that are based on images from leaflets. Hence, we provide the first annotated dataset containing of product promotions cropped from leaflets.

3 Dataset Description

Data Sources. Our dataset is based on a large collection of full page images in JPG format provided by the company Markant Services International GmbH. The publicly available digital or manually scanned leaflets are circulated by well-known European retailers. Figure 2 shows three representative samples. The leaflets were randomly collected from 132 different[3] retail chains between calendar week 39 in 2016 to calendar week 38 in 2022. The leaflets advertise mainly food and beverages. But also non-food products like household goods, cosmetics, pet foods, or (small) electric devices are promoted. Each leaflet page has been manually segmented into product information boxes by humans. Each box must contain the product image, price, and description. Additional logos, price tags, or quality seals can also be contained. The cropped boxes from the leaflet pages form our provided image dataset. The original leaflets as well as the text information like prices, discounts, or product descriptions are not included in dataset.

---

[3] Large retail chains often have different subsidiaries and brands which are potentially using the same product images for advertisement. In order to avoid poisoning of the test sets, we grouped all sub-brands into single retailers.

Fig. 2. Three leaflet pages from different European retailers.

Dataset Properties. Each class in the dataset represents one product. The dataset is composed of 832 classes and 41.6k images in total, split into a training set of 33,280 images and a test set of 8,320 images. Each class has 40 images in the training set and 10 images in the test set. All images adhere to a minimal resolution of at least 92 pixels in width and 138 pixels in height while the longer edge is always fixed to 512. This dataset has a size of 3.4 GB. Reducing the longer edge to a fixed value of 256, results in a dataset of a size of 1.2 GB. We provide both versions for public download. Figure 3 shows the training and the test set for a class. The images of a class are similar and diverse at the same time. There can be multiple images from one retailer in a split set but a retailer can only be represented in the training or in the test set for a class.

Fig. 3. Training set (left) and test set (right) of a class.

4 Baseline Solutions

We investigate baseline solutions of the FC task on our dataset by using different information of an image. The *Classification by Image* uses the whole image as input information. The extracted text of an image is the input for the method *Classification by Text*. Moreover, the combination of both information, *Classification by Image and Text*, is analyzed.

146

4.1 Classification by Image.

We provide first baseline results on our datasets, applying current state-of-the-art methods for FC. We used the dataset version with a longer edge length of 256.
[5] provides a review of current FC methods. For our baseline evaluation, we chose four different image classification models: *ResNet50* [6], *MobileNet V2* [7], a *Vision Transformer* model [8] with a *vit_b_16* architecture and *ConvNeXt* [9] using the base model architecture. Each model has been pretrained on *ImageNet* [10]. The following settings are used for the training process on a NVIDIA GeForce RTX 3090 GPU: 30 epochs of fine-tuning, a batch size of 32, and a SGD optimizer with a learning rate of 0.001 plus a momentum of 0.9. As Table 1 shows, *MobileNet V2* gives the lowest test set accuracy result with an accuracy of 0.894. However, this model needed the shortest training time. The *ResNet50* model and the *Vision Transformer* model have nearly the same test set accuracy of 0.907 and 0.909, respectively. Also the Top-5 accuracy scores of both models is about 0.971. However, the *Vision Transformer* requires almost twice of the training time compared to the *ResNet50* model. The best test set accuracy is reported for the *ConvNeXt* model with 0.921, but its training took almost four hours. We use the *ResNet50* model as the final image model because of its solid accuracy relative to an acceptable training time. In the fine-tuned model the last *Fully Connected* layer is replaced with a *Linear* layer of 2048-1024, followed by a *ReLu* activation function and closing with a *Linear* layer of 1024-832. The hyperparameter used are: batch size of 16 and optimizer momentum of 0.95. A torch ColorJitter is applied as data augmentation with a saturation of 0.5.
Error Analysis. By analysing falsely classified products, errors between visually similar classes occur. Figure 4 exemplifies such products, which still match at a higher abstraction being very similar items from the same producer.

Table 1. Listing of the test accuracy scores, training time, and the Top-5 accuracy for tour FC dataset. The models *ConvNeXt, ResNet50, Vision Transformer* and *MobileNet V2* were trained.

model	test accuracy	training time [h]	Top-5 accuracy
ConvNeXt	0.921	3.9	0.974
Vision Transformer	0.909	2.1	0.971
ResNet50	0.907	1.3	0.971
MobileNet V2	0.894	0.9	0.964

4.2 Classification by Text.

The error analysis of the image model shows the difficulty to distinguish especially between products that are served in different package sizes or product variations that are not clearly visually represented on the product image itself. The leaflet product images mostly include a product description and the serving size. Those texts are not available in extracted digital form yet. To process them in further steps, they need to be extracted from the images first.
OCR Extraction. The Tesseract OCR Engine [11] is chosen to extract texts from the images. This tool combines the text detection and the text recognition in one model.

Different page segmentation modes (PSMs) [4] can be set which alters the way text in the image is treated. Changing modes leads to different extracted text results. Since there is no digital truth text available to evaluate the quality of the extracted text, the accuracy of the classification problem is used to evaluate the quality of the extraction.

To extract much text out of different designed product images with changing background colors, text positions and text sizes, several image preprocessing steps and PSMs are combined. The dataset with the larger image sizes with a longer edge of 512 is used to extract the text to utilize the higher text resolution.

The extraction methods were developed incrementally. An extraction method was put together, the resulting wrong predicted images were inspected and a new fitting method for the poorly performing text extractions was developed. In the final model the following methods were used to extract text from the varying input images. OpenCV[12] and its methods were utilized to import the images and to apply preprocessing. The first four methods use Tesseract with PSMs of 3, 6, 11, 12. Another method applies a grayscaling with cvtColor and cv2.COLOR_BGR2GRAY before applying Tesseract with the standard PSM of 3. For additional methods the gray color change was applied together with a resize of the image times four with Tesseract and PSMs 6 and 11. Finally a grayscaling, resize and cv2.threshold with cv2.THRESH_OTSU, a thresh of 0, maxval 255 and Tesseract with PSM 11 is used. The following preprocessing steps and model are used to evaluate the described methods. They resulted in between 73.6% and 86.4% accuracy.

The combination of all developed extraction methods, to extract as much text from the image as possible, boosts the performance to 91.5%. Even if that means that there are duplicate texts in the collection. Extracting the 33,280 train images sequentially took 16.83h.

Text preprocessing. The extracted texts include wrong letters and signs from noise detected as text in the image. The preprocessing has to be done carefully since package size descriptions can include OCR caused confusions of numbers and letters. In the final solution no stopword or special sign removal is done. The raw extracted texts which include duplicate text from combining the methods are encoded with scikit-learn TfidfVectorizer [13].

Text model. A linear SGDClassifier is trained on the tfidf encoded text and the loss function "modified_huber" with an adaptive learning rate is used. Table 2 shows the model accuracy is 91.5% with a Top-5 score of 96.7%. The text models performance is slightly worse than the image models but the correct predicted images differ. The following step combines those two models.

---

[4] For further explanation: https://github.com/tesseract-ocr/tessdoc

Fig. 4. Illustration of two similar products. The left product is often confused with the right one by three of the four baselines.

6

4.3 Classification by Image and Text.

The first step to combine the text and image models is to apply a softmax to each probability predictions. After that a weighted probability stacking is done by giving the text model a higher weight, as the SGDClassifier predicts way lower decision probabilities than the ResNet50 model. Weighting the text model allows to make an impact after stacking the probabilities. The final predictions are differentiated in predictions where both models agree (labeled as high confidence prediction) and predictions where they do not (labeled as low confidence prediction). This allows to separate those low confidence predictions to be potentially reviewed manually with the help of a Top-3 selection with a very high accuracy of 99.2%. Figure 5 displays the process of combining the models and the resulting predictions.
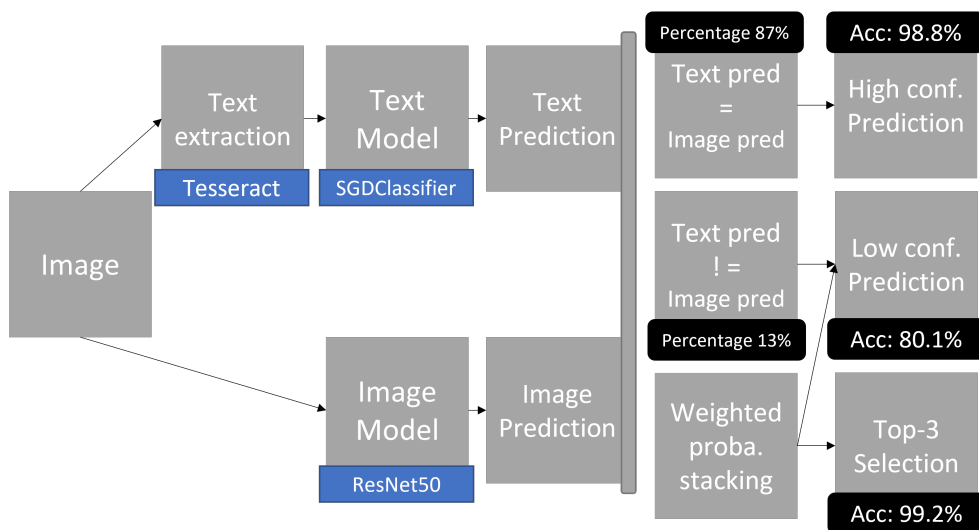

Fig. 5. Model Combination Process

Combination Results. The prediction stacking method by combining the probabilities of the text and image model results in an overall accuracy of 96.4% which is a significant increase from the models alone. Interesting is the calculated Top-3 accuracy of 99.2%. This shows the ability of the combined models to find the correct class in the fine-grained problem and rate it high. The gap between the prediction and Top-3 selection displays the difficulty to make the correct choice when the products do not differ much, sometimes only one attribute differentiates them. There is still space for improvement in the combination process of the two models. The accuracy that the image model prediction or the text model prediction is correct lays by 98.0% which is not quite reached by the used weighted probability stacking method.

Error Analysis. A manual review of the wrong predicted images and classes of the final combined model revealed the following findings. As in the beginning presumed the products that are served in different variations or package sizes can cause confusions for the final prediction. The high Top-5 score shows that the correct product can be found but the final choice which variation is presented is still a difficult task. Figure 4 already shows an example of two similar product servings. Figure 6 shows two classes of products that are distinguished by serving size. The Top-3 products are served in 250g and the bottom products are served in 290g. The package size displayed on the product varies, even within the class. This explains the confusion between those classes for the image

Table 2. Model and Combination Results

Model	Method	Accuracy	Top-3	Top-5
Final ResNet50	Image	0.925	0.961	0.969
SGDClassifier	Text	0.915	0.962	0.967
Combined	Weighted proba.	0.964	0.992	0.993

model. The text model could possibly distinguish those two from the given package size. Although the last image labels the package size as "250g + 40g" instead of the expected 290g which can lead to confusions. In this case the combined model could predict both classes correctly, but this is an example how difficult the differentiation between similar classes can be. Changing product descriptions or difficult text extractions impact the text model predictions negatively.

Fig. 6. Model Confusion Example

5   Conclusions

In this paper, we provide a dataset and present a baseline solution for fine-grained image classification of retail products on leaflets. We propose methods to optimize the product classification by combining image classification with text classification based on text directly extracted from the images. The presented methods allow to extract and process text from varying product images. Improved results compared to the image classification alone support the idea to make use of existing text. Combining text and image classification improves the ability to distinguish between visually similar product representations. As a first part of the price monitoring task we create a sufficient baseline for product matching. Future work will be the extracting of the prices of the product promotions. This is challenging due to the huge variance in color, size, and position in the promotion.

8

References

1. Elayanithottathil, F.S., Keuper, J.: A retail product categorisation dataset. arXiv preprint arXiv:2103.13864 (2021)
2. Bai, Y., Chen, Y., Yu, W., Wang, L., Zhang, W.: Products-10k: A large-scale product recognition dataset. arXiv preprint arXiv:2008.10545 (2020)
3. Tan, M., Le, Q.: Efficientnet: Rethinking model scaling for convolutional neural networks. In: International conference on machine learning, PMLR (2019) 6105–6114
4. Karlinsky, L., Shtok, J., Tzur, Y., Tzadok, A.: Fine-grained recognition of thousands of object categories with single-example training. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2017) 4113–4122
5. Wei, X.S., Song, Y.Z., Mac Aodha, O., Wu, J., Peng, Y., Tang, J., Yang, J., Belongie, S.: Fine-grained image analysis with deep learning: A survey. IEEE Transactions on Pattern Analysis and Machine Intelligence (2021)
6. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2016) 770–778
7. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.C.: Mobilenetv2: Inverted residuals and linear bottlenecks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2018) 4510–4520
8. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020)
9. Liu, Z., Mao, H., Wu, C.Y., Feichtenhofer, C., Darrell, T., Xie, S.: A convnet for the 2020s. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2022) 11976–11986
10. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition, Ieee (2009) 248–255
11. Smith, R.: An overview of the tesseract ocr engine. In: Ninth international conference on document analysis and recognition (ICDAR 2007). Volume 2., IEEE (2007) 629–633
12. Bradski, G.: The OpenCV Library. Dr. Dobb's Journal of Software Tools (2000)
13. : sklearn.feature_extraction.text.tfidfvectorizer. https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html

Time Series Extrinsic Regression for Physical Rehabilitation Assessment

Elvin Ismayilzada[1], Maxime Devanne[1], Jonathan Weber[1], and Germain Forestier[1]

IRIMAS, Université de Haute-Alsace
`firstname.lastname@uha.fr`

Abstract. Rehabilitation is the process of assisting people with disabilities in regaining their function and independence. As artificial neural networks are trained on large datasets using deep learning, rehabilitation can be improved by providing individualized and efficient treatment options. As human rehabilitation involves multivariate time series data, we review well-known algorithms for the classification of time series data. We also discuss the challenges and opportunities presented by the use of deep learning in rehabilitation, including the need for large and diverse datasets and the potential for bias in algorithms. Overall, our analysis indicates that deep learning has the potential to improve rehabilitation outcomes and the lives of disabled individuals. A comparison of many methodologies was conducted in order to establish a framework capable of supporting and reliably evaluating patients' workouts throughout recovery programs. In order to assess the algorithms, two datasets pertaining to human rehabilitation are used: KIMORE, and UI-PRMD for regression tasks.

1 Introduction

One of the most efficient methods to diagnose musculoskeletal issues and rehabilitate post-stroke participants is through physical therapy treatment through workouts on specific tasks. However, it is neither practical nor cost-effective for a physician to attend every rehabilitative exercise program. Patients typically execute these exercises at home, without the involvement of specialists or therapists. Despite the fact that patients are supposed to document and record their progress as well as contact the doctors on a regular basis, various medical organizations have stated that patients are not able to complete the exercises correctly, causing the recovery process to be extended. As a result, patients are unable to receive proper supervision and feedback for the required activity. These challenges make human rehabilitation a hot topic in a research environment [1]. Thanks to recent Computer Vision algorithms, it is possible to capture human motion by estimating from an image joints' 3D coordinates, forming a humanoid skeleton. In this paper, we use Deep Learning to analyze human rehabilitation programs, represented by 3D skeleton sequences, leveraging precise and outstanding performance. We propose to recast the problem of human rehabilitation movement assessment as a multivariate time series analysis. Time series analysis has been investigated for various tasks such as classification [2], clustering [3], averaging [4] and adversarial attacks [5]. In this work we particularly study a well-known architecture in the field of time series classification [6] for the task of rehabilitation assessment. However, there are two main issues that must be addressed prior to deployment:

1. First, we must switch the implementation domain from classification to regression in order forecast a single numerical score corresponding to patient's performance.

2. Second, because deep learning models assume fixed-size inputs, pre-processing procedures are mandatory before any deep learning model can be implemented on the human rehabilitation dataset.

We established our experiments on the KIMORE [7] and UI-PRMD [8] datasets. In particular, Kimore dataset is more practical for real-world situations since it incorporates both healthy and unhealthy subjects. We believe that our proposed method could be very helpful for physical rehabilitation assessment and could for instance be embedded into a autonomous system (like a robot coach [9]) for monitoring rehabilitation sessions in rehabilitation center or at home.

2 Background and Related Work

2.1 Multivariate Time Series

A multivariate time series [6] contains multiple time-dependent features. Multivariate time series $X = [X^1, X^2, ..., X^M]$ contains M individual univariate time-series where $X^i \in \mathbb{R}^T$.

2.2 Convolution

Convolution can be used to [6] perform and sliding filter through time series. It just has one-dimensional filters (time) rather than two dimensions like images. The filter can alternatively be viewed as a non-linear change of time series.

$$Ct = f(\omega * Xt - l/2 : t + l/2 + b) \mid \forall t \in [1, T], \tag{1}$$

2.3 Classification

Various works have been investigated in order to replace the expensive and arbitrary judgment of human experts with an automated process. Classification evaluations are used to estimate categorical values that indicate the ability level of the activities that are performed. Such evaluations categorize executed motions into distinct groups that belong to a rank but can be challenging to precisely define. Assessments may fall into one of two groups in a straightforward classification system (correct or incorrect). On the other hand, adding more classes allows us to make more exact distinctions between executed motions. More examples from each class are required for better performance. As a result, if there are numerous classes or features that enhance complexity, they may create a scaling difficulty.

2.4 Parametric Assessment

Rather than a classification task, a parametric assessment provides continuous value in order to evaluate the performance of rehabilitation exercises. These methods apply strategies similar to other approaches but they emphasize domain-specific factors. As a result, such methods offer helpful detail in their evaluations of executed human rehabilitation exercises. Some research for exercise assessment tasks often concentrates on learning distance measures [10]. Those techniques can find similarities between two random exercises, however, they can't represent task-specific exercises. To solve this issue, another line of research relies on probabilistic techniques for evaluating workouts, such

as Hidden Markov models [11], [12] and Gaussian mixtures [13]. These techniques impede end-to-end processing as they need many pre-processing phases and the knowledge of experts in the particular field. We intend to analyze exercises using deep learning techniques since deep learning algorithms are better suitable for end-to-end processing.

2.5 Evaluation of Rehabilitation Exercise

There hasn't been enough research on this subject. [14] Lee et al. classified a variety of motions into true and false categories with the help of hand-crafted features. In the work of [15] for the evaluation of human rehabilitation exercises spatio-temporal architecture is suggested. In order to boost performance, multi-branch convolution, recurrent networks, and temporal pyramid. The downside of these approaches they employ a variety of feature engineering and pre-processing steps. In light of the recent achievement of deep learning approaches in various fields, we conduct human rehabilitation exercise assessments with deep learning frameworks. We use common time series classification algorithms [6] for this study since human rehabilitation activities involve coordinates of joint positions throughout time.

3 Proposed Approach

In this paper, we adapted the inception network to human rehabilitation exercises. The inception network [16] is convolutional neural network initially created for a more deep representation of time series classification problems. The inception network is made up of two main concepts: bottleneck layers and sliding several filters. By utilizing a bottleneck layer, time series data dimensions can be reduced while capturing complex features and overfitting issues can be minimized. Moreover, this architecture allow to slide several filters of varying lengths over the given input time series at the same time in order to capture meaningful patterns at different scales. Our inception network is adapted to human rehabilitation and extrinsic regression to generate a numerical value that represents the score associated with the input motion sequence. Figure 1 illustrates the proposed architecture.
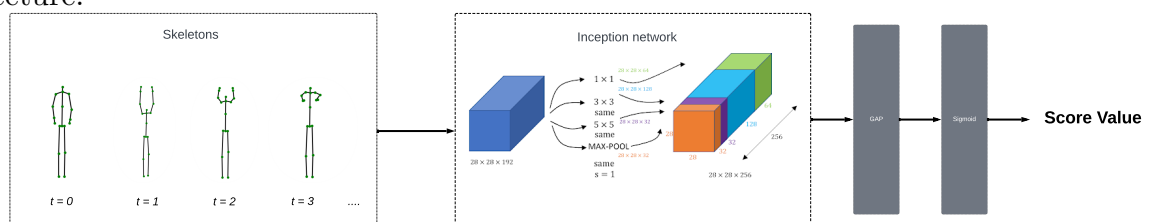

Fig. 1: The overall framework of our proposed rehabilitation exercise assessment

4 Experiments and Results

4.1 Datasets

For evaluating the proposed approach, we use two separate datasets: KIMORRE and UI-PRMD. KIMORE dataset [7] contains RGB-D videos collected by Kinect sensor and clinical scores evaluation of human motions. There are 5 distinct exercises done by control groups and an unhealthy group. The control group included 44 healthy people, 12 of

whom were physiotherapists with competence in the treatment of postural and back pain, while the remaining 32 were non-expert healthy people. The unhealthy group consists of 34 patients who suffer from pain and postural issues and have chronic motor impairments. (2) UI-PRMD dataset [8] is made up of ten rehabilitation activities that were gathered from 10 healthy individuals utilizing Kinect and Vicon sensing devices. There were ten repetitions of the same activity performed by each individual. A comprehensive collection of data is provided, which includes the positions and angles of the joints throughout the body.

4.2 Normalization

Before feeding data through deep learning algorithms, normalization processes must be performed on the data. Because variability in input feature scales can increase the complexity of the model performance. In general, models that are constructed with heavy weights are unstable, meaning that they perform poorly during learning and exhibit high sensitivity to input values. The learning process can become unstable when a target variable has a wide spread of values, and the error gradients may be large, causing weight values to change drastically. As long as your output activation function has a scale of [0, 1], so the target values also fall inside that range. As a result, we normalize all human motion data into the range [-1, 1], and clinical values between 0 and 1. When we apply three different methods to normalize clinical scores:

- Initially, we divide the truth values by 100. The results are negatively affected by dividing by 100 since there is a large difference between the truth values of healthy and unhealthy subjects. Due to this, a model is not capable of capturing these differences between healthy and unhealthy clinical scores.
- Second, we apply minmaxscaler to clinical score labels to obtain values between 0 and 1. This strategy is effective when using the second and third splits, but for the first split, as train and test data is separated it treats the different distribution of data as the same values, therefore deep learning algorithms treat different clinical scores in the same manner. This leads models to mislead to find the desired output values correctly.

4.3 Evaluation Process

For evaluating regression tasks for human rehabilitation exercises we use two metrics: Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE), as defined below:

$$MAE = \frac{1}{n} \sum_{i=1}^{n} |x_i - y_i| \tag{2}$$

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (x_i - y_i)^2} \tag{3}$$

The Root Mean Square Error indicates how condensed the values are close to the best-fit line. The Mean Absolute Error computes an absolute average gap between the actual and predicted data.

4.4 Experimental Results on Kimore

We train our model on the splitting technique using the leave-one-only cross-validation (LOOCV) on unhealthy samples. As part of this technique, one subject is used for testing, and the remaining data serve as a training set, which is combined with healthy and unhealthy samples. As Deep Learning models are dependent on their initial random initialization, we run our algorithm 5 times and report in Table 1 average values with standard deviations for both metrics on each exercise separately.

Table 1: Results of Inception algorithm 5 exercises (Ex) conducted on the KIMORE dataset by MAD and RMSE metrics

Metric	Ex1	Ex2	Ex3	Ex4	Ex5
RMSE	0.33 ± 0.04	0.31 ± 0.05	0.42 ± 0.04	0.34 ± 0.03	0.28 ± 0.02
MAE	0.19 ± 0.03	0.15 ± 0.03	0.27 ± 0.04	0.17 ± 0.03	0.16 ± 0.02

Considering that we used to leave one out cross-validation which improved model performance. Moreover, Figure 2 shows the comparison of real (green) and predicted (red) scores obtained by our approach on test sequences of exercise 2 (lateral tilt of the trunk with the arms in extension). We can see that our proposed method allow to correctly predict corresponding scores with low errors.
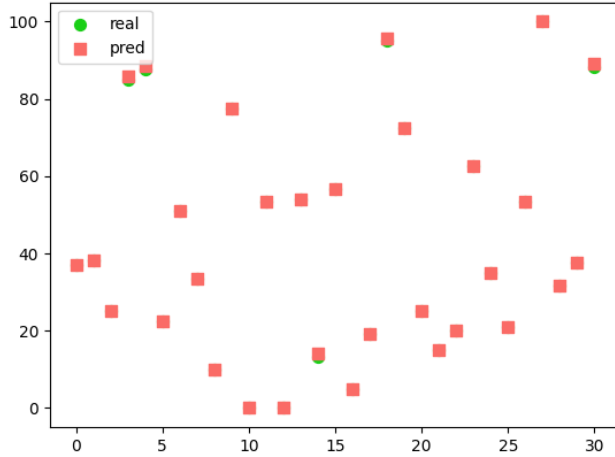

Fig. 2: The comparison plot between predicted and actual values for the Inception network on KIMORE dataset

The UI-PRMD dataset was subjected to the same experiments as the KIMORE dataset using the Inception model. In addition to other experiments related to human rehabilitation exercises, we applied subject-based leave-one-out cross-validation to prevent the use of the same subject information in both the train and test set at the same time.

