Evaluation of Transformer Architectures for Electrical Load Time-Series Forecasting

Matthias Hertel*, Simon Ott*, Benjamin Schäfer, Ralf Mikut, Veit Hagenmeyer, Oliver Neumann

Institute for Automation and Applied Informatics,
Karlsruhe Institute of Technology
Hermann-von-Helmholtz-Platz 1, 76344 Eggenstein-Leopoldshafen
E-mail: matthias.hertel@kit.edu, simon.ott@student.kit.edu
* equal contribution

Abstract

Accurate forecasts of the electrical load are needed to stabilize the electrical grid and maximize the use of renewable energies. Many good forecasting methods exist, including neural networks, and we compare them to the recently developed Transformers, which are the state-of-the-art machine learning technique for many sequence-related tasks. We apply different types of Transformers, namely the Time-Series Transformer, the Convolutional Self-Attention Transformer and the Informer, to electrical load data from Baden-Württemberg. Our results show that the Transformes give up to 11% better forecasts than multi-layer perceptrons for long prediction horizons. Furthermore, we analyze the Transformers' attention scores to get insights into the model.

1 Introduction

Transmission system operators (TSOs) must balance the electricity supply and electrical load in the grid at every moment [1]. Otherwise, the grid becomes instable, which can lead to electricity outages. In order to plan the dispatch of energy storages and remaining fossil power plants, as well as the control of flexible consumers, accurate forecasts of the electrical load for the next hours to days are needed. Renewable power plants can be curtailed more easily than fossil power plants, which need more time to reduce their generation. Therefore, when the electrical load is overestimated, usually the renewable energy sources get curtailed. This means good forecasts of the electrical load are also necessary to maximize the usage of renewable energy.

Recent work on time-series forecasting showed good results with Transformers [2] for different applications [3, 4, 5, 6], including electrical load forecasting [7, 8, 9, 10, 11, 12]. Transformers can process long sequences and model long-term dependencies with the attention mechanism [2]. Therefore, they have the potential to give good results in electrical load forecasting and work especially well for long prediction horizons. In this work, we analyze whether the Transformer beats multiple baselines in forecasting the electrical load for the German state Baden-Württemberg, and discuss possible future usages of Transformers in energy forecasting.

Our contributions are the following:

- We compare multiple types of Transformers, namely the Time-Series Transformer [3], the Convolutional Self-Attention Transformer [5] and the Informer [6] on forecasting the electrical load of the state Baden-Württemberg.
- We compare the Transformers with multiple baselines, including a load profile baseline, linear regression models and multi-layer perceptrons.
- We analyze the attention scores of one of the Transformer models to get insights into the model's predictions, and we propose Transformer architectures that we want to test in the future based on our experience.
- We make all the code for our experiments publicly available.[1]

[1] github.com/KIT-IAI/Transformer-Networks-for-Electrical-Load-Time-Series-Forecasting

The paper is organized as follows: First, we discuss the related work on electrical load forecasting and Transformers in Section 2. Then, we define the electrical load forecasting task in Section 3. The different Transformer architectures are introduced in Section 4. The experimental setup, results and analysis of the attention scores are described in Section 5. Finally, we conclude in Section 6 with an outlook to future work on the topic.

2 Related Work

Modeling the patterns that underlie social behavior as energy load is difficult, which is why data-driven solutions are used in practice. Classical approaches rely on statistical methods with manually engineered features, such as linear regression, ARIMA and Support Vector Machines. To overcome the manual feature engineering, new methods based on deep learning were developed. González Ordiano et al. [13] give an overview on existing energy time-series forecasting methods, including linear regression and multi-layer perceptrons, which we are going to use as baseline models (see Section 5). More sophisticated methods were developed in the meantime, such as profile neural networks [14].

Transformers [2] were originally developed in the field of Natural Language Processing, where they became the state of the art in many tasks. Transformers use attention to retrieve information from the input time series and are thereby capable of modeling long-term dependencies. Multiple publications adapt the Transformer architecture to overcome specific disadvantages. The Convolutional Self-Attention Transformer [5] combines the attention mechanism with convolutions, to be able to better recognize patterns in the time series. The Informer [6] introduces ProbSparse attention to reduce the time and space complexity of the attention mechanism, and adds convolutions and max-pooling layers which reduce the length of the time series after each encoder layer. Zeng et al. [15] on the other hand question whether Transformers are really effective for time series forecasting. Our goal is to apply the different proposed Transformer architectures to state-level aggregated electrical load data from Baden-Württemberg and compare them against strong baselines.

3 Task Definition

We address the following electrical load forecasting problem: At a time step t, given the hourly electrical load of the previous p time steps $x_{(t-p+1):t} = (x_{t-p+1},...,x_t)$, m covariate sequences $z^j_{(t-p+1):t}$ with $1 \leq j \leq m$, and n a priori known covariate sequences $z^l_{(t+1):(t+\tau)}$ with $1 \leq l \leq n$, the goal is to predict the next τ electrical load values $x_{(t+1):(t+\tau)}$. We use one week's values as input (i.e. $p = 168$), and a forecasting horizon of $\tau = 96$ hours. We use time and calendar features as covariates, as explained in detail in Section 5. In the future, the covariates can be extended to cover external data such as weather data.

4 Approach

We use three different Transformer architectures. First, the Time-Series Transformer; second, the Convolutional Self-Attention Transformer; and third, the Informer. The architecture of the Time-Series Transformer is described in Section 4.1. It is the base of the other two Transformer architectures. The differences between the Convolutional Self-Attention Transformer and Informer and the Time-Series Transformer are described in Sections 4.2 and 4.3 respectively. The hyperparameters of the Transformer models are given in Section 5.3.

4.1 Time-Series Transformer

An overview of the Time-Series Transformer architecture is shown in Figure 1. The model consists of an encoder (shown in the left-hand part of the figure) and a decoder (shown in the right-hand part of the figure), both described in the following.

The input to the encoder is a sequence of p vectors, one for each past time step used by the model. Each vector contains one entry for the electrical load and additional entries for the time and calendar features for this time step. Before giving the vectors as an input to the encoder, we run them through a linear layer with d_{model} units, so that the input to the first encoder layer has shape $p \times d_{\text{model}}$, where d_{model} is the hidden dimension of the Transformer. Each encoder layer attends to the