Based on the Figure 3, we can conclude that our proposed method in the UI-PRMD dataset also shows promising results

Table 2: Results of 10 exercises conducted on the UI-PRMD dataset by MAD and RMSE metrics for the Inception model

Ex	RMSE	MAE
Ex1	0.0227 ± 0.0047	0.018 ± 0.0037
Ex2	0.0128 ± 0.0038	0.01 ± 0.0032
Ex3	0.0245 ± 0.0010	0.0178 ± 0.0011
Ex4	0.0282 ± 0.0026	0.0233 ± 0.0027
Ex5	0.242 ± 0.454	0.0841 ± 0.1442
Ex6	0.0186 ± 0.0017	0.0138 ± 0.0009
Ex7	0.0219 ± 0.0045	0.0172 ± 0.0029
Ex8	0.0316 ± 0.0050	0.0223 ± 0.0029
Ex9	0.02 ± 0.0022	0.017 ± 0.0025
Ex10	0.7436 ± 1.3082	0.2866 ± 0.4182
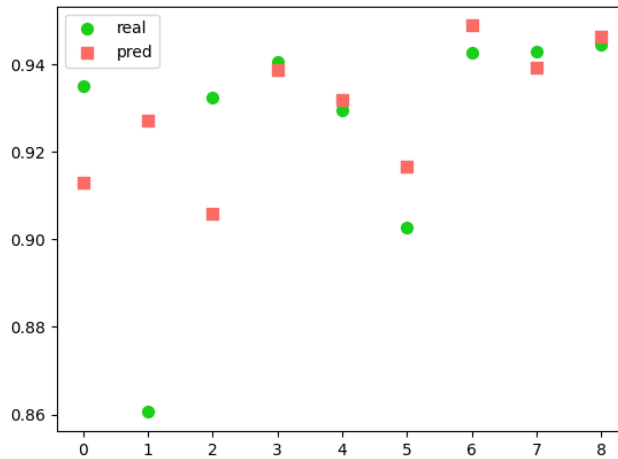
Fig. 3: The comparison plot between predicted and actual values for the Inception network on UI-PRMD dataset

5 Conclusion

In this paper, we proposed a human rehabilitation assessment approach using the Inception network. We modified the original architecture to provide numerical values by considering our analysis as a multivariate time series extrinsic regression problem. Our approach is evaluated on two datasets that represent rehabilitation exercises, the KIMORE dataset and the UI-PRMD dataset. Our results indicate that the Inception network algorithm can be used for human rehabilitation exercises with promising results. As future work, we aim at investigating explainability methods allowing to understand which part of the rehabilitation motion is more responsible of a given score.

Acknowledgement

References

1. Blanchard, A., Nguyen, S.M., Devanne, M., Simonnet, M., Le Goff-Pronost, M., Rémy-Néris, O.: Technical feasibility of supervision of stretching exercises by a humanoid robot coach for chronic low back pain: The r-cool randomized trial. BioMed Research International **2022**(1) (2022) 1–10

2. Ismail-Fawaz, A., Devanne, M., Weber, J., Forestier, G.: Deep learning for time series classification using new hand-crafted convolution filters. In: IEEE International Conference on Big Data (BigData), IEEE (2022)

3. Ismail-Fawaz, A., Ismail Fawaz, H., Petitjean, F., Devanne, M., Weber, J., Berretti, S., Webb, G., Forestier, G.: Shapedba: Generating effective time series prototypes using shapedtw barycenter averaging. In: ECML/PKDD Workshop on Advanced Analytics and Learning on Temporal Data. (2023)

4. Terefe, T., Devanne, M., Weber, J., Hailemariam, D., Forestier, G.: Estimating time series averages from latent space of multi-tasking neural networks. Knowledge and Information Systems **65**(11) (2023) 4967–5004

5. Pialla, G., Ismail Fawaz, H., Devanne, M., Weber, J., Idoumghar, L., Muller, P., Bermgeir, C., Schmidt, D., Webb, G., Forestier, G.: Smooth perturbations for time series adversarial attacks. In: Pacific Asia Conference on Knowledge Discovery and Data Mining (PAKDD), Springer (2022) 1–12

6. Ismail Fawaz, H., Forestier, G., Weber, J., Idoumghar, L., Muller, P.A.: Deep learning for time series classification: a review. Data mining and knowledge discovery **33**(4) (2019) 917–963

7. Capecci, M., Ceravolo, M.G., Ferracuti, F., Iarlori, S., Monteriu, A., Romeo, L., Verdini, F.: The kimore dataset: Kinematic assessment of movement and clinical scores for remote monitoring of physical rehabilitation. IEEE Transactions on Neural Systems and Rehabilitation Engineering **27**(7) (2019) 1436–1448

8. Vakanski, A., Jun, H.p., Paul, D., Baker, R.: A data set of human body movements for physical rehabilitation exercises. Data **3**(1) (2018) 2

9. Devanne, M., Nguyen, S.M., Rémy-Néris, O., Le Gals-Garnett, B., Kermarrec, G., Thépaut, A.: A co-design approach for a rehabilitation robot coach for physical rehabilitation based on the error classification of motion errors. In: Workshop on Collaboration of Humans, Agents, Robots, Machines and Sensors (CHARMS), in conjunction with IEEE IRC, IEEE (2018) 352–357

10. Su, C.J., Chiang, C.Y., Huang, J.Y.: Kinect-enabled home-based rehabilitation system using dynamic time warping and fuzzy logic. Applied Soft Computing **22** (2014) 652–666

11. Lin, J.F.S., Karg, M., Kulić, D.: Movement primitive segmentation for human motion modeling: A framework for analysis. IEEE Transactions on Human-Machine Systems **46**(3) (2016) 325–339

12. Vakanski, A., Ferguson, J., Lee, S.: Mathematical modeling and evaluation of human motions in physical therapy using mixture density neural networks. Journal of physiotherapy & physical rehabilitation **1**(4) (2016)

13. Wang, Z., Yan, W., Oates, T.: Time series classification from scratch with deep neural networks: A strong baseline. In: 2017 International joint conference on neural networks (IJCNN), IEEE (2017) 1578–1585

14. Lee, M.H., Siewiorek, D.P., Smailagic, A., Bernardino, A., Badia, S.B.i.: Learning to assess the quality of stroke rehabilitation exercises. In: Proceedings of the 24th International Conference on Intelligent User Interfaces. (2019) 218–228

15. Liao, Y., Vakanski, A., Xian, M.: A deep learning framework for assessing physical rehabilitation exercises. IEEE Transactions on Neural Systems and Rehabilitation Engineering **28**(2) (2020) 468–477

16. Ismail Fawaz, H., Lucas, B., Forestier, G., Pelletier, C., Schmidt, D.F., Weber, J., Webb, G.I., Idoumghar, L., Muller, P.A., Petitjean, F.: Inceptiontime: Finding alexnet for time series classification. Data Mining and Knowledge Discovery **34**(6) (2020) 1936–1962

Evaluation and Improvement of State-of-the-art RADAR Object Detection Networks

Deniz Braun[1], Niclas Zeller[2]

[1] Karlsruhe University of Applied Sciences
Brde1020@h-ka.de
[2] Karlsruhe University of Applied Sciences
Niclas.Zeller@h-ka.de

Keywords: Autonomous Driving; Object Detection; Camera-Radar-Fusion.

1 Introduction

Autonomous driving has seen rapid development in recent years and is considered one of the most promising technologies that could revolutionize the way we get around. However, for a vehicle to act autonomously, it must be able to accurately perceive and interpret its surroundings [1]. This is where camera-radar fusion (CRF) systems come into play, playing a crucial role in providing accurate information about the vehicle's environment.

In the past, cameras and radar sensors were mainly used as stand-alone systems to gather information. Cameras provide high-resolution visual data and enable detailed detection of objects and their features. On the other hand, radar sensors provide information about the distance, speed, and direction of objects by emitting electromagnetic waves at near the speed of light and measuring their reflections [2].

The combination of cameras and radar sensors makes it possible to use the strengths of both technologies and balance their weaknesses. By fusing the data, accurate and comprehensive information about the vehicle environment can be generated. This is crucial for autonomous driving, as it allows the vehicle to make precise decisions and interact safely with its environment.

The importance of camera-radar fusion systems in the context of autonomous driving lies in their ability to ensure reliable perception of the environment in the optimum case. By combining the advantages of cameras and radar sensors, the system can accurately detect and track obstacles, pedestrians, traffic signs and other vehicles. It enables improved object detection, classification, and localization, even under suboptimal weather conditions or in situations with limited visibility [3].

In addition, camera-radar fusion systems help to increase safety in autonomous driving. They minimize the risk of misinterpretation and provide a robust basis for autonomous vehicle decision making. By providing a comprehensive perception of the vehicle's environment, they help detect potentially dangerous situations at an early stage and react accordingly.

Overall, camera-radar fusion systems play a crucial role in the further development of autonomous driving. They help to improve the performance of vehicle perception, enhance safety, or ultimately promote the acceptance of autonomous vehicles in society. Continuous further development and optimization of these fusion systems are therefore of great importance in making the vision of a fully autonomous vehicle fleet a reality.

2 Contents.

In this comprehensive scientific paper, we present a detailed investigation and comprehensive analysis of several radar object detection networks that represent the current state of the art. [4][5][6]

In doing so, radar data will be analyzed using a CNN to obtain precise and reliable information about detected objects.

This approach needs to be further explored in the following to make a qualitative statement about its advantage over radar detection through conventional and well researched signal processing.

In the course of our work, we will thoroughly analyze and evaluate the strengths and weaknesses of these networks. We will pay particular attention to a specific framework, which we will study intensively and make several modifications to optimize the existing pipeline in order to achieve significantly increased object detection accuracy.

Our overall goal is to define a unified data format for the neural network, which will serve as a fundamental basis for future research.

Furthermore, we would like to create the possibility to create and use our own camera radar data sets at the Karlsruhe University of Applied Sciences.

Through this approach, we aim to comprehensively improve the integration of camera-radar-fusion systems, thereby significantly increasing the overall performance of radar object detection.

3 References

1. https://www.adac-owl.de/wp-content/uploads/2018/03/Fahrerassistenzsysteme_Automatisiertes-Fahren_Sippl_OWL-1.pdf , 2018
2. https://ondosense.com/radar-know-how-radarsensoren-optimal-einsetzen/was-ist-radar-was-leisten-radarsensoren/ , 2023
3. Shanliang Yao, Runwei Guan, Xiaoyu Huang, Zhuoxiao Li1, Xiangyu Sha, Yong Yue, Eng Gee Lim, Hyungjoon Seo, Ka Lok Man, Xiaohui Zhu and Yutao Yue

"Radar-Camera Fusion for Object Detection and Semantic Segmentation in Autonomous Driving: A Comprehensive Review", 2023

4. Yizhou Wang, Zhongyu Jiang, Xiangyu Gao, Jenq-Neng Hwang, Guanbin Xing and Hui Liu "RODNet: Radar Object Detection using Cross-Modal Supervision", 2021

5. Ao Zhang, Farzan Erlik Nowruzi1, Robert Laganiere "RADDet: Range-Azimuth-Doppler based Radar Object Detection for Dynamic Road Users", 2021

6. Arthur Ouaknine, Alasdair Newson, Julien Rebut, Florence Tupin and Patrick Pérez "CARRADA Dataset: Camera and Automotive Radar with Range-Angle-Doppler Annotations", 2021

Development of an AI-based Energy Prediction for Electric Vehicles

Yannick Rauch[1], Nico Drobe[2], Tuyen Nguyen[1], and Reiner Kriesten[1]

[1] Karlsruhe University of Applied Sciences
Institute of Energy Efficient Mobility
`yannick.rauch@h-ka.de, tuyen.nguyen@h-ka.de, reiner.kriesten@h-ka.de`
[2] Karlsruhe University of Applied Sciences
Mechanical Engineering and Mechatronics

Abstract. The rise of electric vehicles offers new challenges as increasing energy efficiency and electric range through energy management systems as well as avoiding range anxiety by providing reliable range information. To realise this, predictive approaches based on a calculation of the expected energy consumption for the route to be driven are a feasible option. However, such approaches, realised through appropriate algorithms, usually require significant computing time, which can hinder their application. Therefore, this paper presents an approach to realise energy predictions using artificial intelligence (*AI*) approaches. The training of these *AI* models is performed with simulated data, generated by the algorithm to be replaced. Three *AI* models are build, trained, evaluated and optimised to predict a vehicle's energy consumption. A feed forward (*FNN*) and a recurrent neural network (*RNN*) model utilise deep learning approaches while a *XGBoost* model represents conventional machine learning techniques. In conclusion, the deep learning models struggle to match the results of the reference prediction algorithm, while the *RNN* model even fails to reduce calculation times. In contrast, the *XGBoost* model is able to generates accurate energy predictions, while drastically reducing the calculation time.

Keywords: Electric Vehicles, Energy Prediction, Energy Management

1 Introduction

With an increased awareness for energy and resource efficiency as well as stricter emission regulations a rise of hybrid- and all-electric vehicles occurred over the past years [1],[2],[3]. Energy management systems can provide smart solutions for the operation of these vehicles [4],[5]. Accordingly, increased electric ranges and energy efficiency can be achieved as well as a avoiding range anxiety by providing reliable range information [6]. To achieve this, predictive approaches that rely on prior knowledge of a route are a suitable solution [7],[8]. The resulting energy prediction can be used for the calculation of a route specific strategy to control energy consumption/distribution and/or to enable vehicle users to evaluate range capabilities to mitigate their range anxiety. For both use cases, a prediction must provide results with sufficient accuracy in acceptable computation time. Calculation methods based on the previous energy consumption from historical data can be used in short calculation times, but do not provide precise results. Empirical prediction methods can also be used, but require large and extensive data sets that are not widely available [8]. This leaves the use of physical and technical descriptions to create a model based prediction approach. However, these approaches to route

specific energy prediction cannot be computed in sufficient time and are therefore not applicable [9],[10]. This is especially the case when detailed models of a vehicle are used, which leads to an application of relatively simple calculation methods. In contrast, the best possible prediction of energy consumption for a specific route requires extensive and detailed models.

For a general applicability, the prediction should be able to provide detailed results on total energy consumption to derive range information and mitigate range anxiety. Furthermore the calculation of a predicted energy consumption profile, may be required, e.g. for implementing an energy control system. Accordingly, the energy prediction must calculate sufficiently accurate profiles of required data as well as overall correct totals, such as the total energy consumption. However, this must be done in acceptable calculation times so that the prediction can be integrated into a route planning process or used by energy management methods. In our context, a prediction algorithm based on an physical/technical model of vehicle, environment and driver is currently used to calculate the required energy consumption in the course of a route [8]. This algorithm provides sufficiently accurate results, but requires several seconds of computation time for execution and may therefore be too slow for an implementation in real vehicles or for acceptable user interaction.

Section 2 presents the approach pursued as well as the concept and developing method for an energy prediction artificial intelligence (AI). Subsequently, section 3 describes the implementation of AI energy prediction, focusing on evaluation and optimisation. In Section 4 the resulting models are examined based on three test scenarios. To conclude, Section 5 summarises this paper and provides a preview to future work.

2 Simulation Based Development of the Energy Prediction AI

Instead of the existing physical/technical based prediction algorithm, the introduced approach considers the use of machine learning methods to calculate the required prediction results, e.g. the energy consumption. This approach requires only an initial route specification to predict the corresponding energy consumption and is presented in Fig. 1.
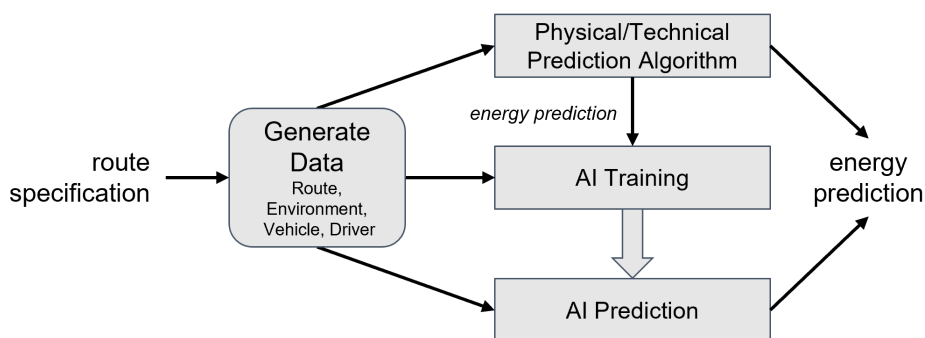

Fig. 1. Approach towards the use of machine learning for the prediction of a vehicle's energy consumption.

The input data for the *Artificial Intelligence* (AI) model is provided by a route information calculation [11], while the corresponding energy prediction for the AI training is provided by an energy and dynamic simulation algorithm [8]. Thus, the machine learning model is trained to replicate the simulated results and is expected to require significantly

less computing time than the calculation of the previously used prediction algorithm. Therefore, this approach uses a *MATLAB* implementation of the prediction algorithm. The required route data for this is generated by a self-developed *MATLAB* tool [11], that allows to provide various route information by accessing corresponding Web-*APIs* (*Application Programming Interfaces*). This information includes, but is not limited to, the geographical route date, gradients, traffic facilities and weather information. Based on this and a dataset for the vehicle and the driver, the physical/technical prediction algorithm generates the energy consumption profile over the course of a route. In order to reduce the dataset required to evaluate the general approach rather than producing a fully developed prediction, the driver and vehicle type is fixed and only the route and environment data are varied.

2.1 Development Method and Selection of the *AI* Models

To develop an *AI* model for energy prediction the method displayed in Fig. 2 is used. The necessary input data and energy prediction is generated by the algorithms implemented in *MATLAB* and then used for analysis and pre-processing according to the requirements of the chosen *AI* model. Subsequently the *AI* training loop is carried out, by running, evaluating and optimising the *AI* model.
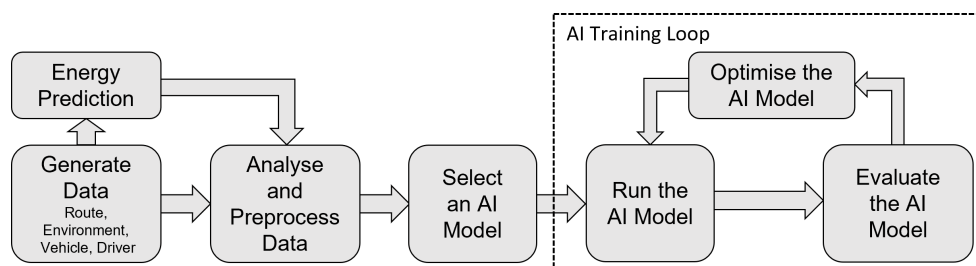

Fig. 2. Method for a simulation based development of a machine learning approach for energy prediction.

Data Generation and Preprocessing Through preprocessing and analysis of the generated data, representative datasets are created to be used in model training. The data generation process includes reading the generated input data and corresponding prediction results to normalise, standardise and categorise them to ensure their suitability for the *AI* models. The subsequent analysis of the data includes a plausibility check that assesses overall quality and integrity to ensure reliability of subsequent analyses and model training. In addition, insight into the characteristics of the data is gained for further development, e.g. through histograms, to understand distribution patterns.

Model Selection As a basic model, a feed forward neural network (*FNN*) is selected based on its successful use in various prediction tasks across different domains and its ability to handle complex relationships [12]. As the prediction of energy consumption involves intricate interactions between multiple variables, the capacity of a *FNN* to learn and model such complex patterns becomes particularly advantageous. With the availability of deep learning frameworks, e.g. *PyTorch* or *TensorFlow*, the implementation is also not that complex. In addition, the preparation of the training data is much easier compared to a recurrent neural network (*RNN*), which will also be used. This is due to

its specific design to capture dependencies within sequential data, which is the case for energy and dynamic behaviour of a vehicle. In the context of predicting energy consumption, which mainly results from the dynamics and is influenced by past states, the ability of a *RNN* to retain and use information from previous inputs seems promising.

While the models already presented use deep learning methods, *XGBoost* was selected as a third model, which is based on decision trees and ensemble learning. This model offers efficiency and scalability making it suitable for dealing with large datasets and computationally intensive tasks. *XGBoost* can efficiently handle the complexity and deliver fast and reliable predictions, which is proven successful in various machine learning competitions and real-world applications. Its robustness and track record make it a popular choice for tackling diverse prediction problems, which was also demonstrated across a variety of domains [13].

2.2 Development of the *AI* Model

After selecting the *AI* models to be studied for their ability to predict the energy consumption of an electric vehicle, the training of these models is to be carried out. For this purpose, a framework for development has to be selected first. The training is realised by *AI Training Loop*, displayed in Fig. 2, which is executed through two steps. The first execution is done during the process of model building and training. The second loop run is performed with the results of an already trained *AI* model, which are to be evaluated and may lead to an optimisation of the model.

Framework To build an *AI* model the selection of an adequate framework ensures the efficiency of the model development. For deep learning approaches *PyTorch* and *TensorFlow* are among the most widely used frameworks and have gained significant popularity and a large user base [14]. For this work the implementation of the *FNN* and *RNN* models uses *PyTorch* as framework, due to its similarities to *Python* which is already used for preprocessing, while the *XGBoost* library itself provides all necessary resources for building the model.

Building and Training For training, evaluation and optimisation of a model the dataset is split into a training and validation set. This allows the calculation of an appropriate loss to prevent a bias towards the dataset used for training [15]. The choice of training to validation split ratio depends on the specific requirements and objectives of the project. In this case, a ratio of 70:30 is used to create a balance between a sufficient amount of data for training and an extensive amount of evaluation. This balance is crucial to avoid overfitting, where the model becomes too specialised to the training data and fails to generalise well to new instances not yet seen. Additionally, the data is randomly shuffled to achieve more equitable representation of different driving scenarios within the training and test datasets, which ensures that the model learns from a diverse range of situations. Furthermore, to accelerate the training process, scaling of the features is applied before finally training the *AI* model with the given dataset. For the implementation of *FNN* and *RNN* models, *ReLU* is used as activation function and *Adam* is used as optimiser.

Evaluation For evaluation of the trained *AI* model two methods are used. As a first step, the hold-out method is used to check whether the model generalises well. This involves

a single instance of data split to training and validation datasets as already described. As a second step cross-validation is used to gain a more comprehensive understanding of the model's performance. This involves multiple datasets of differently split training and validation data, which are used in the training process. To further evaluate the generalisation capabilities of the trained models several test scenarios are created that the model has not yet seen. For this, scenarios are created over different environments, including city, countryside, and highway settings. To quantify the discrepancy between the *AI* predicted and reference energy consumption values, the mean square error (*MSE*) and the total energy consumption of a route are used. Furthermore, a comparison of the computation time between the chosen *AI* models and the *MATLAB* simulation tool is performed to assess the calculation speed and efficiency.

Optimisation To optimise the *AI* models, both a systematic approach in terms of hyperparameters and a manual approach of feature engineering are considered. The hyperparameter optimisation is performed with a grid search algorithm, which iterates a set of parameters defined by the developer and evaluates the model performance for each variant. For optimisation the *GridSearchCV* algorithm is used, which combines cross-validation with hyperparameter tuning [16]. Accordingly, this brute force search can find the hyperparameter set that performs best over all possible parameter combinations, also taking into account the generalisation of the model. Furthermore, a optimisation of the model is to be achieved by adding features created by the developer, which are not presented in the given dataset but seem promising to increase the model performance.

3 Implementation

To develop the chosen *AI* models for energy prediction an implementation for training and executing these models is required. Furthermore, corresponding representative datasets are required. Therefore, test scenarios are defined and the resulting data is analysed to ensure data quality. Finally the features and labels for the creation of the *AI* models are defined so that evaluation and optimisation can be carried out.

3.1 Data Generation and Analysis

As a first step for building and training of the *AI* models test scenarios are defined and corresponding data is generated by the given *MATLAB* tool. Based on this data it is crucial to examine the data for sanity and reasonability.

Data Generation For generating the required test scenarios a variety of routes is created across various driving scenarios, including city, countryside, and highway routes, to provide diverse and representative conditions. The dataset should contain a diverse specification of route properties, such as road gradient, number of traffic lights, weather and wind conditions, season and daytime conditions as well as tyre configurations.

Sanity Checks and Histogram Analysis The examination of the generated data revealed some problems with the functionality of the *MATLAB* tool used, particularly related to its use for generating representative data sets for training *AI* models. For example the tool generates weather related data according to the actual conditions. However, a histogram analysis of these generated datasets shows that this does not lead

to a proper distribution of weather related data. Accordingly, the tool was adjusted to manually set weather conditions to achieve a diverse distribution of weather data. Another finding was that generated times for sunrise, sunset and actual time are difficult to interpret and therefore not suited well as features. However, this data is only used by the prediction algorithm to determine the daytime. As a consequence, the tool was changed to manually setting the daytime to be used as a feature. In addition, it is also important to check the distribution of the data, as was done for the given curve radius data. A noteworthy observation in this data plot is a significant concentration of data at a value of 5000 m, which indicates that the steering wheel is in a straight position. With the knowledge of this skewed distribution, an optimisation approach can be carried out.

Selection of Features and Labels After careful investigation on the provided data the features for building the *AI* model can be defined. The chosen features are tire type, ambient temperature, weather state, daytime, curve radius, gradient, air density, resulting wind speed, speed limits, tunnels and traffic signals. To represent the energy prediction capability, the electrical energy consumption is chosen as label.

3.2 Evaluation and Optimisation

After the successful building of the *AI* models, the evaluation and optimisation of these models are the next essential steps to improve overall performance and accuracy. The deep learning models will be evaluated using the loss curves calculated by the *MSE* and serve as a visual tool for understanding a model's learning behaviour.

Increasing Data Points For the first experiment, a *FNN* model, consisting of four hidden layers with 1024 neurons in each layer, a learning rate of 0.001 and 300 training epochs is implemented. This model is trained with a dataset of 100,000 data points and results in the loss curve displayed in Fig. 3.
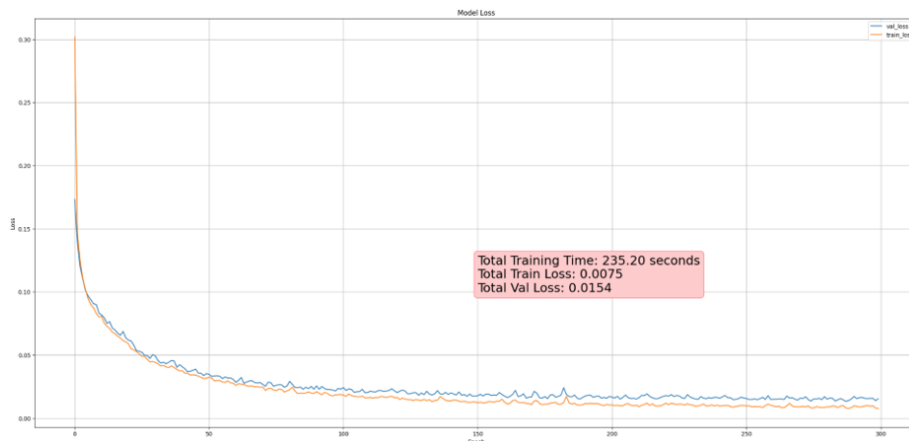

Fig. 3. Loss curve of a *FNN* experiment.

At first glance, the loss trend of the train and the validation data appears promising. However, for final evaluation the model was used to make a prediction on a previously unforeseen test scenario. This evaluation shows that the model falls short when applied to

previously unseen data, which could lead to the assumption that the model is overfitting. The corresponding loss curve in Fig. 3, however, has no clues, as the validation curve hardly deviates from the training curve. Therefore, the assumption is made that the number of training points is not sufficient.

Evaluation of the Amount of Training Data Establishing the precise number of data points necessary to build an accurate *AI* model presents a challenge since it highly depends on the specific problem at hand. Therefore, the *scikit*-learn *Python* library provides a method where different amounts of data points are tested via cross validation to investigate, whether an increase in data points leads to an improvement in the model's predictive accuracy. Fig. 4 presents a visual representation of this method, applied to an *XGBoost* model by using the *MSE* as a metric to evaluate the model's accuracy, using the total energy consumption as a target variable. The used *XGBoost* model consists of 100 decision trees with a maximum depth of 5 for each tree, while a learning rate of 0.1 and a 5-fold cross validation are used. This example shows, that between 50,000 and 150,000 data points the curve still drops significantly, while past the 300,000 data points mark, no significant changes are observable.
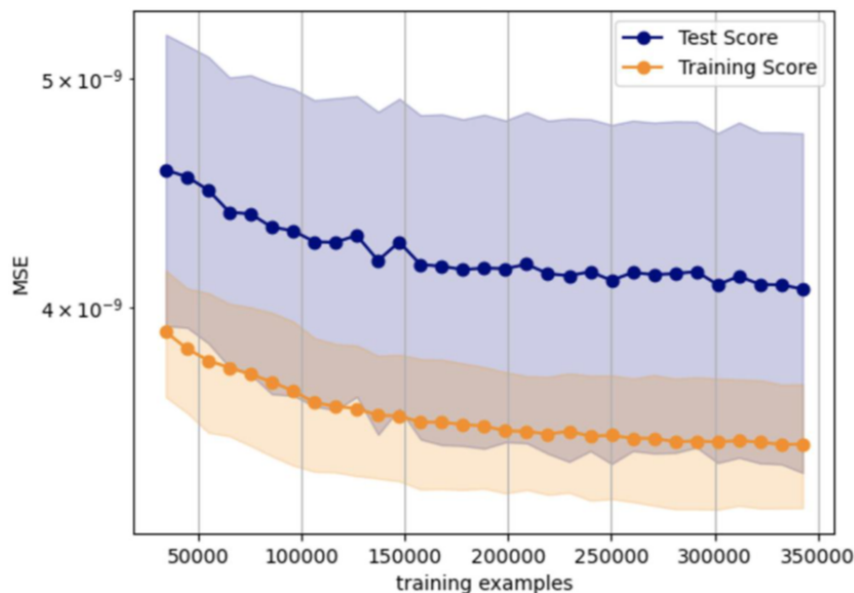
Fig. 4. *XGBoost* learning curve by evaluating different sizes of training examples.

Optimising with *GridSearchCV* To optimise the hyperparameter the *GridSearchCV* algorithm provided by *scikit*-learn is used. As this functions utilises brute-force iteration of the hyperparameters and combines this with an application of cross-validation it should be noted that it is a time intensive calculation. However, the optimisations that can be achieved with this are limited, especially compared to other optimisation approaches such as feature engineering. For example, when applied to an *FNN* model, the training loss is only improved by 10.68 % and the validation loss is only improved by 4.62 %, while the training time triples.

Feature Engineering The evaluation of the trained *AI* models as well as the knowledge of the used prediction algorithm shows that the representation of traffic signals can

have a significant influence on the resulting prediction. Therefore, a new feature called *distance_trafficSignals* is introduced, which describes the distance to the next traffic signal in order to represent the approaching behaviour of a vehicle. To evaluate, whether the new feature provides an advantage a *FNN* model consisting of six hidden layers with 1024 neurons in each layer, a learning rate of 0.0001 and 300 training epochs is used. With the inclusion of the new feature, a remarkable improvement of 36.72 % in training loss and an enhancement of 37.53 % in validation loss is achieved, while there was only a marginal increase in the total training time.

Optimising by Data Scaling Adjustment Based on the analysis of the histograms, it was suggested that different methods may be useful for scaling the data. Accordingly, two experiments are conducted to determine whether the use of a quantile transformation leads to an improvement in model accuracy compared to the use of standardisation. To carry this out a *FNN* model consisting of six two layers with 512 neurons in each layer, a learning rate of 0.0001 and 200 training epochs is used. This experiment shows, that the use of a quantile transformation as data scaling technique results in a reduction of 16.57 % of training loss and 11.97 % of validation loss.

Optimising the *RNN* model The evaluation and optimisation of the *RNN* model is of particular interest, as different sequence lengths have a significant impact on the performance. Accordingly, the sequence length is varied during the evaluation process to gain comprehensive insights how this factor influences the model's ability to learn, generalise, and predict. To speed up the experimentation, a simplified RNN model consisting of two layers with 128 neurons in each layer, a learning rate of 0.001 and 100 training epochs is used. As a baseline the *RNN* is essentially transformed to a *FNN* by using a sequence length of one and is compared to a RNN with a sequence length of six. This shows significant improvements regarding training and validation loss, while tripling the time required for training. A further increase to a sequence length of ten does not lead to another improvement but reveals issues with the simplified model. As a consequence the complexity of the *RNN* is increased to four hidden layers with 512 neurons in each layer, a learning rate of 0.0001 and 200 training epochs. Using this model, it can then be shown that a further increase in the sequence length does not offer improvements, which is why a sequence length of ten will be used for the *RNN* model.

3.3 Resulting Models

After applying the evaluation and optimisation activities to the models, this results in three descriptions for further use in discussion. To ensure a fair comparison between the deep learning models, the *FNN* and *RNN* model are configured with identical hyperparameters. The models consist of four hidden layers with 512 neurons in each layer, while training is done in 200 epochs with a learning rate of 0.0001. As already mentioned, the *RNN* uses a sequence length of 10. In contrast, the optimised *XGBoost* model employs different parameters It consists of 400 decision trees with a maximum depth of 18 for each tree and uses a learning rate of 0.1.

4 Evaluation and Discussion

The evaluation and discussion of the developed *AI* models is based on the application to 3 test scenarios that are unknown to the models. For evaluation the calculation time of

the prediction models as well as the accuracy of the resulting energy consumption profile and corresponding total are considered and compared with the originally used *MATLAB* prediction algorithm.

4.1 Test Scenarios

For evaluation of the energy prediction capabilities of the *AI* models three test routes representing a specific route type are used. The first scenario is an urban route with a length of approximately 5 km, 11 traffic lights and mainly driven at a speed of 50 km/h. It is simulated during the day with summer tyres under overcast skies and an ambient temperature of 11 ˚C. The second scenario is a route with a significant highway part, predominantly high speed sections over 100 km/h over a distance of around 26 km with only 6 traffic lights. The simulation is based on broken clouds during the day, an ambient temperature of 14 ˚C and winter tyres. A third scenario is used, representing a countryside journey, usually at a speed of over 70 km/h over a distance of approximately 22 km and significant differences in altitude between 633 and 1233 metres above sea level. The journey is simulated using winter tyres at an ambient temperature of 13 ˚C and scattered clouds during the day.

4.2 Evaluation

The energy prediction results of the *MATLAB* prediction algorithm used as a reference and the results of the three *AI* models are presented for each test scenario in Tables 1, 2 and 3. Across all three routes, the models produce different results. On the city route, the *FNN* slightly under predicts the energy consumption by approximately 7 % compared to *MATLAB*, while both *RNN* and *XGBoost* overpredict, with differences of 15.4 % (*RNN*) and 11.7 % (*XGBoost*) respectively. For the countryside route, the differences between *MATLAB* and the *AI* models are more significant. The *FNN* notably predicts negative energy consumption, while the *RNN* significantly underestimates it by 40.5 %. In contrast, *XGBoost* slightly overpredicts the energy usage by 5.2 %. On the highway route, all three models underestimate energy consumption to varying degrees, with the prediction of *XGBoost* again coming closest to the reference, but still underestimating by 13 %.

Table 1. Evaluation of characteristics of test scenario 1 (urban route).

	MATLAB Algorithm	*FNN*	*RNN*	*XGBoost*
MSE of Energy Consumption	-	0.069 Wh	0.064 Wh	0.059 Wh
Total Energy Consumption	0.466 kWh	0.434 kWh	0.538 kWh	0.521 kWh
Calculation Time	3.66 s	0.34 s	5.78 s	0.01 s

In general, *XGBoost* seems to offer more consistent results and appears to perform particularly well across different routes compared to the deep learning models. Furthermore, it is notable that for all three routes, the *XGBoost* model achieved the lowest *MSE* values, indicating that its predictions are closer to the reference values. In terms of calculation time, the *XGBoost* model again stands out among the three *AI* models. Not only does the model provide the most accurate predictions, it also has by far the shortest

calculation times. For all three routes, the calculation time to run the *XGBoost* model is measured in the order of hundredths of a second. In comparison, the *MATLAB* algorithm and the *RNN* model take several seconds and the calculation of the *FNN* model takes at least a few milliseconds.

Table 2. Evaluation of characteristics of test scenario 2 (highway route).

	MATLAB Algorithm	*FNN*	*RNN*	*XGBoost*
MSE of Energy Consumption	-	0.110 Wh	0.082 Wh	0.054 Wh
Total Energy Consumption	4.270 kWh	3.048 kWh	3.173 kWh	3.713 kWh
Calculation Time	7.75 s	1.36 s	29.57 s	0.03 s

Table 3. Evaluation of characteristics of test scenario 3 (countryside route).

	MATLAB Algorithm	*FNN*	*RNN*	*XGBoost*
MSE of Energy Consumption	-	0.160 Wh	0.077 Wh	0.038 Wh
Total Energy Consumption	2.341 kWh	-0.211 kWh	1.393 kWh	2.462 kWh
Calculation Time	6.93 s	1.17 s	22.81 s	0.02 s

When comparing the prediction of energy consumption profiles, Fig. 5 shows that all models have problems with the accurate prediction of energy peak values. In particular, the models tend to underestimate the peaks, especially in stop-and-go scenarios, such as red traffic lights. In some cases, the models also shift the energy peaks, indicating an incorrect prediction of the traffic light condition.

4.3 Discussion

In conclusion, the *XGBoost* model, which represents the conventional machine learning approach, achieves the best results in terms of predictive accuracy and calculation time. Regarding computational efficiency, the *FNN* model outperforms the *RNN* model, illustrating a trade-off between predictive accuracy and computational speed. However, it should also be noted that due to the recursive nature of the *RNN* model, calculation times increase drastically for longer distances. This issue disqualifies the use of the *RNN* model as a replacement of the *MATLAB* algorithm. Nevertheless, with the *FNN* model, there is a risk that the prediction is significantly inaccurate, such as in the prediction for the countryside scenario, which could also hinder the use of this approach.

5 Conclusion

With the motivation to use detailed models for predicting energy consumption for specified routes, an approach is pursued that uses machine learning to reproduce these results in less computing time. Towards this, two deep-learning models and one model based on
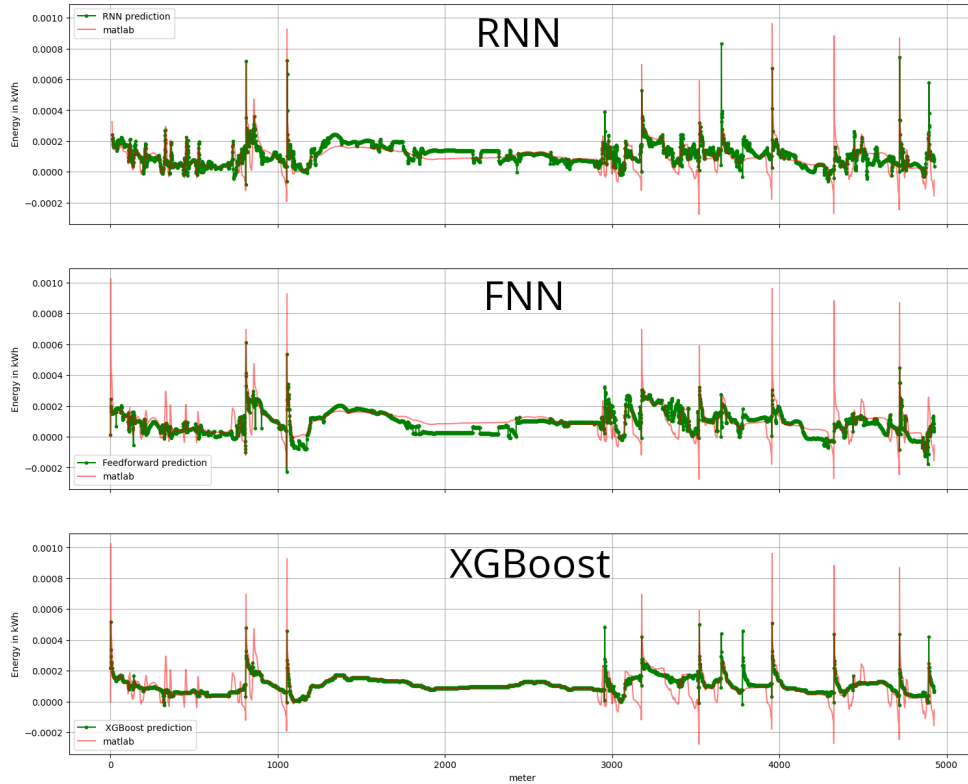
Fig. 5. Predicted energy consumption of test scenario 1 (city route) of the *MATLAB* prediction compared to the prediction of the *AI* models.

traditional machine learning are used. As deep learning models, both a *FNN* as a simple approach and a *RNN* model to account for the recursive nature of energy prediction are used. In addition, a *XGBoost* model represents a conventional machine learning approach using decision trees and ensemble learning methods. All three models are trained, evaluated and optimised to predict the electrical energy consumption of an electric vehicle based on given route and environment characteristics. The corresponding training data is generated by a *MATLAB* prediction algorithm, which is to be replaced by the *AI* models. However, the results show that the deep learning models struggle to match the results of the reference prediction algorithm. In contrast, the *XGBoost* model using a conventional approach successfully generates accurate energy predictions, while drastically reducing the calculation time. To conclude, it is shown that it is possible to make an energy prediction with an alternative method to the use of a physics-based algorithm.

The first assessment of the models is to be expanded in future work. On the one hand further evaluation is to be carried out, comparing the predictions of the *AI* models to measured energy consumption. On the other hand the scope of the *AI* models prediction can be expanded, e.g. by predicting further energy and dynamic quantities.

Acknowledgment

References

[1]   Kraftfahrtbundesamt. *Anzahl der Hybridautos in Deutschland von 2006 bis 2023*. Edited by Statista. 2023. URL: `https://de.statista.com/statistik/daten/studie/265993/umfrage/anzahl-der-hybridautos-in-deutschland/` (visited on 07/14/2023).

[2]   Kraftfahrtbundesamt. *Anzahl der Elektroautos in Deutschland von 2006 bis April 2023*. Edited by Statista. 2023. URL: `https://de.statista.com/statistik/daten/studie/265995/umfrage/anzahl-der-elektroautos-in-deutschland/` (visited on 07/14/2023).

[3]   BMU. *Das System der CO2-Flottengrenzwerte für Pkw und leichte Nutzfahrzeuge*. Edited by Bundesministerium für Umwelt, Naturschutz und nukleare Sicherheit. 2020.

[4]   A. Basler. "Eine modulare Funktionsarchitektur zur Umsetzung einer gesamtheitlichen Betriebsstrategie für Elektrofahrzeuge". Institut für Fahrzeugsystemtechnik. Dissertation. Karlsruhe: Karlsruher Institut für Technologie, 2015. 246 pages.

[5]   K. Çağatay Bayindir, M. A. Gözüküçük, and A. Teke. "A comprehensive overview of hybrid electric vehicle: Powertrain configurations, powertrain control techniques and electronic control units". In: *Energy Conversion and Management* 52.2 (2011), pages 1305–1313.

[6]   K. Kruppok, C. Gutenkunst, R. Kriesten, and E. Sax. "Prediction of energy consumption for an automatic ancillary unit regulation". In: *17. Internationales Stuttgarter Symposium*. Edited by M. Bargende, H.-C. Reuss, and J. Wiedemann. Proceedings. Wiesbaden: Springer Fachmedien Wiesbaden, 2017, pages 41–56.

[7]   K. Kruppok. *Analyse der Energieeinsparpotenziale zur bedarfsgerechten Reichweitenerhöhung von Elektrofahrzeugen*. ger. 1st ed. Kruppok, Kurt (VerfasserIn). Tübingen: expert verlag, 2020. 293 pages.

[8]   C. Gutenkunst. "Prädiktive Routenenergieberechnung eines Elektrofahrzeugs". de. Dissertation. Karlsruhe: Karlsruher Institut für Technologie, 2020. 214 pages.

[9]   R. Oberfell. "Stochastische Simulation von Energieflüssen im Nutzfahrzeug Ein einsatzorientiertes Bewertungs- und Optimierungsverfahren. Ein einsatzorientiertes Bewertungs- und Optimierungsverfahren". Institut für Fahrzeugsystemtechnik. Dissertation. Karlsruhe: Karlsruher Institut für Technologie, 2015. 344 pages.

[10] S. F. Tie and C. W. Tan. "A review of energy sources and energy management system in electric vehicles". In: *Renewable and Sustainable Energy Reviews* 20 (2013). PII: S1364032112006910, pages 82–102.

[11] T. Nguyen and Y. Rauch. "Real Route Generation for Simulation Based Developement". In: *Reports on Energy Efficient Mobility – Volume 2*. Edited by D. Feßler et al. 2nd edition. Karlsruhe: Zenodo, 2022, pages 58–64.

[12] S. Murat H. "A brief review of feed-forward neural networks". In: *Communications Faculty Of Science University of Ankara* 50.1 (2006). PII: commua1-2, pages 11–17.

[13] T. Chen and C. Guestrin. "XGBoost: A Scalable Tree Boosting System". In: 11.34 (2016). KDD'16 changed all figures to type1, pages 785–794.

[14] R. O'Connor. *PyTorch vs TensorFlow in 2023*. Edited by AssemblyAI Inc. 2021. URL: `https://www.assemblyai.com/blog/pytorch-vs-tensorflow-in-2023/` (visited on 10/13/2023).

[15] V. R. Joseph. "Optimal Ratio for Data Splitting". In: *Statistical Analysis and Data Mining: The ASA Data Science Journal* 15.4 (2022). Statistical Analysis and Data Mining: The ASA Data Science Journal, 2022, pages 531–538.

[16] scikit-learn developers. *scikit GridSearchCV*. 2023. URL: `https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html` (visited on 10/14/2023).

Facilitating the Adoption of AI Technologies for SMEs Using an Expanded Version of the Periodic Table of AI

Simone Braun, Tobias Hagen, and Damian Läufer

IMLA – Institute of Machine Learning and Analytics
Offenburg University of Applied Sciences
{simone.braun|tobias.hagen|damian.laeufer}@hs-offenburg.de

Abstract. In this paper we present the concept of the "KI-Labor Südbaden" to support regional companies in the use of AI technologies. The approach is based on the "Periodic Table of AI" and is extended with both new dimensions for sustainability, and the impact of AI on the working environment. It is illustrated on the basis of three real-world use cases: 1. The detection of humans with low-resolution infrared (IR) images for collaborative robotics; 2. The use of machine data from specifically designed vehicles; 3. State-of-the-art Large Language Models (LLMs) applied to internal company documents. We explain the use cases, thereby demonstrating how to apply the Periodic Table of AI to structure AI applications.

Keywords: AI use cases, Periodic Table of AI, KI-Labor Südbaden

1   Introduction

Recent economic studies suggest that AI and related technologies have enormous potential to add value to the German economy. [1]. It is expected, particularly through advances in new technologies from generative AI, that an increasing number of tasks in a variety of professions can be supported or automated by AI technologies. This will not only lead to significant productivity gains, but also to the transformation of daily work routines for many people.

However, this expected value contribution of AI-based systems, despite their advantages, has not yet been currently seen in implementation. Especially small and medium sized businesses face challenges in integrating AI systems within the current workforce, internal processes, and business strategy. The lack of comprehension and expertise in implementing best practices further complicates and hinders commencement of AI system integration [2, 3]. Publically funded programs, through facilitation services, are striving to support SMEs with digitalization assistance as well as with access to AI expertise.

The "KI-Labor Südbaden"[1] is part of a network of 16 "AI Labs" in the state of Baden-Württemberg and has a strong regional focus to support small and medium companies in the Upper Rhine region. In this paper, we present the concept of the "KI-Labor Südbaden" to support regional companies in the adoption and use of AI technologies. As one part of our activities, we developed a concept based on the "Periodic Table of AI" [4, 5]. The contribution is structured as follows: First we discuss the current state of the concept of the Periodic Table of AI and provide background information on AI within work systems. We then describe three real-world AI use cases from different domains

---

[1] https://ki-suedbaden.de/

and using different AI technologies that were co-developed with local businesses through the KI-Labor Südbaden project. Next, we demonstrate how the Periodic Table of AI can be applied and illustrate the various challenges companies are facing when they start to use AI technologies. After that, we present the current state of our extended version of the periodic table of AI, and apply it to the three use cases, before concluding with an outlook on future work.

2 Background

2.1 Periodic Table of AI

In 2016, Kristian Hammond [4] published the idea of the "Periodic Table of AI" (PTAI) as a structural and conceptual framework for AI functionalities, independent of specific technologies. An an analogy to the periodic system of elements, the PTAI consists of 28 basic elements that can be combined like LEGO® bricks for assembly into complex AI use cases and applications. Examples of such elements are *Speech Recognition (Sr)*, *Predictive Inference (Pi)*, or *Decision Making (Dm)* (see Fig.1). The elements are organized into three groups: 1) *Assess* – Receiving and recognising information, 2) *Infer* – Processing, inferencing and learning from that information, and 3) *Respond* – Taking actions. Fig. 1 shows the elements of group *Assess* in blue, elements of group *Infer* in green and orange, and elements of group *Respond* in red.

- The group *Assess* is comprised of the following recognition and identification elements: Speech recognition (Sr), Audio recognition (Ar), Face recognition (Fr), Image recognition (Ir), General recognition (Gr), Speech identification (Si), Audio identification (Ai), Face identification (Fi), Image identification (Ii), General identification (Gi), Data analytics (Da), and Text extraction (Te).
- The group *Infer* includes: Predictive inference (Pi), Explanatory inference (Ei), Synthetic reasoning (Sy), Planning (Pl), Problem solving (Ps), Decision making (Dm), Language generation (Lg), Language understanding (Lu), Relationship learning (Lr), Category learning (Lc), and Knowledge refinement (Lt).
- The group *Respond* consists of the elements: Mobility large (Ml), Mobility small (Ms), Manipulation (Ma), Communication (Cm), and Control (Cn).

When selecting at least one element from each group, this results in an "AI element triple" that might inform an AI-driven use case or application [4]. In that way, the PTAI facilitates the conceptual configuration of such use cases, resulting in a high-level
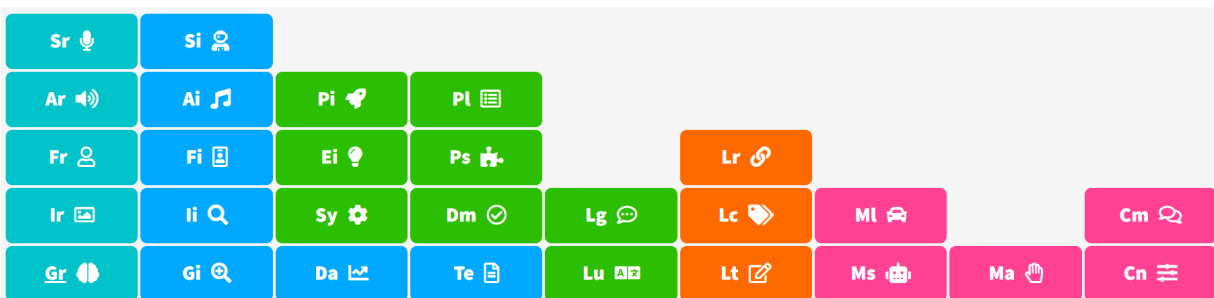

Fig. 1. Periodic Table of AI [5, 4]. *Assess* elements in blue, *Infer* in green and orange, and *Respond* in red

.

description of the use case or application; this description reflects the complexity and technological requirements [6].

In addition, the PTAI can serve as an educational tool for both digital experts and non-experts, providing fundamental knowledge of AI functionalities and thereby facilitating communication across varying levels of AI proficiency. Consequently, PTAI supports the comprehension of AI by both technical experts and non-experts. For example, Mylo et al. [7], applied the method within a project between machine learning experts and military operators, and demonstrated that it fostered effective communication and collaboration between these two groups [5, 6, 8].

In 2018, Germany's digital association "Bitkom" echoed Hammond's call for further development, and elaborated on the description of the elements and their use in practice [5]. However, as stated by both Hammond and Bitkom, there is still room for improvement in the PTAI. For example, there are missing elements for speech or image generation (while language generation is included) [5]. The user study by Mylo et al. [7] revealed misconceptions arising from either broad element titles or confusion between "recognition" and "identification" in the *Assess* group of elements, suggesting that the PTAI could be improved in terms of its comprehensibility.

Dietzmann and Alt [8] developed PTAI v2. They reduced the PTAI to 25 elements and rearranged them: firstly, along the abilities of perception, processing, action, and learning (based on Russell and Norvig's [9] concept of the intelligent agent), and secondly, by complexity. For this, they extended the PTAI by assigning types of human intelligence defined by Gardner [10] to each element in order to express its complexity. The number of assigned intelligence types gives each element its degree of complexity. Based on the PTAI v2, Dietzmann [6] developed an additional AI application taxonomy, with a focus on the configuration of AI applications detailing functionalities and characteristics while using PTAI v2 for use case analysis. However, the approach by [6, 8] does not include the work system integration and it considers only aspects that affect processes when it comes to organizational impact.

2.2   AI in Work Systems

The successful integration of AI applications within businesses necessitates a comprehensive socio-technical system approach [11–15]. The importance of this is recently demonstrated by the second edition of the DIN/DKE AI standardization roadmap [16, p. 153-176], in which a separate chapter on socio-technical systems has been added. The socio-technical system theory aims to "describe and explain the behavior of organizations and their members while providing critical insights into the relationships among people, technology, and outcomes" [17, p. 66]. The socio-technical perspective suggests that the social and technical subsystems are interdependent and inseparable within a work system and thus determine its effectiveness. In that way, a socio-technical system approach for AI takes into account the development of both social and technical subsystems of work while in relation to a specific context / relevant external environment. The implementation of AI, as a technological change, can have either a positive or negative impact on the overall work system, thereby ultimately affecting individuals [13]. Fig. 2 illustrates the integration of AI technology within a larger socio-technical work system and its impact on the overall system and individual interactions [12]. While some research started to draw attention on the technology, tasks, and people components, the context – specifically the organization and environment components, as well as the interactions among all components – are under-investigated [18, 11].
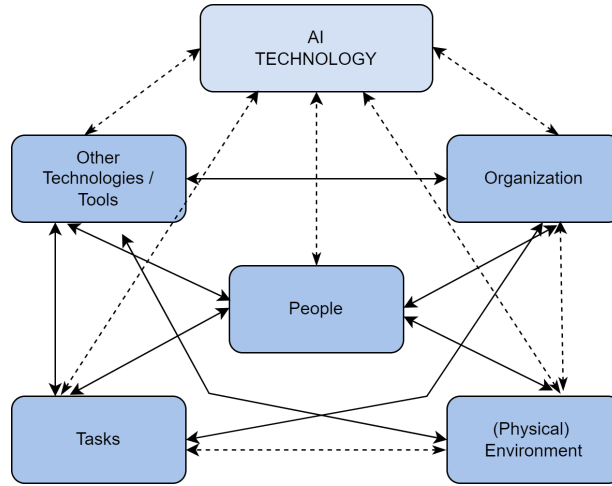
Fig. 2. AI integrated work system adapted from [12].

The DIN/DKE AI standardization roadmap [16, p. 153-176] proposes considering the socio-technical perspective along the complete AI life cycle; especially meeting sustainability criteria would be required. The roadmap sheds a first light on the AI life cycle with respect to the relevant socio-technical issues along the phases *Initiation*, *Design and Development*, *Verification and Validation*, *Transfer to the Operational Environment*, *Operation and Monitoring*, *Reevaluation* as well as *Continuous Validation and Decommissioning*. In addition, the need for standardization with regard to socio-technical aspects in the design of AI systems is identified.

von Garrel et al. [19] extend Hammond's general idea of structuring AI use cases. While the PTAI only focuses on the logical structuring of AI use cases from an AI-centric-perspective, von Garrel et al. focus on structuring AI use cases based on a socio-technical perspective, i.e. that the human perspective is taken into account. With this in consideration, the morphological box includes 14 characteristic features that describe the aspects of the AI use case, as well as how it affects the work system and its employees. Similarly, von Garrel et al. also call for the expansion and further development of the morphological box [19].

3 Use Cases

In this section, we explain the three real-world use cases, thereby demonstrating how to apply the PTAI to structure AI applications.

3.1 The Detection of Humans with Low-resolution IR Images for Collaborative Robotics

The first use case focuses on applying established methods from image processing and object recognition in a new environment. In addition to solving technical issues, this use case also has to address questions of labelled data acquisition, legal issues of data protection and privacy, and employee safety.

In general, this use case is about improving human-machine interaction in the ongoing production process and avoiding accidents between robots and humans in the production environment. To that end, the following tasks are necessary:

– **Step I:** Detecting humans in the working environment of a collaborative robot in a production hall,
– **Step II:** Calculating the probability of when a person is in the vicinity and approaching the robot, and deciding on the appropriate behavior of the robot,
– **Step III:** Initiating slowing down or stopping the robot.

Based on the decomposition into specific task steps and their description, suitable AI elements are mapped and put into context:

– **Step I:** In this specific case, low-resolution infrared images are used for cost-efficiency and data privacy reasons. Having a look at the PTAI, we can find *Assess*-element *Image Recognition (Ir)*. The image recognition functionality enables "the detection of certain object types in images or video signals" [5]. In this case, the object type sought is the human being.
– **Step II:** The results of the image recognition process are used to infer whether a detected person will approach the machine, and to determine the appropriate behavior of the robot. The *Infer*-element *Predictive Inference (Pi)* predicts events based on the current state of the environment. The *Decision Making (Dm)* element chooses a particular course of action or solution based on available information, alternative options, and a set of objectives. These courses of action range from avoidance to stop manoeuvres. While ensuring that employees are not put at risk is crucial, it is also important to maintain high productivity, thereby necessitating the avoidance of unnecessary stops or overly complicated evasive actions.
– **Step III:** Depending on the decision made beforehand, the autonomous control of the robot's movement would be enacted. The *Respond*-element deemed suitable for this purpose is referred to as *Mobility Small (Ms)*, and it effectively governs the movement of robots that operate within indoor environments while performing tasks and interacting with individuals.

3.2 The Use of Machine Data from Specifically Designed Vehicles

The second use case aims to make AI methods actionable for a regional manufacturer of specifically designed vehicles. Time series data from the vehicles is currently collected but not really used. The initial focus in this use case is on questions of data quality, data preparation / exploratory data analysis, and predictive maintenance. In the predictive maintenance use case, time series data such as engine speed and fuel consumption is used to predict defects before they occur. This specific use case again can be broken down into three separate steps:

– **Step I:** Analyze the data to extract the information it contains.
– **Step II:** Understand the relationship between the data and the presence of defects based on past states, to be able to predict future defects based on the current state.
– **Step III:** Pass on the information to the responsible personnel.

Mapping the AI elements:

– **Step I:** In this use case, the data must be analyzed in detail in order to extract the information it contains. The corresponding *Assess*-element is *Data Analytics (Da)*, which refers to the analysis of data for the identification of the current facts and events represented by the data.

- **Step II:** The second step uses time series data to build a predictive model. This model learns statistical relationships between the given features and the presence of defects. Therefore, the probability of a defect occurring in the near future could be predicted. The corresponding *Infer*-element is *Prediction Inference (Pi)*, which predicts future events based on known data about the current state.
- **Step III:** Finally, the information has to be communicated to the appropriate personnel (for example, by triggering an alert). This task is described by the *Respond*-element *Communication (CM)*. It is generally defined as a mechanism that supports communication between human and machine.

3.3 Chatting with Internal Company Documents Based on Current LLMs

Finally, the third use case deals with the use of current large language models (LLMs) within companies. The intention is to make internal company knowledge accessible to users via a Q&A chatbot. The main focus here is on reliability and accountability of the answers. As in the previous examples, we explain this use case based on the elements of the PTAI. The use case can be broken down into three separate steps:

- **Step I:** Extract information from the company's documents and the question posed,
- **Step II:** Understand the meaning and the context of the question and the document, and then answer the question in natural language based on the given information.
- **Step III:** Pass on the information to the user in natural language.

Mapping the AI elements:

- **Step I:** In this use case, it is necessary to accurately extract information from the company's documents and to extract the information from the user's question. This task is described by the *Asses*-element *Text Extraction (Te)*, which analyzes and extracts the information of a given text.
- **Step II:** The second step is divided into two separate subtasks. Firstly, it is necessary to understand both the question and the documents in order to be able to map related elements within the documents, and also to match the document information with the question's requested information. The matching *Infer*-element is *Language Understanding (Lu)*, which is responsible for understanding the meaning and the context of the texts by creating semantic representations. Secondly, the matching information of the documents has to be converted into natural language. The responsible *Infer*-element for this task is *Language Generation (Lg)*, which generates natural text from a logical knowledge representation.
- **Step III:** Finally, the information has to be communicated to the user, e.g. visually or acoustically. For this purpose, the *Respond*-element *Communication (CM)* is used. It is generally defined as a mechanism that supports communication between human and machine.

4 Extended Periodic Table of AI

After the original PTAI offered the possibility to logically structure AI use cases, and von Garrel et al. [19] extended the general idea by focusing on the socio-technical aspects of AI use cases, we combine these aspects and further extend the system with a sustainability perspective [16]. For this approach, the standard PTAI serves as a basis and is extended by two additional independent dimensions focusing on the impact on

the working environment and on sustainability. To represent the impact on the working environment, the elements of von Garrel et al.'s morphological box are adopted. The sustainability dimension is based on the UN's 17 sustainability goals [20]. The specific integration of the two added dimensions to form a cohesive framework is still a work in progress. Nevertheless, some implications for the working environment and for sustainability are illustrated below by reference to the three use cases previously introduced. For these examples we focus on the features, "nature of work", "level of autonomy", "mode of interaction", "social presence", "explainability/transparency", "change of the working system" and "user" from von Garrel et al.'s morphological box to examine the impact of the use cases on the working environment. To study the impact of these use cases on sustainability, we individually focus on a subset of the UN's 17 sustainability goals.

4.1 The Detection of Humans with Low-resolution IR Images for Collaborative Robotics

Working Environment The Infrared human detection is intended for use in collaborative robotics, i.e. the robot's goal is to navigate independently around objects and people, while assisting workers. Independent human avoidance makes the "nature of work" reactive and the "level of autonomy" "automated" or even "fully automated". The "user" is a worker and the "mode of interaction" can be classified as "physical", "collaborative" and "mobile". As the use case is based on object detection, it is possible to mark the object that is recognised as human, leading to at least a low level of "explainability/ transparency". Overall, the "change of the work system" is moderate. This is mainly due to the fact that collaborative robots are designed to be flexible [21, 22]. However, a major challenge might be employee acceptence of AI and collaborative robots as a tool, seen as something that supports them rather than something that replaces them.

Sustainability The Infrared human detection use case mainly relates to four of the 17 sustainability goals:

- Goal 5: Gender Equality
- Goal 8: Decent work and economic growth
- Goal 10: Reduced Inequalities

Gender equality and reduced inequalities: Gender equality concerns creating a level playing field for people regardless of gender, particularly in social, legal, political and economic spheres. The goal of 'reduced inequalities', particularly in sub-goal 10.3, is to ensure equal opportunities for all and to end discrimination. Injustice does not necessarily have to be caused by humans. Artificial intelligence, for example, can also build up prejudices and thus treat people unequally. For example, in the context of the given use case, people of a certain gender, body type, height or skin colour might be better recognised than others. In order to avoid such outcomes,countermeasures are appropriate, such as the use of low-resolution infrared images in combination with optimization methods to improve image quality.

Decent work and economic growth: This Sustainable Development Goal seeks both economic growth, and safe and decent working conditions. For example, Goal 8.8 calls for the protection of labour rights and the promotion of safe working environments. It should be mentioned that very monotonous work can also pose a health risk. For example, findings by Farmer and Sundberg [23] or Sommers and Vodanovich 2000 [24] suggest a link between perceived boredom at work and mental illnesses such as anxiety

and depression. Robots along with collaborative robots, as in our use case, can take over monotonous tasks and thus potentially contribute to reducing the perceived boredom of workers. Object recognition is used to recognise and avoid people. Therefore, it also helps to ensure safety in human-robot interaction.

4.2 The Use of Machine Data from Specifically Designed Vehicles

Working Environment In this use case, the data from the vehicles is used to predict impending failures. The relevant personnel are alerted if the current data indicates future failure. To do this, existing information is combined to draw logical conclusions, making the "nature of work" "combinative." The "level of autonomy" can be classified as "decision support," so that the "user," the relevant personnel, would be the final decider. This person is informed by means of displayed output, which makes the "mode of interaction" "conventional software". Especially in this use case, it is pertinent to also display the original data, in order to help the relevant personnel to make the correct decision. This leads to at least a low level of "explainability/ transparency". Overall, the "change of the work system" would be low, because the work in general would not change drastically. The conventional way of communication would also presumably facilitate employee acceptance of AI as a tool for improving their decisions.

Sustainability The sustainability goals that are primarily affected by this use case are:

- Goal 8: Decent work and economic growth
- Goal 12: Responsible consumption and production

Higher Level of productivity: The aim of predictive maintenance is to predict failures before they occur. This can increase productivity by reducing the downtime of the machines, which is in line with sub-goal 8.2. It calls for productivity to be increased through, among other things, the implementation of technological innovation.

Resource efficiency: Predictive maintenance not only increases productivity by reducing downtime, but it can also extend the lifespan of the machine by keeping it consistantly well maintained. This is in line with both sub-goal 8.2 and sub-goal 12.5. While 8.2 calls for greater resource efficiency and more sustainable consumption and production, 12.5 calls for the reduction of waste, including through prevention.

4.3 Chatting with Internal Company Documents Based on Current LLMs

Working Environment In this use case, a chatbot is used to inform clients about complicated contract documents. To gather more reliable information, the clients still need to consult an expert. The text generation from information in the documents can be seen as a "combinative" or even a "creative" "nature of work". Also, the chatbot works completely independently, which makes the "level of autonomy" fully automated. As mentioned above, the "user" is a "client". The chatbot communicates with the user via a user interface, which makes the "mode of interaction" conventional software. To enable the user to check the information in the original document, the referenced sections of the text are provided. This at least leads to a low level of "explainability/ transparency". Overall, the "change of the work system" is minor, because the work in general does not radically change. The expert will still be consulted when necessary. Again, the main challenge might be employee and client acceptance of the chatbot as a useful tool.

Sustainability The sustainability goal that is primarily affected by this use case is:

– Goal 8: Decent work and economic growth

Higher Level of productivity: By implementing the chatbot, many simple questions can be answered without consulting an expert. This allows the experts to focus on the cases where their expertise is actually needed. It is also likely to lower the inhibition threshold for clients to inquire about the document details, as expert consultation becomes required less often. Taken together, these two effects have the potential to increase productivity, which is in line with sub-goal 8.2.

5 Conclusions

The article explains the aforementioned use cases and demonstrates how the Periodic Table of AI can be used to structure AI use cases and applications. It illustrates how simple AI element triples can be created to configure a manageable, yet complex AI use case and application from elusive problem settings such as employee protection from accidents with collaborative robots. It also further facilitates the discussion between domain experts and digital / AI experts [6]. Furthermore, an extension of the PTAI is proposed that includes a dimension based on von Garrel et al.'s morphological box [19] and a dimension based on the 17 UN sustainability goals [20]. This concept helps to represent the whole impact of a specific AI use case, both inside and outside a business use case. The application of the two added dimensions is demonstrated with three real-world examples, providing a holistic view of each of the use cases. Further work focuses on the implementation of the adaptations proposed by Dietzmann et al. [6] for the conception and creation of a coherent extended PTAI framework.

References

1. IW Consult: Der digitale Faktor. `https://der-digitale-faktor.de` Accessed: 2023-09-26.
2. Chowdhury, S., Budhwar, P., Dey, P.K., Joel-Edgar, S., Abadie, A.: Ai-employee collaboration and business performance: Integrating knowledge-based view, socio-technical systems and organisational socialisation framework. Journal of Business Research **144** (2022) 31–49
3. von Garrel, J., Jahn, C.: Design Framework for the Implementation of AI-based (Service) Business Models for Small and Medium-sized Manufacturing Enterprises. Journal of the Knowledge Economy (March 2022)
4. Hammond, K.: The Periodic Table of AI. `https://www.xprize.org/prizes/artificial-intelligence/articles/the-periodic-table-of-ai` (2016) Accessed: 2023-07-12.
5. Balada, C., Bellanova, A., Brüst, R., Bruss, M., Buchberger, S., Cirullies, J., Corro, L.D., Festag, R., Fuhs, G., Goetze, S., Gressling, T., Hartmann, T., Havemann, M., Hoffart, J., Holtel, S., Hufenstuhl, A., Kraus, W., Pfleger, N., Pikus, Y., Plumbaum, T., Rolletschek, G., Satow, L., Schnakenburg, I., Siepmann, R., Steffner, R., Shozo, M.T., Weber, M., Wieczorek, S., Wittenburg, G.: Digitalisierung gestalten mit dem Periodensystem der Künstlichen Intelligenz. Technical report (2018)
6. Dietzmann, C.: Towards a framework for assessing the business impact of artificial intelligence. PhD thesis, Universität Leipzig (2023)

7. Mylo, M., Dykta, P., Schoepe, D.: A Periodic System of Artificial Intelligence as an Effective Means of Communication between Machine Learning Experts and Military Operators. In: Proceedings of the STO Information Systems and Technology (IST) Panel Symposium (RSM) on Artificial Intelligence, Machine Learning and Big Data for Hybrid Military Operations (AI4HMO) – STO-MP-IST-190, STO/NATO 2021 (October 2021)

8. Dietzmann, C., Alt, R.: Assessing the Business Impact of Artificial Intelligence. In: Proceedings of the Annual Hawaii International Conference on System Sciences, Hawaii International Conference on System Sciences (2020)

9. Russell, S., Norvig, P.: Artificial intelligence: A modern approach, global edition. 4 edn. Pearson Education, London, England (May 2021)

10. Gardner, H.: Intelligence reframed: Multiple intelligences for the 21st century. Basic Books, New York, NY, US (1999)

11. Niehaus, F., Wiesche, M.: A Socio-Technical Perspective on Organizational Interaction with AI: A Literature Review. In: Proceedings of the Twenty-Ninth European Conference on Information Systems (ECIS 2021). (2021)

12. Salwei, M.E., Carayon, P.: A Sociotechnical Systems Framework for the Application of Artificial Intelligence in Health Care Delivery. Journal of Cognitive Engineering and Decision Making **16**(4) (2022) 194–206 PMID: 36704421.

13. Holdsworth, M., Zaghloul, F.: The Impact of AI in the UK Healthcare Industry: A Socio-Technical System Theory Perspective. In: Proceedings of the 8th International Workshop on Socio-Technical Perspective in Information Systems Development. (2022) 19–20

14. Enholm, I.M., Papagiannidis, E., Mikalef, P., Krogstie, J.: Artificial Intelligence and Business Value: a Literature Review. Information Systems Frontiers **24**(5) (August 2021) 1709–1734

15. Latniak, E., Tisch, A., Kauffeld, S.: Zur Aktualität soziotechnischer Arbeits- und Systemgestaltungsansätze in Zeiten von Digitalisierung und KI. Gruppe. Interaktion. Organisation. Zeitschrift für Angewandte Organisationspsychologie (GIO) **54**(1) (March 2023) 1–8

16. DIN/DKE: Deutsche Normungsroadmap Künstliche Intelligenz. `https://www.dke.de/res ource/blob/2008010/776dd87a4b9ec18d4ab295025ccbb722/nr-ki-deutsch---downloa d-data.pdf` (2022) Accessed: 2023-09-22.

17. Kull, T.J., Ellis, S.C., Narasimhan, R.: Reducing Behavioral Constraints to Supplier Integration: A Socio-Technical Systems Perspective. Journal of Supply Chain Management **49**(1) (2013) 64–86

18. Safaei, D., Haki, K., Morin, J.H.: Artificial Intelligence in Information Systems Research: A Socio-technical Perspective. In: Proceedings of the 19th Conference of the Italian chapter of AIS & the 14th Mediterranean conference on information systems. (2022)

19. von Garrel, J., Jahn, C., Schröter, D.: Der Einsatz Künstlicher Intelligenz in produzierenden Unternehmen. Eine Morphologie industrieller, KI-basierter Arbeitssysteme. Zeitschrift für wirtschaftlichen Fabrikbetrieb **117**(5) (2022) 338–343

20. United Nations: THE 17 GOALS. `https://sdgs.un.org/goals` Accessed: 2023-10-09.

21. Sherwani, F., Asad, M.M., Ibrahim, B.: Collaborative Robots and Industrial Revolution 4.0 (IR 4.0). In: 2020 International Conference on Emerging Trends in Smart Technologies (ICETST). (2020) 1–5

22. Mihelj, M., Bajd, T., Ude, A., Lenarčič, J., Stanovnik, A., Munih, M., Rejc, J., Šlajpah, S. In: Collaborative Robots. Springer International Publishing, Cham (2019) 173–187

23. Farmer, R., Sundberg, N.D.: Boredom Proneness–The Development and Correlates of a New Scale. Journal of Personality Assessment **50**(1) (1986) 4–17 PMID: 3723312.

24. Sommers, J., Vodanovich, S.J.: Boredom proneness: Its relationship to psychological-and physical-health symptoms. Journal of clinical psychology **56**(1) (2000) 149–155

Enhancement of Visual SLAM Precision Using Semantic Image Segmentation for Automotive Systems

Patrick Rebling[1], Michael Weber[1,2], Tobias Weiß[1,2], Franck Gechter[2], Reiner Kriesten[1], and Philipp Nenninger[1]

[1] Institute of Energy Efficient Mobility (IEEM), Karlsruhe University of Applied Sciences
{patrick.rebling, michael.weber, tobias.weiss, reiner.kriesten, philipp.nenninger}@h-ka.de
[2] CIAD (UMR 7533), Université Bourgogne Franche-Comte, UTBM
LORIA-MOSEL (UMR 7503), Université de Lorraine
franck.gechter@utbm.fr

Abstract. In the field of advanced driver assistance systems (ADAS) testing and autonomous driving (AD) feature evaluation, novel approaches relying on augmented reality (AR) promise to deliver cost-saving benefits. These forward-looking approaches leverage vSLAM techniques to create mapping solutions that are essential for augmentation. A critical challenge, however, is maintaining the high precision required for these maps and, by extension, the SLAM algorithm itself. This precision is often compromised by the presence of false-positive detections of feature points. In response to this challenge, this paper presents an improvement to the ORB-SLAM3 algorithm. The proposed approach incorporates semantic segmentation without compromising processing speed to increase the precision and reliability of the SLAM system. This is to ensure that the integration of AR-based solutions in the automotive sector is both effective and sustainable, providing tangible benefits in the testing and development of ADAS and autonomous driving technologies.

Keywords: Artificial Intelligence, Augmented Reality, Advanced Driver Assistance Systems, Visual Simultaneous Localization and Mapping, 3-Dimensional Modeling, Image Segmentation, Object Detection

1 Introduction and Motivation

Understanding of the surrounding world of vehicles and robots is essential for automation [1, 2] and testing of automated systems using augmented reality (AR) [3]. While there are approaches that only use the current perception of the vehicle for planning , others are based on pre-built maps for localization and orientation. These maps are required for automated driving to interpret and analyse the current state of the environment and to navigate the vehicle efficiently. A common approach for map generation is the usage of *Light Detection and Ranging* (Lidar) which allows robust creating of maps and the relocalization inside those maps but the sensors are quite expensive. Pure visual approaches require only cameras for mapping and localization and are therefore cost-efficient and suitable for automotive and robotic applications. For visual approaches, it is necessary to have knowledge about meaningful feature points for relocalization. Therefore, we present an approach for enhancing visual Simultaneous Localisation and Mapping (vSLAM) in automotive applications by filtering of feature points for gaining robustness for AR-based testing of advanced driver assistance systems (ADAS).

2 State-of-the-Art

In this section, we will focus on the state-of-the-art in mapping and localization of vehicles and information extraction from images based on semantic segmentation. After a brief overview on techniques and approaches, the algorithms chosen for our experiments will be presented and their functionality will be explained.

2.1 Simultaneous Localization and Mapping (SLAM)

Various methods have been established to map the environment only based on images in a three-dimensional virtual map like Structure From Motion (SfM), vSLAM, and Visual Odometry (VO) [4]. This offers the possibility that no sensors other than the camera system are needed to create a 3D model of the environment. Thus, there is no need for costly sensors like Lidar or Radar. While VO attempts to reconstruct the path step by step, pose by pose and may optimize only the last n poses of the trajectory using a window-based bundle fitting approach, vSLAM is primarily concerned with ensuring the consistency of the global map. The framework of vSLAM algorithms is mainly composed of three basic modules: Initialization, Tracking, Mapping, and two additional modules: Re-localization and Global Map Optimization (including Loop Closing) [5]. For the use of SLAM in automotive vehicles and the associated properties such as fast scene changes and low texturing of the environment, various approaches are available using vSLAM [6, 7]. In this paper, we focus on the *ORB-SLAM3* [8] algorithm and show how its precision can be enhanced by using semantic image segmentation.

The ORB-SLAM Algorithm. The first version of ORB-SLAM was presented in 2015 and the current version, ORB-SLAM3, supports of different camera systems and performs better than comparable SLAM algorithms [7]. Here, ORB-SLAM3 represents a complete SLAM system for monocular, stereo, and RGB-D cameras which operates in real-time and achieves remarkable results in terms of accuracy and robustness in a variety of different environments. ORB-SLAM3 is used for indoor sequences, drones, and urban driving. It consists of three main parallel threads: Tracking, Local Mapping, and Loop Closing. This algorithm is a feature-based approach, which represents the detected points in a three-dimensional *MapPoint*. Initially, the approach was to use a mono-based camera and an *Inertial Measurement Unit* (IMU) for mapping. It was noticeable that the robustness of the relocalization and the estimation of the scaling are not sufficient due to the lack of depth information. Switching to stereo-based cameras allow more accurate mapping of the environment with more realistic scaling.

2.2 Image Segmentation

Image segmentation, a fundamental process in digital image analysis, involves dividing a digital image into different segments to simplify its complexity and enable subsequent analysis of the individual segments. In this technical process, labels are assigned to individual pixels, identifying objects, people, or other relevant elements in the image [9]. An important application of image segmentation is in object recognition [10]. Instead of processing the entire image as a whole, an image segmentation algorithm is often used to identify objects of interest in the image. Then, an object detector can operate within predefined bounding boxes set by the segmentation algorithm. With this approach, the detector does not have to analyze the entire image, which increases accuracy and reduces the time required for inference. Image segmentation plays a central role in the

field of computer vision technologies and algorithms and is widely used in various practical applications. These applications include medical image analysis, computer vision for autonomous vehicles, face recognition and detection, video surveillance, and satellite image analysis [11]. While there are several methods for image segmentation, Figure 1 illustrates the three main types.
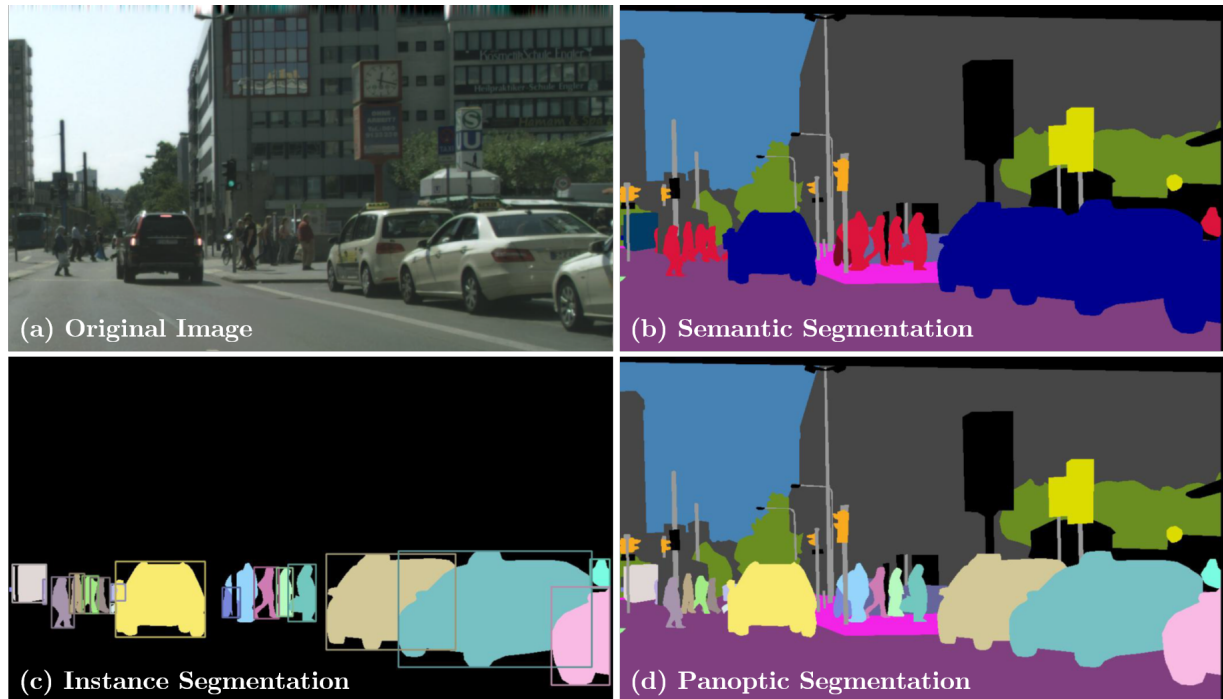

Fig. 1. Some of the main types for image segmentation. While semantic segmentation (b) does not highlight different instances of the same object type, instance segmentation (c) and panoptic segmentation (d) provides detailed information, e.g. for tracking of objects. [12, p. 1]

3 Problem Formulation

When testing ADAS through AR applications, an algorithm that can handle both mapping and tracking is a critical requirement. The challenges posed in this context are significantly more demanding than for conventional AR applications. This increased complexity arises primarily from two key factors. First, the mapping and tracking processes must exhibit exceptionally high precision, even at higher speeds. Second, these tasks must be performed within the test scenarios of the European New Car Assessment Programme (Euro NCAP) [13]. The unique test environments defined by Euro NCAP present a number of challenges, as they are very large and there are hardly any existing textures that can prevent the algorithm from finding its way. In response to this multi-faceted problem, our paper attempts to answer the following question:

> What is a Robust Simultaneous Localization And Mapping Algorithm for Mapping and Tracking to Test Camera-based Advanced Driver Assistance Systems in Low Texture European New Car Assessment Programme-Test Scenarios?

As shown on the left side in Figure 2, feature points are detected in the sky as well as on the hood of the ego vehicle, which hinder the performance of the vSLAM algorithm and causes wrong scaling of the environment inside generated maps. Experiments show that ORB-SLAM3 is struggling with the moving vehicle hood and therefore misunderstands the scales of the environment and static objects. To filter these false-positive feature points, two possible approaches are given: (1) optical flow, (2) a suitable image segmentation. As the difference between sky and distant background which provides indeed good orientation points will result in similar optical flow, we will focus on the more promising approach of image segmentation.
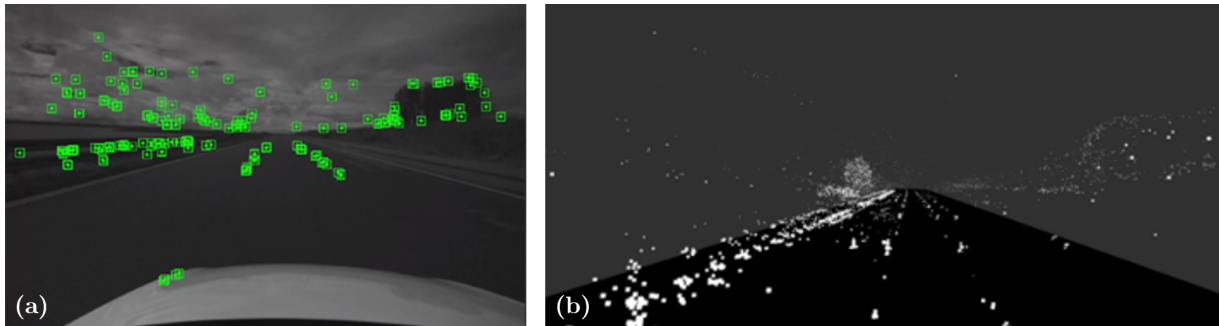
Fig. 2. ORB-SLAM3 detection results (a) and the resulting map (b) with plenty of false-positive feature points regarding the sky and the vehicle hood.

4 Approach

Our proposed solution can be divided into two core areas: 3D Global Point Cloud Map Generation, implemented using the ORB-SLAM3 framework, and the Implementation of Image Segmentation as shown in Figure 3 based on the idea of the DS-SLAM [14].
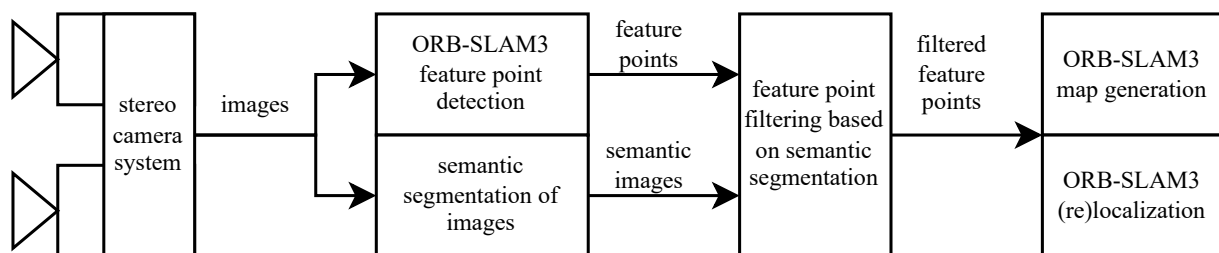
Fig. 3. Enhanced mapping pipeline of ORB-SLAM3. Parallel calculation of feature points and semantic segmentation is done on the raw images from the stereo camera. Afterwards feature points are filtered and feed into the ORB-SLAM3 map generation.

4.1 Suitable Image Segmentation Algorithm

Our initial foray into the literature revealed that we chose the *DeepLabV3+* model [15, 16]. DeepLabV3+, which uses the *Xception-65* model as its backbone [17], has great versatility in training on different datasets. The use of DeepLabV3+ trained on the *ADE20k*

dataset [18, 19] yields commendable results, accurately detecting the sky and road, albeit with some limitations, notably occasional inaccuracies in hood detection due to mirror effects and suboptimal road detection. Nevertheless, the processing speed optimized by *TensorRT* achieves a frame rate of about 30 fps, which meets the requirements of our particular use case.

Given the compelling need to correctly classify feature points within these regions, we explored alternative approaches. Xie et al. performed a comparative analysis of different neural networks for different datasets [20] shown in Figure 5. Their evaluation showed that *SegFormer* outperforms other options by providing a better balance between processing speed and accuracy, especially on the ADE20K dataset. SegFormer is shown to be robust to noise, fog, blur, snow, frost and splatter, as NVIDIA [21] points out.

With these considerations in mind, we seamlessly integrated the SegFormer B3 algorithm, as shown in Figure 4, into our semantic segmentation framework.
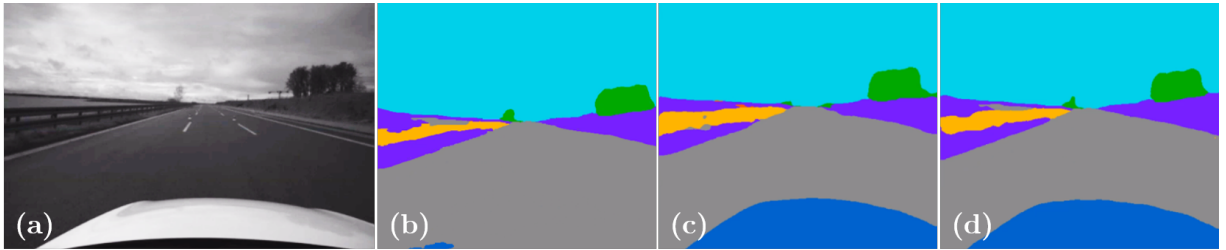
Fig. 4. Comparison between DeepLabV3+ (b), SegFormer B2 (c) and SegFormer B3 (d) on the original image (a). As shown in the figure, DeepLabV3+ is not able to detect the vehicle hood in this frame which would lead to false-positive feature points.

4.2 Feature Point Filtering

After assigning the individual pixels to their respective regions, the next phase involves processing feature points based on these region assignments. First, the region assigned to the pixel is linked to the corresponding feature point, establishing a link between the position of the feature point in the image and a particular pixel. Based on the previous assignment of the region to the pixel, this region is linked to the feature point via a corresponding index. A decision is then made as to whether the feature point is in a region that should be excluded from further use, or whether it falls in a region where feature points should be retained. At this point, additional information is assigned to each feature point, which is specified with the values -1, 0 or 1, depending on the region and the corresponding index.

$$FeaturePoint = \begin{cases} -1, & \text{for Region } Sky \\ -1, & \text{for Region } Hood \\ 1, & \text{for Region } Street \\ 0, & \text{for other Regions} \end{cases}$$

In particular, we assign specific indices to the feature points detected on the road, a crucial aspect for the subsequent phase of our comprehensive algorithm. This index information plays a critical role in determining whether feature points contribute to the creation of the global 3D point cloud map. In this process, we compare the previously
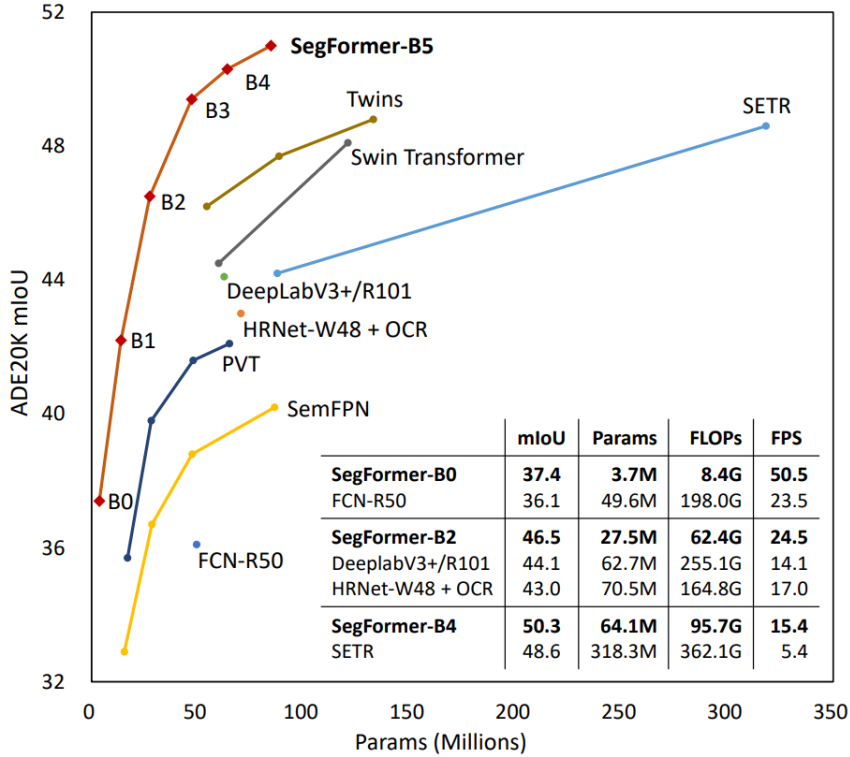
Fig. 5. Overview and comparison of the performance and processing speed of different neural networks for semantic image segmentation on the ADE20K dataset. The figure demonstrates the higher performance of SegFormer compared to DeepLabV3+. [20, p. 1]

assigned indices, keeping feature points with values of 0 and 1, while excluding those with a value of -1 from further consideration.

$$FeaturePointWithClasses = \begin{cases} Deleted, \text{ for Value } \textit{-1} \\ Keeped, \text{ for Value } \textit{0} \text{ and Value1} \end{cases}$$

As ORB-SLAM3 progresses, feature points that become map points are assigned additional labels such as "road" or "no road" depending on the values assigned. In the final phase of our current implementation, we need to evaluate the impact of semantic segmentation on the scaling of the global 3D point cloud map and the accuracy of the trajectory. Figure 6 gives a first qualitative insight into the results of our pipeline.

5   Evaluation

Environment mapping is done using different stereo cameras such as the *ZED2i* equipped with a polar filter and a baseline of 120 mm, and the *Intel RealSense D455* with a baseline of 95 mm. ORB-SLAM3 is applied to these cameras. The results presented in the images were obtained using the Intel Realsense D455. Detected features are recorded and stored in a three-dimensional point cloud. The impact of the low texture of the environment must be taken into consideration. To overcome the repetition of scene images, traffic cones are placed along the test track with gaps of 20 m, alternating in number on either side of the track. This ensures that feature matching and proper orientation in the point cloud occur. The camera is mounted at the rear-view mirror height on the top of the windshield, as is typical for cars. Figure 7 displays the test setup and track.
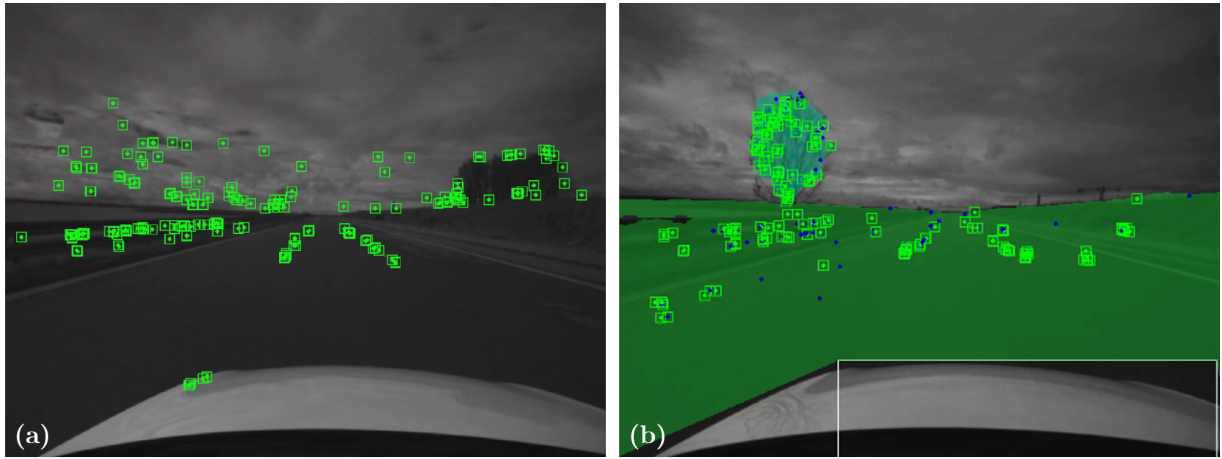
Fig. 6. ORB-SLAM3 detection results without filtering (a) and with filtered feature points (green rectangles) based on the semantic of the image (b). Using this method the false detection feature points in the sky and on the hood of the car are removed and being ignored by the mapping algorithm leading to better scaling and relocalization.
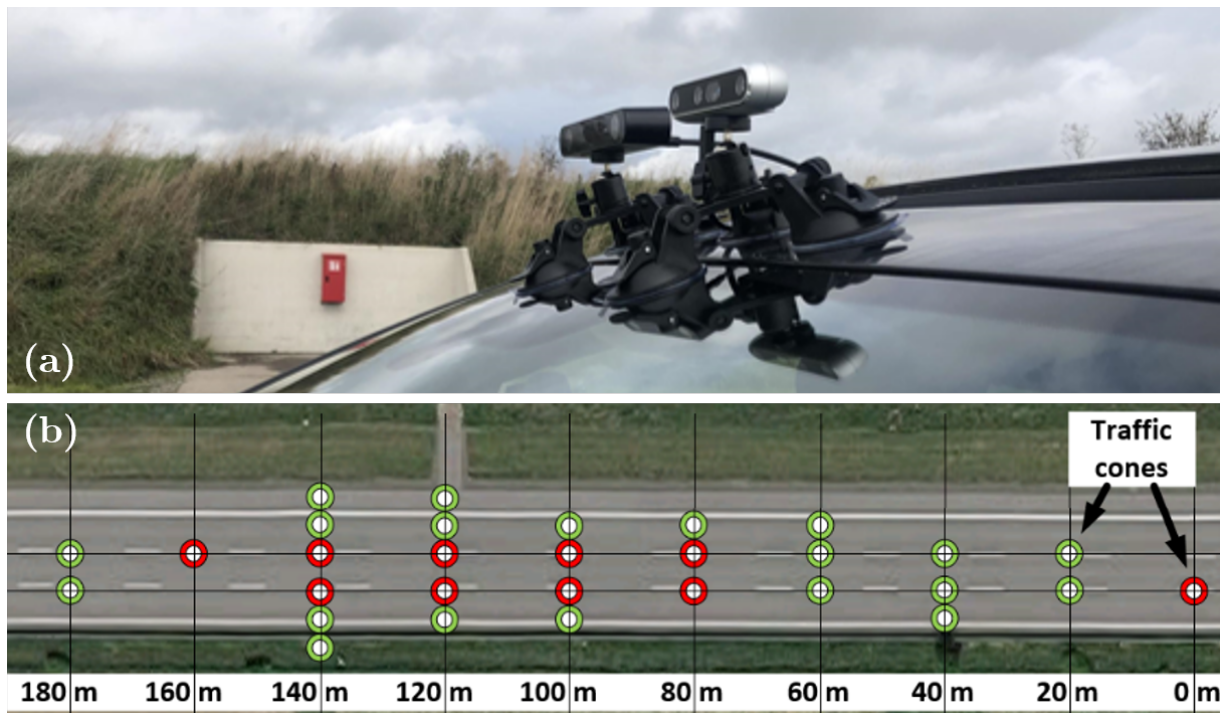
Fig. 7. Mounted stereo cameras on the test vehicle (a) and the test track with equipped traffic cones (b). The intervals are used in the evaluation to test the precision regarding scaling of the SLAM algorithm.

First of all, it will be examined whether semantic segmentation has a significant influence on the number of feature points. Table 1 shows, that the number of detected feature points decreased by more than 50 %. Assuming that the semantic image segmentation is correct, this means that more than 50 % of the detected feature points are located in the sky or on the hood of the vehicle and are therefore not suitable for further processing.

Table 1. Comparison of the number of detected map points with and without semantic segmentation.

Dataset	Number of Detected Map Points
Without Semantic Segmentation	8361
With Semantic Segmentation	3990
Total Deviations	-4371
Deviations in %	-52,3

To evaluate the impact on scaling, the distances between traffic cones are shown in table 2. The beneficial effects of semantic segmentation are easily seen. For an interval of 20 m, the average error is only -0.129 m compared to 1.414 m without segmentation. This leads to an relative reduction of -0.642 % compared to -7.073 % without segmentation. For the overall distance of the test track of 160 meters with semantic segmentation, the difference is -1.027 m (without segmentation 11.117 m). In summary, the use of semantic segmentation clearly adds significant value. Nevertheless, some additional considerations to the proposed approach are presented in the following section.

Table 2. Comparison of ground truth data (GT) with the original ORB-SLAM3 (No Segmentation) and with our approach (Segmentation). The intervals are given by the distances between the traffic cones as shown in Figure 7.

Interval	GT Length	No Segmentation	Segmentation
20 m - 40 m	20 m	18.603 m	19.840 m
40 m - 60 m	20 m	18.769 m	19.631 m
60 m - 80 m	20 m	18.974 m	19.921 m
80 m - 100 m	20 m	18.871 m	19.924 m
100 m - 120 m	20 m	18.871 m	19.993 m
120 m - 140 m	20 m	18.228 m	19.992 m
140 m - 160 m	20 m	18.365 m	19.678 m
160 m - 180 m	20 m	17.967 m	19.994 m
Average		18.586 m	19.871 m
Total Deviations		-1.414 m	-0.129 m
Deviations in %		-7.073	-0.642

6 Conclusion and Future Work

The proposed solution to increase the robustness and precision of vSLAM in automotive systems proved to be effective. Thus, the scaling error of the map could be significantly reduced by removing false-positive feature points. Furthermore, the information content of the map could be further increased by using semantic segmentation. The classes *Road* and *No Road* are now assigned to the corresponding feature points, which can be used in further calculations. Optimization opportunities arise with regard to processing speed, and image matching effects in homogeneous test scenarios as well as in more complex test procedures, which will be incorporated into Euro NCAP test scenarios in the future. Two proposals are being developed to improve the mapping of the environment:

- Change ORB-SLAM's feature detector and feature descriptor by a neuronal network to make the vSLAM more robust to a homogeneous environment as well as to rotations.
- Increase the processing speed of the entire vSLAM approach.

Acknowledgements

References

1. Pendleton, S.D., Andersen, H., Du, X., Shen, X., Meghjani, M., Eng, Y.H., Rus, D., Ang, M.H.: Perception, Planning, Control, and Coordination for Autonomous Vehicles. Machines **5**(1) (2017)
2. Weber, M., Weiß, T., Gechter, F., Kriesten, R.: Use of Artificial Intelligence and Image Segmentation for 3-Dimensional Modeling. In The Upper Rhine Artificial Intelligence Symposium UR-AI 2021, ed.: Artificial Intellgence – Application in Life Sciences and Beyond. (2021) 212–223
3. Weber, M., Weiß, T.: Object Detection by Using Image Segmentation for Augmented Reality in Advanced Driver Assistance Systems. In Institute of Energy Efficient Mobility, ed.: Reports on Energy Efficient Mobility – Volume 3. Volume 3. (2023) 14–28
4. Gao, X., Zhang, T.: Introduction to Visual SLAM: From Theory to Practice. 1st ed. 2021 edn. Springer eBook Collection. Springer Singapore and Imprint Springer, Singapore (2021)
5. Taketomi, T., Uchiyama, H., Ikeda, S.: Visual SLAM algorithms: a survey from 2010 to 2016. IPSJ Transactions on Computer Vision and Applications (2017)
6. Cheng, J., Zhang, L., Chen, Q., Hu, X., Cai, J.: A review of visual SLAM methods for autonomous driving vehicles. Engineering Applications of Artificial Intelligence **114** (2022) 104992
7. Campos, C., Elvira, R., G\´omez, J.J., Montiel, J.M.M., Tardós, J.D.: ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual-Inertial and Multi-Map SLAM. IEEE Transactions on Robotics **37**(6) (2021) 1874–1890
8. Mur-Artal, R., Montiel, J., Tardos, J.: ORB-SLAM: a versatile and accurate monocular SLAM system. IEEE Transactions on Robotics (2015) 1147–1163
9. Minaee, S., Boykov, Y., Porikli, F., Plaza, A., Kehtarnavaz, N., Terzopoulos, D.: Image Segmentation Using Deep Learning: A Survey. IEEE Transactions on Pattern Analysis and Machine Intelligence **44**(7) (2022) 3523–3542

10. Wang, L., Shi, J., Song, G., Shen, I.f.: Object Detection Combining Recognition and Segmentation. In Yagi, Y., Kang, S.B., Kweon, I.S., Zha, H., eds.: Computer Vision – ACCV 2007, Berlin, Heidelberg, Springer Berlin Heidelberg (2007) 189–199

11. Kuruvilla, J., Sukumaran, D., Sankar, A., Joy, S.P.: A review on image processing and image segmentation. In R., R., ed.: Proceedings of 2016 International Conference on Data Mining and Advanced Computing (SAPIENCE), Piscataway, NJ, IEEE (2016) 198–203

12. Kirillov, A., He, K., Girshick, R., Rother, C., Dollar, P.: Panoptic Segmentation. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Piscataway, NJ, IEEE (2019) 9396–9405

13. van Ratingen, M.R.: The Euro NCAP Safety Rating. In Piskun, A., ed.: Karosseriebautage Hamburg 2017, Wiesbaden, Springer Fachmedien Wiesbaden (2017) 11–20

14. Yu, C., Liu, Z., Liu, X.J., Xie, F., Yang, Y., Wei, Q., Fei, Q.: DS-SLAM: A Semantic Visual SLAM towards Dynamic Environments. In: Towards a robotic society, Piscataway, NJ, IEEE (2018) 1168–1174

15. Chen, L.C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H.: Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation. In Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y., eds.: Computer Vision – ECCV 2018, Cham, Springer International Publishing (2018) 833–851

16. Kamann, C., Rother, C.: Benchmarking the Robustness of Semantic Segmentation Models. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). (2020) 8825–8835

17. Chollet, F.: Xception: Deep Learning with Depthwise Separable Convolutions. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Los Alamitos, CA, USA, IEEE Computer Society (2017) 1800–1807

18. Zhou, B., Zhao, H., Puig, X., Fidler, S., Barriuso, A., Torralba, A.: Scene Parsing through ADE20K Dataset. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Los Alamitos, CA, USA, IEEE Computer Society (2017) 5122–5130

19. Zhou, B., Zhao, H., Puig, X., Xiao, T., Fidler, S., Barriuso, A., Torralba, A.: Semantic Understanding of Scenes Through the ADE20K Dataset. International Journal of Computer Vision **127**(3) (2019) 302–321

20. Xie, E., Wang, W., Yu, Z., Anandkumar, A., Alvarez, J.M., Luo, P.: SegFormer: Simple and Efficient Design for Semantic Segmentation with Transformers. In: Neural Information Processing Systems (NeurIPS). (2021)

21. NVIDIA Developer: Robust Perception with Vision Transformer SegFormer (2023)

Borderless Authentic –
Authentication in the Upper-Rhine Area with AI

Abdelhafid Abouaissa[1], Wilfrid Azan[2], Yannick Boehmann[3], Michael Böttger[4], Ugo Devoille[3], Marc Gilg[1], Martin Gwerder[5], Roxana Hess[6], Bastian Leferink[4], Pascal Lorenz[1], David Monschein[7], Zoltan Nochta[7], José Antonio Peregrina Pérez[7], Tim Piotrowski[7], Ioan Szilagyi[3], Oliver Waldhorst[7], Jochen Wendel[6], Oliver Wolf[4], Christian Zirpins[7,*]

[1] Université de Haute Alsace (UHA), 2 rue des Frères Lumière, 68093 Mulhouse Cedex, France
`[firstname.lastname]@uha.fr`
[2] Université Lumière Lyon 2: Lyon, Auvergne-Rhône-Alpes, France
`Wilfrid.Azan@univ-lyon2.fr`
[3] Neomia.ai, 3 Rue Pau-Henri Spaak, 68390 Sausheim, France
`[y.Boehmann|u.Devoille|i.Szilagyi]@neomia.ai`
[4] raumobil GmbH, Auer Straße 19, 76227 Karlsruhe
`[m.boettger|b.leferink|o.wolf]@raumobil.com`
[5] Fachhochschule Nordwestschweiz (FHNW), Bahnhofstrasse 5, 5210 Windisch, Switzerland
`[firstname.lastname]@fhnw.ch`
[6] INIT GmbH, Käppelestraße 4-10, 76131 Karlsruhe, Germany
`[Rhess|Jwendel]@initse.com`
[7] Karlsruhe University of Applied Sciences (HKA), Institute of Applied Research (IAF), Data-Centric Software Systems (DSS) Research Group, Moltkestr. 30, 76133 Karlsruhe, Germany
`[firstname.lastname]@h-ka.de`

Abstract. Efficient identity management is not only a concern of the virtual world but also paramount for modern open societies like the European Union. Non-intrusive, AI-based techniques of continuous authentication have recently been proposed to increase the security, efficiency and user friendliness of online systems and services. We introduce the research project aura.ai that will investigate how to transfer and apply these novel means of AI-based authentication in the public transportation area of the Upper-Rhine region.

Keywords: Continuous Authentication; Federated ML, Public Transport and Mobility

1 Introduction

Identity is a crucial feature of anything valuable. It is even more critical for human beings and their various roles, such as being a European citizen. Their identity is key to the property they own and the rights/obligations they have. Identity is intricately linked with the ability to prove it and the ability to validate such proofs, which is known as authentication. There are various means of authenticating humans, including physical means like ID, passport, driving license, or banking cards, and virtual means like passwords, certificates, credentials, and others. All of these hold the risk of getting lost or even being misused. Depending on the case, this might be trivial or disastrous. Such risks lead to additional security measures like multi-factor authentication, combining two or more means [1]. On the one hand, this safeguards authentication, but on the other hand, it increases complexity and effort [2], which opens room for various trade-offs.

In the digital domain (including the digitalized, cyber-physical world), progress has been made to seamlessly authenticate people in a cost-efficient way. This builds on early ideas to

---

* Corresponding Author

194

recognize behavioral patterns that are typical or outright unique to human individuals, like their keystrokes [3]. Today, artificial intelligence (AI) offers novel ways to realize recognition of behavioral patterns and apply it to strengthen other means of authentication [4] (for enhanced security) or even to substitute them (for convenience). In the following, we introduce one such technology, namely FML-CA, and its application for public transport in the Upper-Rhine region.

2 FML-CA: Federated ML for Continuous Authentication

Ongoing automated authentication based on contextual information is known as continuous authentication (CA) [5]. A novel way of realizing CA uses machine learning (ML) of behavior [6], such as spatial location, movements or swiping on a touch device. The network is another valuable source of behavioral data [7], where, for example, communication patterns, access location, low-level authentication features, and technical fingerprints may be collected. In any case, learning behavior requires training based on behavioral data of individuals or generalizable data of groups. The latter case benefits from public or shared data of people and organizations.

Handling behavioral data and patterns always raises privacy concerns. Regarding behavioral data, federated machine learning (FML) [8] allows the distributed training of partial patterns at the place where they originated and without the need for sharing them. In the end, the partial patterns are aggregated into a single AI model that represents the knowledge of all federation partners (people or organizations). Only the aggregated model is shared, and security technologies ensure the preservation of individual privacy [9], adequate contributions of all participants [10], and the absence of fraudulent manipulation.

The application of AI models for authentication is possible without revealing the models or their input data to the party that requests the authentication by means of homomorphic encryption (HE) [11]. Also, the real-time behavioral data of individuals being authenticated may be gathered by third parties without exposing their identities by means of pseudonymization. Altogether, FML-based CA (FML-CA) can be achieved while respecting privacy.

3 FML-CA for Public Transport in the Upper-Rhine Region

An interesting application area of FML-CA is public transportation, especially for such cases that include international operations crossing borders. To this end, we are focusing on the Upper-Rhine region that spans 350 kilometers along the river Rhine at the borders between France Germany and Switzerland. This tri-national region is involved in the European Territorial Cooperation program (also known as Interreg[†]). Therefore, it offers optimum conditions to implement innovative trans-national applications of FML-CA.

The research project aura.ai aims to tap the potential of FML-CA in the public transport area of the Upper-Rhine region. The goals of aura.ai are to make Upper-Rhine mobility more secure, less expensive, and more comfortable by using FML-CA. More concretely, the project will capitalize on experiences in the context of the regiomove project[‡]. regiomove is a mobility platform in the Karlsruhe area involving multiple federated providers of mobility services. Currently, a similar platform is being built for the Ortenau region and it is planned to connect the two platforms enabling interregional booking. Furthermore, the goal is to enable cross-border connectivity to Strasbourg. Therefore, regiomove is an excellent example of interregional and cross-

---

[†] https://interreg.eu/
[‡] https://www.regiomove.de/

border mobility. It will provide the application context for utilizing FML-CA, leading to technologies and lessons learned that can be also transferred to other regions in the Upper Rhine area, such as the Basel region.

Today, regiomove builds on close legal relationships of partners on the core platform to provide integrated IT-support for multimodal mobility (route planning, ticketing, bike/e-scooter/car sharing, etc.) in Karlsruhe. Currently, travelers are centrally authenticated to enable seamless use of all services, which is paramount for the required user experience. While reliable permanent login is already challenging in the centralized case, the integration of mobility services of other regions and beyond the core platform federation cannot be done centrally anymore. This hampers the extension of the mobility solution to cover the whole tri-national region and forces travelers to self-organize multiple accounts of regional providers, re-authenticate, and experience media breaks, thereby also reducing the acceptance and usage of public transport as such.

Aura.ai aims to demonstrate that FML-CA can help to solve authentication-related hurdles of interregional multimodal mobility solutions. As a Proof-of-Concept (PoC), we want to apply FML-CA to enable permanent login for services without traveler distraction by automated re-authentication and demonstrate this for the mobile app of the regiomove platform. Building on this core, aura.ai is going to study how to enable automated switches between regional mobility providers based on FML-CA instead of manual logins for services of the Karlsruhe, Ortenau and Strasbourg areas. Overall, this will open the perspective of a seamless network of mobility services across the tri-national region.

4 Aura.ai Technical Building Blocks

Concerning the methods and technologies to be applied in aura.ai, fundamental methods for FML-CA have been studied in the research project KIWI[§] at Karlsruhe University of Applied Sciences [12]. These approaches will be augmented, extended, and complemented in two directions, to make them applicable in the case of aura.ai.

First, network modeling and monitoring mechanisms in the domain of multi-provider/cross-border public transport, that are developed at the Université de Haute Alsace (UHA), enable the continuous gathering of real-time data, such as low-level login features, as a basis for CA.

Second, a proxy (PrivID) for a common authentication protocol, that is being developed at Fachhochschule Nordwestschweiz (FHNW), bridges the FML-CA approaches to already established technologies on the Web. PrivID extends the standard functionality of a basic authentication protocol (e.g., Oauth2/SAML2 [13]) to allow CA with privacy-preserving pseudonymization between various providers. This pseudonymization is necessary to connect authentication providers who want to share a joint account but are unwilling or unable (e.g., due to jurisdictional restrictions) to provide customer information. PrivID will introduce consumer-specific pseudonyms while the data functionality (e.g., a pseudonymized email address or a name in a front-end) remains maintained without breaking pseudonymity. Such features are exceptionally important in a transnational mobility context, as jurisdictional rules even may forbid sharing such data.

Together, aura.ai aligns these technical building blocks to augment FML-based CA with vital features for innovative cross-border public transport services.

[§]Artificial intelligence in secure web infrastructures with digital identity management (Künstliche Intelligenz in sicheren Web-Infrastrukturen mit digitalem Identitätsmanagement - KIWI) funded by the German Federal Ministry of Education and Research (BMBF, 16KIS1142K, https://www.h-ka.de/iaf/kiwi)

5 Conclusion

The upcoming research project aura.ai is targeting various innovation areas like digitalization, cybersecurity, and sustainable mobility. It will make existing scientific results, namely FML-based CA with pseudonymized network data, usable for the economic sector of public transportation. Measures of the project will include the development of demonstrators and the conduction of pilot applications in cross-border public transportation. The tri-national Upper-Rhine region will significantly benefit from the project as it will help to a) bring together transport service and IT providers from multiple countries, b) foster the free movement of people within the region, and c) make public transport even more attractive to citizens. Furthermore, aura.ai includes an approach to AI governance that will ensure the absence of bias and equal treatment of people regardless of gender, race, or disabilities. Overall, aura.ai will make a difference for the holistic benefit of the Upper-Rhine region in line with the goals of the European Union.

6 References

[1] A. Ometov, S. Bezzateev, N. Mäkitalo, S. Andreev, T. Mikkonen, and Y. Koucheryavy, "Multi-Factor Authentication: A Survey," *Cryptography*, vol. 2, no. 1, p. 1, Jan. 2018, doi: 10.3390/cryptography2010001.

[2] S. Das, B. Wang, Z. Tingle, and L. J. Camp, "Evaluating User Perception of Multi-Factor Authentication: A Systematic Review," 2019, doi: 10.48550/ARXIV.1908.05901.

[3] R. Joyce and G. Gupta, "Identity authentication based on keystroke latencies," *Commun. ACM*, vol. 33, no. 2, pp. 168–176, Feb. 1990, doi: 10.1145/75577.75582.

[4] D. Garabato, C. Dafonte, R. Santoveña, A. Silvelo, F. J. Nóvoa, and M. Manteiga, "AI-based user authentication reinforcement by continuous extraction of behavioral interaction features," *Neural Comput & Applic*, vol. 34, no. 14, pp. 11691–11705, Jul. 2022, doi: 10.1007/s00521-022-07061-3.

[5] F. H. Al-Naji and R. Zagrouba, "A survey on continuous authentication methods in Internet of Things environment," *Computer Communications*, vol. 163, pp. 109–133, Nov. 2020, doi: 10.1016/j.comcom.2020.09.006.

[6] Y. Liang, S. Samtani, B. Guo, and Z. Yu, "Behavioral Biometrics for Continuous Authentication in the Internet-of-Things Era: An Artificial Intelligence Perspective," *IEEE Internet Things J.*, vol. 7, no. 9, pp. 9128–9143, Sep. 2020, doi: 10.1109/JIOT.2020.3004077.

[7] D. Monschein and O. P. Waldhorst, "Privacy-Preserving and Scalable Authentication based on Network Connection Traces," in *Proc. GI/ITG Conf. on Networked Systems (NetSys)*, Virt. Conf., Sep. 2021. [Online]. Available: https://journal.ub.tu-berlin.de/eceasst/article/download/1175/1109

[8] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated Machine Learning: Concept and Applications." arXiv, Feb. 13, 2019. Accessed: Mar. 14, 2023. [Online]. Available: http://arxiv.org/abs/1902.04885

[9] T. Piotrowski and Z. Nochta, "Towards a Secure Peer-to-Peer Federated Learning Framework," in *Advances in Service-Oriented and Cloud Computing*, C. Zirpins, G. Ortiz, Z. Nochta, O. Waldhorst, J. Soldani, M. Villari, and D. Tamburri, Eds., Cham: Springer Nature Switzerland, 2023, pp. 19–31. [Online]. Available: http://dx.doi.org/10.1007/978-3-031-23298-5_2

[10] J. A. Peregrina, G. Ortiz, and C. Zirpins, "Towards Data Governance for Federated Machine Learning," in *Advances in Service-Oriented and Cloud Computing*, C. Zirpins, G. Ortiz, Z. Nochta, O. Waldhorst, J. Soldani, M. Villari, and D. Tamburri, Eds., in Comm. in Computer and Information Science, vol. 1617. Cham: Springer Nature Switzerland, 2023, pp. 59–71. doi: 10.1007/978-3-031-23298-5_5.

[11] D. Monschein and O. P. Waldhorst, "mPSAuth: Privacy-Preserving and Scalable Authentication for Mobile Web Applications." 2022. [Online]. Available: https://arxiv.org/abs/2210.04777

[12] D. Monschein, J. A. Peregrina, T. Piotrowski, Z. Nochta, O. P. Waldhorst, and C. Zirpins, "Towards a Peer-to-Peer Federated Machine Learning Environment for Continuous Authentication," in *Proc. 1st IEEE Int. Workshop on Distributed and Intelligent Systems (DistInSys), co-located with 26th IEEE Symp. on Computers and Communications (ISCC)*, Athens, Greece, Sep. 2021. [Online]. Available: http://dx.doi.org/10.1007/978-3-031-23298-5_2

[13] D. Hardt, "The OAuth 2.0 authorization framework," no. 6749. in Request for comments. RFC Editor, Oct. 2012. doi: 10.17487/RFC6749.

Exploring the Potential of Synthetic Data for Bike Path Surface Classification using Diffusion Models

Valentin Göttisheim[1], Holger Ziekow[1], Peter Schanbacher[1], Oliver Taminé[1], Jochen Baier[1]
Djafar Ould-Abdelsam[2]

[1] Furtwangen University,
`{goev, zie, spet, tao, bai}@hs-furtwangen.de`
[2] Université de Haute Alsace
`djafar.ould-abdeslam@uha.fr`

Abstract. Road surface classification of bike paths enables image recognition applications for bike route planning, navigation optimization or path maintenance. However, acquiring and annotating real data can be costly and time-consuming. Synthetic data can overcome data scarcity and annotation costs. We use synthetic data, generated by Stable Diffusion to improve neural network performance on new or unseen surfaces. We compare model performance for different real-synthetic data ratios. Our results show that synthetic data decreases the amount real data needed and improves neural network performance in road surface classification on new surfaces.

Keywords: synthetic data; diffusion models, image recognition, surface classification.

1 Introduction

We analyze road surface classification in the context of an image recognition application that supports bike route planning, navigation optimization, path maintenance, and traffic safety (Baier et al., 2023). A challenge for the system is the classification of unseen road or path surfaces, as acquiring and annotating large-scale, diverse, and accurately labeled real data can be costly and time-consuming. Synthetic data is artificially generated data that mimics the characteristics of the real-world and offers a potential solution to address the limitations of data availability and annotation efforts. Unseen path surfaces are surfaces that are not present in the training data but appear in the real world. For example, a surface classifier trained on asphalt, concrete, and gravel may encounter difficulties to classify images of bricks, snow, other gravel types or lightings conditions which diverge from the images present in the training data resulting in a poor performance of the classifier. A possible solution to this problem is to enhance new path surfaces with synthetic images which cover a wider range of surfaces variations to improve the generalization and robustness of the classifier (He et al., 2023; Lu et al., 2023).

2 Related Work

Bike path surface classification is a task of identifying the type or condition of road surfaces captured from images, videos or other sensors (e.g. Heidt and Dorer, 2021). Baier et al. (2023) proposed an approach for automatic analysis of bike paths. They used a convolutional neural network (CNN) to detect and classify road surfaces from camera images. Our work shares the experimental setup, however, they did not consider the use of synthetic data. Several methods have been proposed for synthetic image generation, such as generative adversarial networks (GANs) (Goodfellow et al., 2014) or latent diffusion models (LDMs) such as Stable Diffusion (Rombach et al., 2022). The general

potential of using synthetic data from diffusion models has been shown by (Azizi et al., 2023) ImageNet data. In our work we use Stable Diffusion in the use case of path surface classification. We are focusing on easing the addition of new surfaces to a classification system for path surfaces, by expanding small real-world data samples with synthetic data.

Figure 1. "Original and Synthetic Data": Examples of input-image and generated synthetic images.

Original	Synthetic	Original	Synthetic

3 Experiment Design and Results

We tested the hypothesis that synthetic data can enhance the training for new bike paths, in particular with a small number of real images. We compare the performance of two CNNs: one trained on real data and other ones trained on real data plus synthetic data. We evaluate both types on a new bike path dataset. We test different numbers of real images as basis for synthetic data, ranging from 0 to 10 per class, and analyze the performance gain through synthetic data. We collected a new bike path dataset similar to Baier et al., (2023). The dataset contained 4 different paths with 300 images per path, for a total of 1200 real images. For 10 real images of the new dataset, we generate 400 synthetic variations for each image using the common Stable Diffusion image-to-image 768-v-ema (Rombach et al., 2022) method. We use input prompts ("photo of a paved path, concrete, asphalt, Canon EOS R3") and a negative prompt ("painting, digital art") to match the variations of the real data samples. Figure 1 shows examples. Note that we do not evaluate how realistic the synthetic images are. Rather we focus on the improvement of synthetic images on model performance. We constructed 6 training sets (referred to as "real") with each class containing exactly 0, 2, 4, 6, 8 or 10 real new bike path images in addition to the original data set (20444 samples). Further, we created 6 additional data sets by adding the corresponding synthetic images for each real image (referred to as "enhanced"). For example, for 2 real world images per class we added all synthetic variations resulting in 22048 images (20444 original samples +2·2 real world images +1600 synthetic images) in total. As a common

Figure 2. "Performance Synthetic Enhanced vs Real Data": CNNs trained on synthetic enhanced (enha.) and on purely real data (real) with number of real images used (# real images). Reported f1-scores: (a) macro, (b) micro, (c) weighted. The comparison of performances shows the synthetic enhanced outperforms the CNNs trained on real data only.
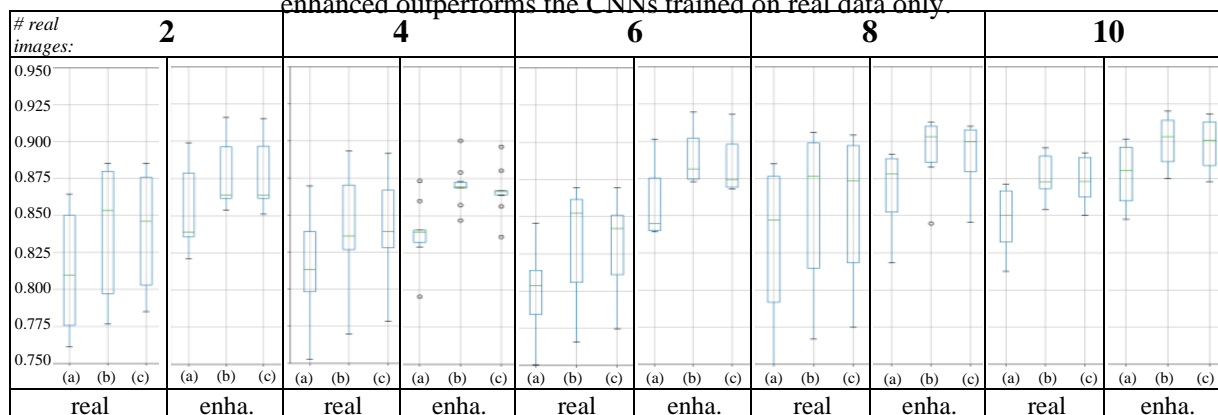
CNN architecture we use one augmentation layer, three convolutional layers with ReLu activation and max-pooling layer, a dropout and two fully connected layers. The augmentation involves random horizontal flipping, zooming and change in brightness. We trained the CNN with subsequent settings: 25 epochs, batch size 32, learning rate of 0.001, sparse cross-entropy loss, Adam-optimizer and accuracy as metric. For evaluation we used the model with the best validation loss. We trained the CNN on the original training dataset and evaluated it on the test set of the original dataset achieving an f1-score of 0.97. However, on the test set of the new bike path data, the f1 dropped to 0.83, showing that the CNN trained on the original dataset performs worse on the new bike paths. To test our hypothesis, we trained the CNN on the remaining final real world and synthetic enhanced datasets and compare the performance on the new bike path test set. We repeated the experiment 10 times. The results are visualized in Figure 2 which shows the boxplots of the weighted, macro and micro scores of f1 versus the amount of real data needed for different models on the bike path test set. Here, we only discuss the micro f1-score for simplicity. The classifiers trained on synthetic enhanced data outperform those trained on real data only (see Figure 2). The positive effect of synthetic data increases with the number of real images used. For example, with 2 real images the median f1 are 0.868 (real data) and 0.875 (enhanced data) compared to 10 images with 0.873 (real data) and 0.91 (enhanced data). We further verified the performance on the original test set during the experiments which showed only small variations of about an f1-score of 0.97 for all trained classifiers.

4 Discussion and Conclusion

We generate synthetic data for bike path surface classification using diffusion models. We compared two CNNs trained on real data and real data plus synthetic data and evaluated them on a new bike path dataset. We showed that enhancing real path images with synthetic data improves the classifier performance and therefore can enhance training for new bike paths surfaces. Our work has some limitations and challenges for future work. First, our synthetic data generation method relies on text prompts and model parameters, which may be subject to further optimization. E.g., the generated asphalted paths could be improved using other inputs. Second, our experiment was conducted on a small dataset of bike path images, which may limit the generalization and robustness. However, our work showed a promising approach, demonstrating the of use synthetic data to improve the CNN performance, especially on new surfaces.

References

1. Baier, J., Taminé, O., Deyringer, J., Mahn, N., Dittrich, I., Alili, B.: A Multi-Stage AI-based Approach for Automatic Analyzation of Bike Paths: Stage 1 – Road Surface Detection, TFC23: Towards AI-Aided Invention and Innovation, 2023
2. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y.: Generative adversarial nets. In Advances in neural information processing systems (pp. 2672-2680), 2014
3. He, R., Sun, S., Yu, X., Xue, C., Zhang, W., Torr, P., Bai, S., & Qi, X.: Is synthetic data from generative models ready for image recognition? arXiv preprint arXiv:2210.07574, 2023.
4. Lu, Y., Shen, M., Wang, H., & Wei, W.: Machine learning for synthetic data generation: A review. arXiv preprint arXiv:2302.04062, 2023.
5. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., & Ommer, B.: High-resolution image synthesis with latent diffusion models. In 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022.

6. Azizi, S., Kornblith, S., Saharia, C., Norouzi, M., & Fleet, D. J.: Synthetic data from diffusion models improves imagenet classification. arXiv preprint arXiv:2304.08466, 2023.
7. Heidt, Johannes, and Klaus Dorer. "Classification and Prediction of Bicycle-Road-Quality using IMU Data." In Artificial Intelligence: Application in Life Sciences and Beyond, pp. 138-149. 2021.

UR-AI Student School

URAI Autumn School dedicated to time series classification from robotics applications: a tutorial in a trinational context

Thomas Josso-Laurain[1], Maxime Devanne[1], Quentin Boerlen[1], Jonathan Ledy[1], Jean-Philippe Lauffenburger[1], Jonathan Weber[1]

[1] IRIMAS, University of Haute-Alsace, Mulhouse, France
Firstname.lastname@uha.fr

Abstract. URAI Symposium is a trinational research symposium organized by university partners inside the Upper Rhine region. This symposium aims to bring together researchers that are working on and/or with artificial intelligence. For the 5th edition in 2023, this symposium is taking place in Mulhouse, France. It is preceded by a one-day Autumn School. The participants are following a tutorial about how artificial intelligence can be used for time series from a robotic platform. The design of the tutorial is presented as well as the collected feedback. The results show that the tutorial is well tuned for AI beginners with an increase on the technical skills. All the participants were globally happy with the organization of the day.

Keywords: Artificial intelligence; Time series; Robotics; tutorial

1   Introduction

Artificial intelligence has become more and more popular in the research community, in computer science domain but also in every other applicative domain. Particularly, robotics have a growing interest in artificial intelligence algorithms since they are able to analyze the large number of information coming from the heterogeneous sensors.

Since 2019, an annual symposium is held in the Upper Rhine region, gathering researchers from universities from France, Germany and Switzerland. Organized by TriRhenaTech, this symposium aims to bring together scientists working in or with Artificial Intelligence. Every year, an application is targeted by TriRhenaTech and the organizing university to show the interest of AI for applicative research: industries (2020), life science (2021), medicine (2022) and robotics (2023).

This 2023 edition dedicated to AI for time series and robotics, is organized by the University of Haute-Alsace (UHA), France, mainly by the ENSISA engineering school and the IRIMAS institute of research. Taking place on the 17/11/23, an Autumn School is proposed on the 16th.

The Autumn School is a 1-day tutorial for Master and PhD students from all the TriRhenaTech partners. It has received a great interest with 12 registrations coming from people from the Upper Rhine region:

COUNTRY OF THE PARTICIPANTS

Hochschule Offenburg	2	Karlsruhe Univ. of Applied Sciences	1
French-German Research institute of Saint-Louis	2	Hochschule Furtwangen	1
ICAM Strasbourg	1	Univ. of Haute-Alsace	5
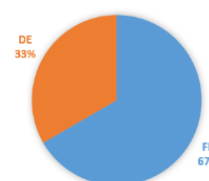
The program of the Autumn School is the following:

<p align="center"><u>Thursday, 16th of November 2023</u></p>

08:30 – 09:00: welcoming
09:00 – 10:00: AI basics
10:00 – 10:30: presentation to the SMART-UHA platform
10:30 – 12:00: machine learning classifiers
12:00 – 13:30: lunch
13:30 – 15:00: deep learning classifiers
15:00 – 15:30: measure acquisition with the SMART-UHA robot
15:30 – 16:30: classifiers validation

First, we will quickly present the robotics platform on which the Autumn School is based. Then, we will detail the work sessions with the objectives and examples of the code and results. Finally, the collected feedback from the participants is analyzed and presented.

2   SMART-UHA robot

The main application for this day is a robotic system part of the SMART-UHA project. A quick description of the project and the robot is provided to introduce the use-case of the tutorial day.

2.1   SMART-UHA project

The SMART-UHA project is a global project gathering all the research institutes and teaching facilities inside the University of Haute-Alsace (UHA). The IRIMAS institute is the project leader. It receives financial support from M2A, SGARE Grand Est, UHA and the FEDER founds.
This project is divided into two parts: one part dedicated to the electric energy management with solar panels and smart measurement units recording current, voltage, weather conditions and so on; a second part dedicated to the implementation of a robotic platform travelling through the campus in order to deliver letters and packages.

2.2   SMART-UHA robot

The robotic platform coming from the SMART-UHA project is equipped with a large number of sensors in order to ensure the safety of the university users. The environment perception is realized through stereovision cameras, LiDARs sensors and ultrasound sensors.
The localization of the robot remains on the GPS with RTK accuracy (a fixed antenna provides a triangulation within a centimeter precision). This ensures safety since the robot is able to track the desired trajectory with a very high precision.
In case of failure in the GPS cover (close to buildings for instance), the robot can navigate using odometry based on cameras (visual odometry) or the Inertial Measurement Unit (IMU). This IMU records a large variety of measures, including the accelerations and the angular velocities in the three axes of the robot (x pointing to forehead, y to lateral left and z to the sky).
The robot is communicating through Wi-Fi network within the UHA cover. An additional network, based on LoRAWAN technology, is providing a safety signal, sending periodically GPS coordinates on a huge range for small energy.

The robot has two control units: one "high-level" based on a NVIDIA Jetson GPU unit and another one "low-level" based on a EMTRION embedded computer. While the Jetson is processing most of the data coming from the LiDARs and the cameras, the EMTRION is controlling the four wheels with brakes and motors (propulsion and direction). Consequently, an overload of the GPU does not endanger the navigation of the robot which can stop at any moment in case of a failure.

2.3 Robotics use-case

The use-case we consider for this "AI for time series applied to robotics" tutorial is the automatic classification of the nature of the surface on which the SMART-UHA robot is navigating. The nature of the surface is a sensitive criterion when it comes to lateral stability since it highly modifies the adhesive coefficient of the tire-ground contact. This parameter is very hard to measure in real time.

In order to assess the nature of the surface, one can rely on the camera and use AI-based computer vision. However, the position of the camera and the technology make this sensor particularly sensitive to noisy environmental conditions (rain, fog, etc.) Another idea is to consider the IMU which is more trustable since it is stable and safely installed inside the robot body. The signals coming from the IMU, i.e. accelerations, are then considered as time series.

Therefore, the use-case for the tutorial day is to classify the nature of the surface between two classes: "concrete" and "grass". First approaches will consider univariable algorithms based on the x-acceleration. More approaches are then proposed along the day, considering multivariable algorithms with acceleration on x and z axes.
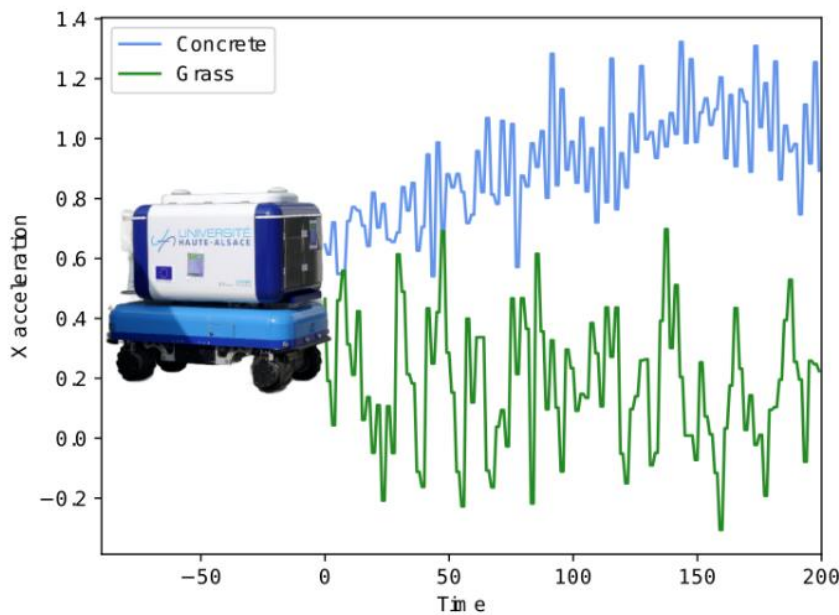
Fig. 1. Use-case: x-acceleration analysis in order to classify the nature of the ground (grass/concrete)

3 Tutorial details

The tutorial is based on Google colab codes completed with full comments, texts and videos to illustrate particular elements such as data normalization of k-nn algorithm.

The tutorial is divided into 5 sessions, 3 in the morning and 2 in the afternoon. For each session, the technical objectives are presented as well as some examples of the code and the results.

3.1   Basics in AI

The first session is dedicated to basics in AI since the level of knowledge of the participants is unknown.

Objectives:
- Using data normalization (z-norm)
- Importing Python libraries
- Visualizing time series
- Computing Euclidean distance between time series (non-normalized and normalized)
- Splitting a dataset
- Understanding Cross-validation

First, a synthetic dataset is considered with two classes: rectangles and triangles. Below is a display of the raw dataset proposed to the participants:
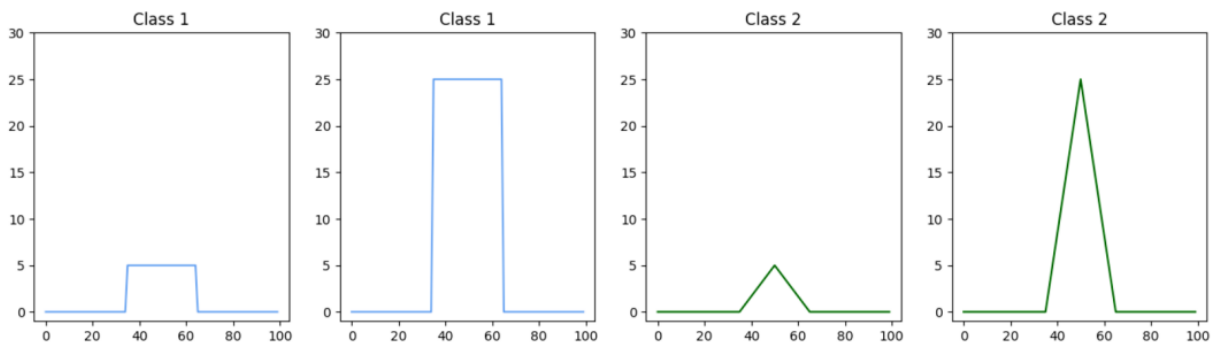
Fig. 2. Synthetic dataset

Computing the Euclidean distance, we can see that the first time series are closer to the small triangle of Class 2. A z-normalization is then applied on the dataset. The Euclidean distance is computed to show that the first time series belongs to Class 1.
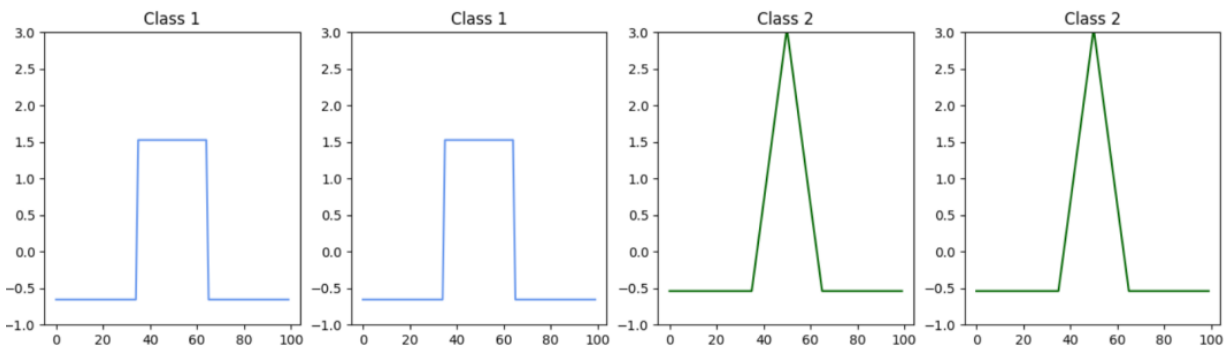
Fig. 3. Normalized synthetic dataset

The second part of this first session of the morning is dedicated to the study of the SMART-UHA dataset including x-acceleration raw data from the robot, with a split between train data and validation data.
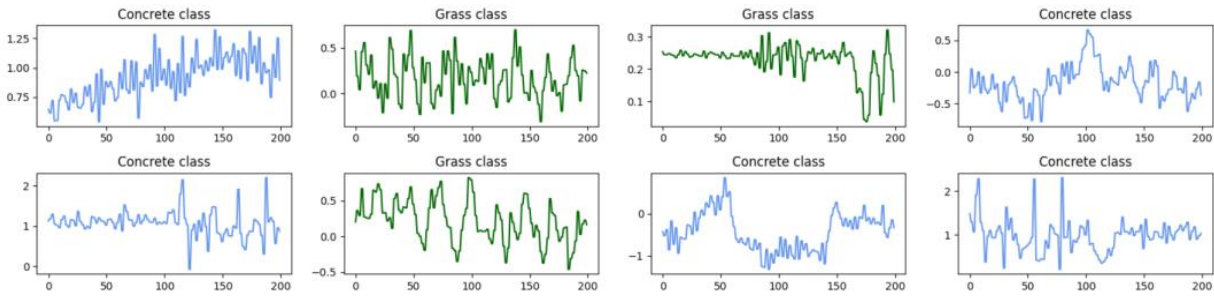
Fig. 4. Examples of SMART-UHA dataset (raw data + label)

3.2   Machine Learning classifiers – part 1

The next session of the morning is dedicated to Machine Learning classifiers.
Objectives:

- Applying k-NN (Nearest Neighbor) algorithms with Euclidean distance
- Reading a Confusion Matrix
- Computing the Dynamic Time Warping (DTW) distance

The Colab tutorial is composed on commented code but also schematic videos uploaded on Youtube to get the main idea of a concept. Below is an example with the 1-NN algorithm:

Fig. 5. Example of the 1-NN video on Youtube (@Maxime Devanne)

The participants have the choice to either compute the commented code that is already written, or to write again the code. In some parts, they have to write from scratch: for instance, the 1-NN algorithm is already written, but not the 3-NN they have to test further.

```
[7]  # counter of good classifications initialized at 0
     good_classification = 0
     # a for loop over all predictions
     for i in range(y_pred.shape[0]):
       # For each prediction, if the predicted label corresponds to the true one, we increment the counter by 1
       if y_pred[i] == y_test[i]:
         good_classification = good_classification + 1

     # The rate of good classifications (accuracy) corresponds to the number of good classification divided by the total number of predictions
     accuracy = good_classification / y_pred.shape[0]
     # printing the classification accuracy in percentage
     print('Percentage of good classifications:', accuracy*100)

     Percentage of good classifications: 50.0
```

Fig. 6. Example of the code provided on Google Colab – here the 1-NN prediction result

At the end, they reach a prediction based on the 3-NN algorithm with Dynamic Time Warping distance. The prediction can be assessed using a Confusion Matrix stating the probability for each class to be detected as it is.
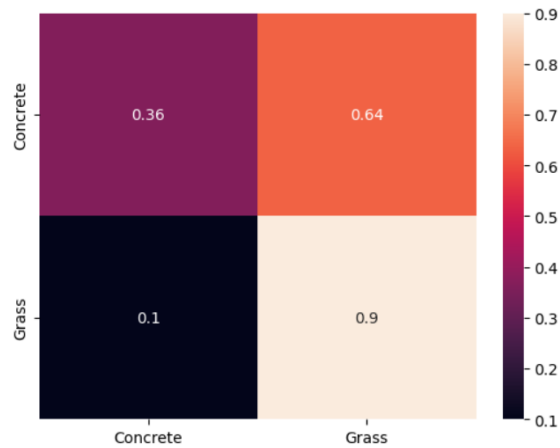
Fig. 7. Final Confusion Matrix obtained for a 3-NN algorithm with DTW distance

3.3 Machine Learning classifiers – part 2

After investigating some simple algorithms based on k-NN, the participants are considering linear classifiers.

Objectives:

- Using Softmax function to get output probabilities
- Computing a loss function based on binary cross entropy
- Optimizing the training of the model by gradient descent
- Applying linear classifiers to make predictions
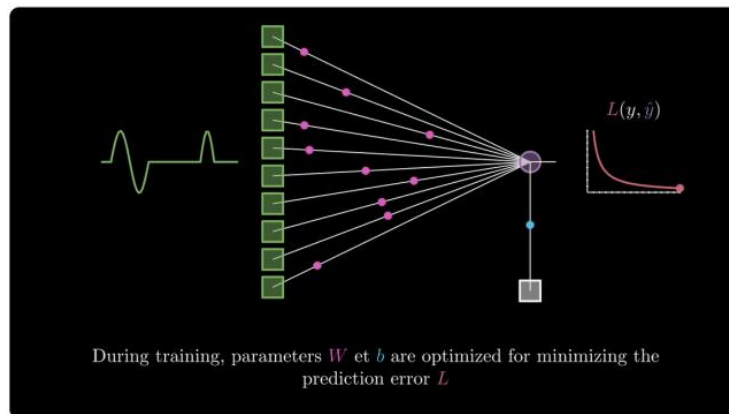
Fig. 8. Example of the video about linear classifier training (Maxime Devanne – Youtube)

```
[6]  # initializing an array for storing Softmax values
     softmax_outputs_serie0 = np.zeros(outputs_serie0.shape)
     # a for loop over the scores
     for i in range(outputs_serie0.shape[0]):
       # applying the Softmax function
       softmax_outputs_serie0[i] = np.exp(outputs_serie0[i]) / np.sum(np.exp(outputs_serie0))

     # printing the Softmax output (probabilities)
     print(softmax_outputs_serie0)

[0.13192183 0.86807817]
```

Fig. 9. Example of the commented code of the Softmax function

208

The goal of the morning is to achieve classification on the SMART-UHA dataset using the Machine Learning technique of linear classifiers. The weights of the linear classifier are computed by optimizing a loss function based on the binary cross entropy and the gradient descent. The cross entropy presented in a text element of the Google Colab such as:

In the case of a classification, rather than a simple difference, we use cross entropy which is more suitable. In the case of binary classification (as in our case), the binary cross entropy is defined as follows:

$$L = -y * log(\hat{y}) + (1 - y) * log(1 - \hat{y}),$$

with y being the true label and $\hat{y}$ is the predicted label.

Fig. 10. Cross entropy explanation in the Google Colab document

The participants learn then the importance of the weights initialization. A 10-fold prediction provides an average performance of the linear classifier on the SMART-UHA dataset:

```
Accuracy of 10 runs: [0.21875 0.34375 0.25    0.3125  0.25    0.34375 0.34375 0.28125 0.21875
 0.3125 ]
Average accuracy: 0.2875+-0.0480071609241788
```

Fig. 11. Linear classifier accuracy on classifying concrete and grass on the SMART-UHA dataset

It is not very performant! The use-case that has been designed makes simple machine learning algorithms fail so there is a need to investigate deep learning classifiers.

3.4   Deep Learning classifiers

The afternoon is dedicated to deep learning classifiers, particularly to the family of Convolutional Neural Networks (CNN)

Objectives:

- Understanding the process of a convolution
- Building a CNN architecture with layers of convolution, ReLU and Global Average Pooling 1D using the Keras framework
- Training a CNN model using Adam optimizer
- Reading a Class Activation Map (CAM)
- Improving the performance with multivariable CNN

```
Model: "sequential"
_____
 Layer (type)                Output Shape              Param #
=================================================================
 conv1d (Conv1D)             (None, 200, 32)           192

 global_average_pooling1d (  (None, 32)                0
 GlobalAveragePooling1D)

 dense (Dense)               (None, 2)                 66

=================================================================
Total params: 258 (1.01 KB)
Trainable params: 258 (1.01 KB)
Non-trainable params: 0 (0.00 Byte)
_____
```

Fig. 12. Simple CNN model built by the participants

```
[12] keras.utils.plot_model(simple_cnn_model, show_shapes=True)
```

conv1d_input	input:	[(None, 200, 1)]
InputLayer	output:	[(None, 200, 1)]

conv1d	input:	(None, 200, 1)
Conv1D	output:	(None, 200, 32)

global_average_pooling1d	input:	(None, 200, 32)
GlobalAveragePooling1D	output:	(None, 32)

dense	input:	(None, 32)
Dense	output:	(None, 2)
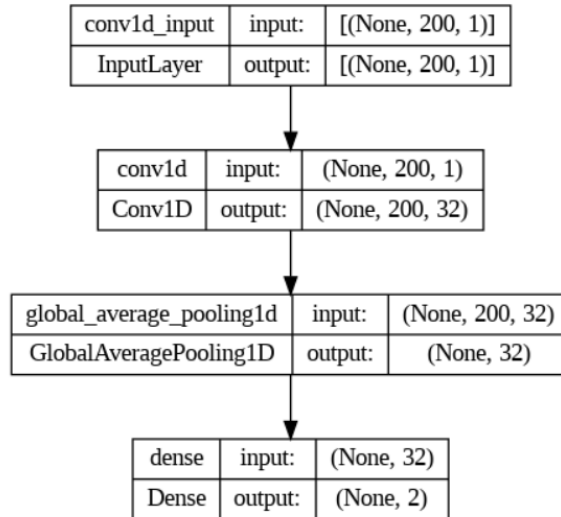
Fig. 13. Display of the simple CNN model using Keras

Once the CNN model has been built, the participants are training it with the SMART-UHA dataset:

```
4/4 [==============================] - 0s 7ms/step - loss: 0.4922 - accuracy: 0.7619
Epoch 997/1000
4/4 [==============================] - 0s 6ms/step - loss: 0.4922 - accuracy: 0.7619
Epoch 998/1000
4/4 [==============================] - 0s 6ms/step - loss: 0.4916 - accuracy: 0.7778
Epoch 999/1000
4/4 [==============================] - 0s 5ms/step - loss: 0.4916 - accuracy: 0.7778
Epoch 1000/1000
4/4 [==============================] - 0s 5ms/step - loss: 0.4914 - accuracy: 0.7619
Training time (s): 29.6695294380188
```

Fig. 14. Training the simple CNN model over 1000 epochs
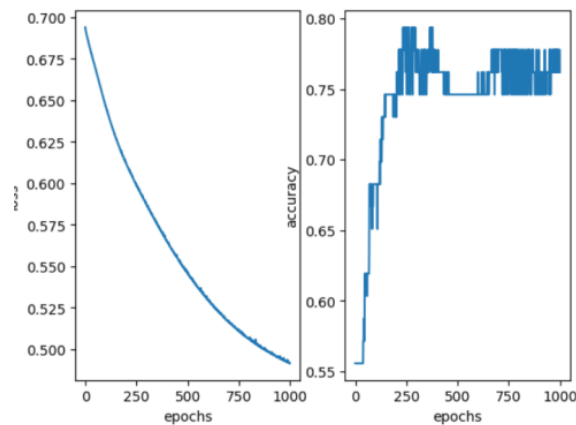
Fig. 15. Loss and accuracy through the training process

The trained CNN model is then used to predict over the test part of the SMART-UHA dataset. The results are (hopefully) better than machine learning algorithms:

```
Accuracy on the test set: 78.125
```

Fig. 16. Result of the 1-Layer CNN prediction

The participants are then asked to build more complex CNNs with more layers. In addition, the hot topic of explainability is introduced. Right now, explaining the results of AI algorithms has received a growing interest from the research community.

First, the activated features are studied layer-by-layer:
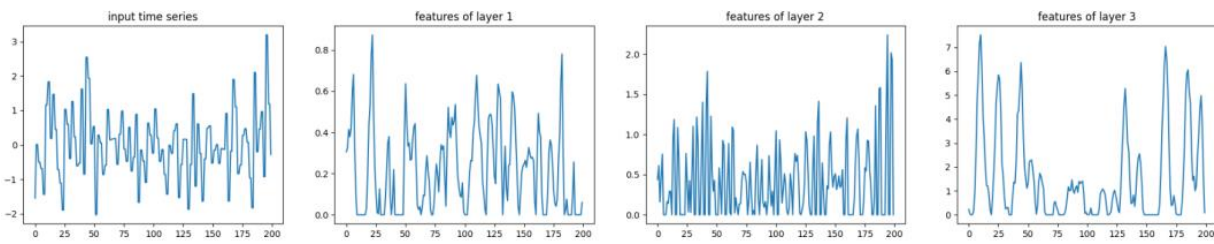
Fig. 17. Features of each layer of a 3-layers CNN

And then by the Class Activation Map:
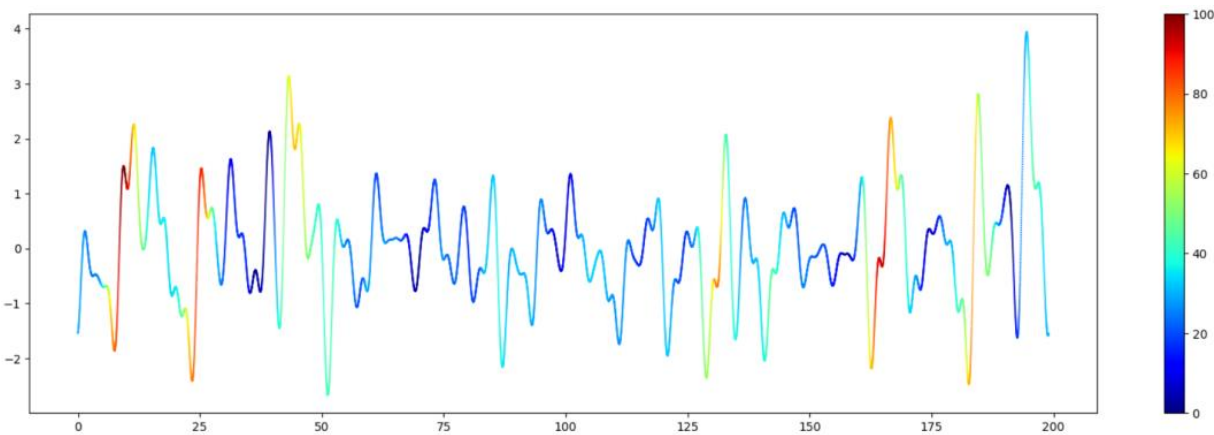
Fig. 18. Class Activation Map of the 3-layers CNN

where the red parts corresponding to the most relevant features for the prediction.

The next step for the participants is to improve the performance of the prediction by investigating multivariable neural networks. To reach that, not only the acceleration on the x-axis is considered but also on the z-one (pointing the sky). After the training, we can see that the multivariable CNN model performs better than the univariable one:

Fig. 19. Accuracy and loss of the multivariable (x,z) 3-layer CNN model

3.5   Validation on a real-field complex dataset

The final stage of the afternoon is to evaluate the performance of the previously designed univariable neural networks over a complex dataset recorded during a real-field use; therefore, it alternates between the grass sections and the concrete sections, changing at different and variable times.

Objectives:

- Training and evaluating a CNN over long complex time series

The considered model is a univariable (x-acceleration) 3-layers CNN. It is trained over 1000 epochs with a learning rate of 0.001.

Below are the ground truth (first plot) and the prediction classes (second plot) where green is grass and blue is concrete:

Accuracy (%): 79.10000000000001
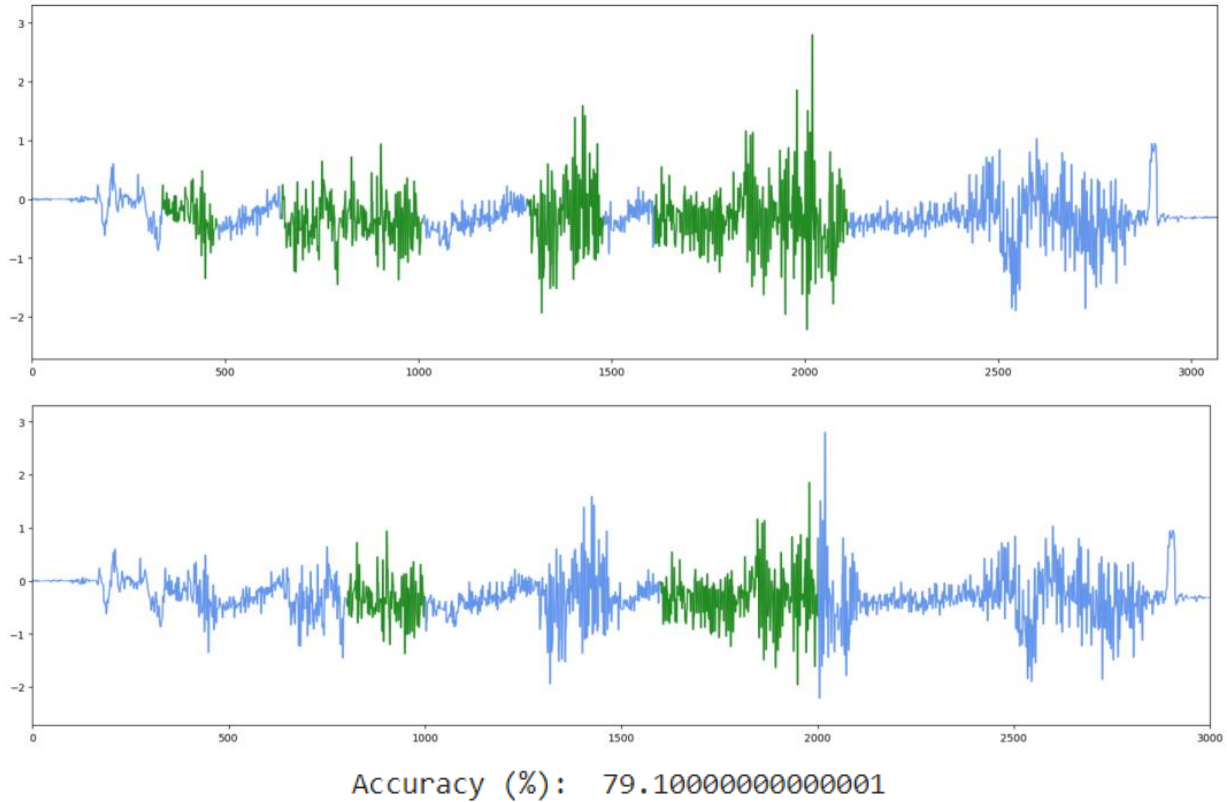
Fig. 20. SMART-UHA real-case validation sequence (up) and 3-layers CNN prediction results (bottom)

4 Participants feedback

4.1 Feedback evaluation protocol

As presented in the previous section, the tutorial content has been designed to address the defined technical objectives. Moreover, it can be interesting to also evaluate the feeling of the participants regarding the organization of the day, from a global point of view (lunch, coffee breaks…) to a scientific content point of view (speed of the tutorial, etc.).

At the end of the tutorial day, a form has been sent to the participants to evaluate if the objectives have been reached. The form is anonymous to ensure liberty of feedback. The survey is composed on the following questions:

Questions	Assessed index	Type
1 Did you enjoy?	Feeling	1-5 scale
2 How was the density of the knowledge?	Feeling/Technical	1-5 scale
3 How was the speed of the tutorial?	Feeling/Technical	1-5 scale
4-1 Level of skill in AI before the tutorial	Technical	1-10 scale
4-2 Level of skill in AI after the tutorial	Technical	1-10 scale
5 Reusability of what you have learnt	Technical	1-3 scale
6 What was the best moment of the day?	Feeling	Multiple Choice
7 Any free comments?	Feeling/Technical	Free comments

The assessed items cover an equal part of feeling and technical as we can see:

FEELING/TECHNICAL EVALUATION ITEM REPARTITION

feeling 45%

technical 55%
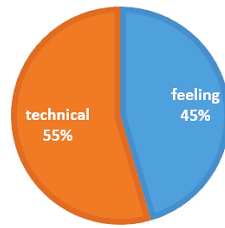
4.2 Feedback analysis

Over the 12 participants, 6 accepted to answer to the feedback form.
First of all, the participants were globally happy and enjoyed very much the tutorial day (Q1).
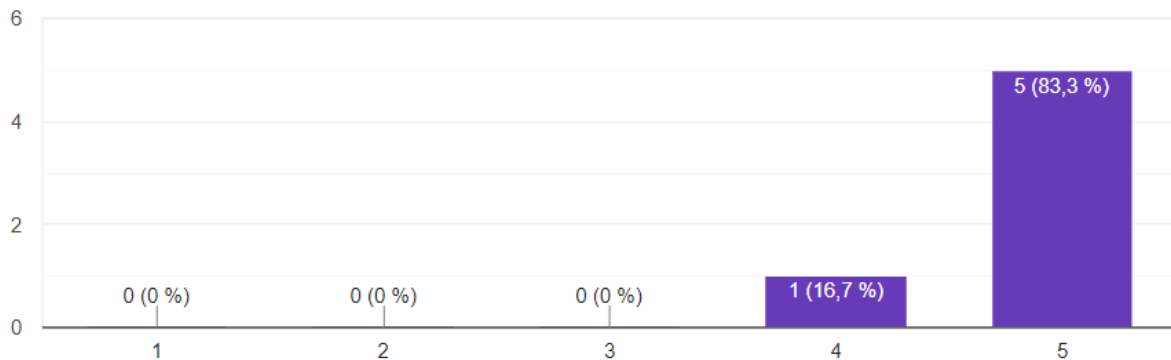
Fig. 21. Happiness assessment results with subjective scale (1 : not happy)

Regarding the tutorial organization, we assessed both the density of the knowledge presented in the tutorial supports, and the speed of the training for the whole day. For the density, we designed a Gaussian-like scale, with 1 = no dense at all, and 5 = way too dense. This means that 3 is a good score. The same scale is used for the speed, with 1 = too slow and 5 = too fast. The results are exactly the same for the two assessments (Q2 and Q3)
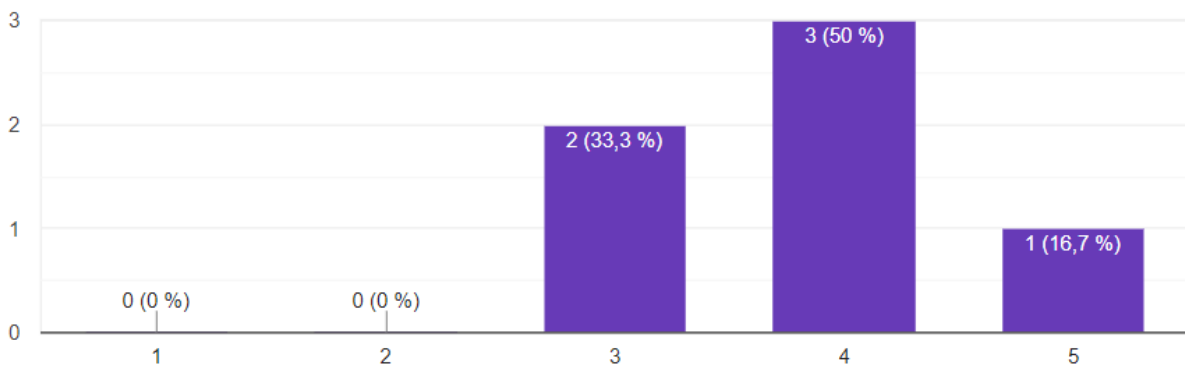
Fig. 22. Density and speed assessment results with Gaussian-like scale (1 = no dense ; 5 = too dense)

The next two questions Q4-1 and Q4-2 allow us to evaluate the technical skills acquired during the day since we asked the participants to assess their mastery of AI before and after the day.

Evolution of the technical skills of the participants
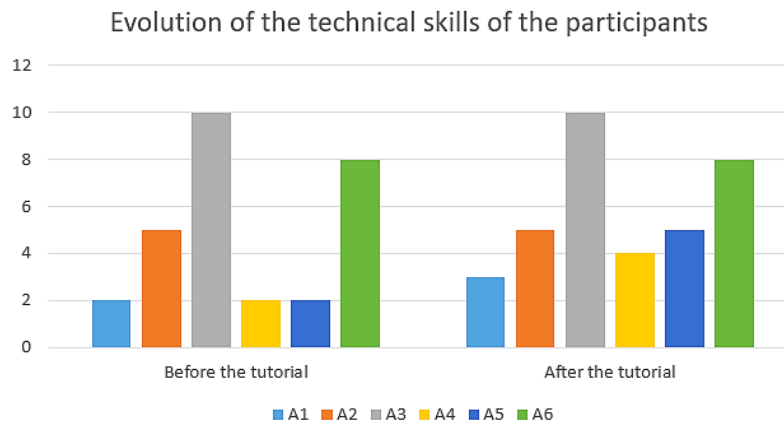
Fig. 23. Density and speed assessment results with Gaussian-like scale (1 = no dense ; 5 = too dense)

The results show different interesting things: first of all, the heterogeneous caracteristic of the audience. We can distinguish three categories of participants: those who do not know AI (starting around 2/10), those who have knowledge about it (around 5/10) and those who already master AI (over 8/10)

For the AI masters, there is no difference between and after since they are daily working with AI. They did not criticize the rhythm of the day, so we can conclude that they were not the targeted audience for a tutorial.

For the AI knowers, we notice no difference between before and after, meaning that they did not improve their technical skills. However, they still enjoyed the day and none of them complained about being bored or the tutorial being too slow. We can conclude that for this category, they enjoyed the tutorial by consolidating their knowledge and discovering the robotics application.

For the AI beginners, the difference is very important, with an average gain of technical skills of +2/10. That means that the tutorial is well designed since it starts with AI basics and goes up the to deep learning. For AI beginners, it is a very good introduction to the world of classification.

Another point we wanted to assess is the reusability of the acquired technical skills in the participants research topics.
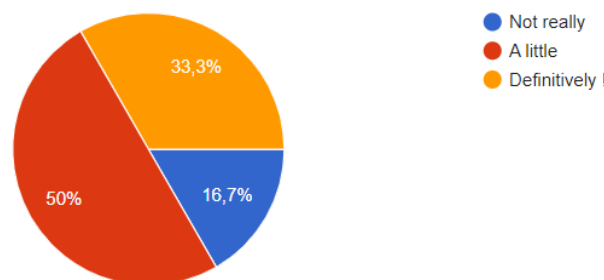
Fig. 24. Reusability results

We can see on the figure that the audience was composed of persons in need for AI as well as curious people. The limits of this one-question result is that is to not possible to determine if the Definitively answers are because the participants already work with AI or if they clearly learnt how to use it within their own research.

The last question results illustrate that the participants liked learning autonomously with the tutorial support as well as discovering the robotics platform of the lab.

The free feedback section contains elements that highlight the interest of the participants in such kind of tutorials, particularly in the Upper Rhine region. This is a good hint for the future!

"Thank you for today's tutorial. It is very good for AI beginners. I hope there will be similar workshops or schools at the next URAI Symposium where people can share knowledge about their research areas."

5 Conclusion

In conclusion, the URAI Autumn School was designed to propose a tutorial based on artificial intelligence for time series applied to robotics. We received 12 participants coming from France and Germany.
The tutorial covered a large panel of AI tools, starting from the basics as data normalization until Convolutional Neural Networks. The algorithms were applied on synthetic data as well as real-field data coming from the SMART-UHA robotic platform (raw data of x-acceleration from the Inertial Measurement Unit)
The feedback of the participants was collected with a form. The results analysis show that the tutorial is well tuned for AI beginners. Participants that already quite master AI are not feeling bored over the day since they are able to apply the tools, they know for time series in robotics. The rhythm allows the participants to work autonomously, alternating with coffee breaks and platform presentation.
For the future, the level of the participants should be assessed during the registration in order to create different groups in function of their mastery level. The tutorial experience should be definitively reconduct in the next URAI Symposiums as highlighted by the participants.

6 Acknowledgment

Author Index

217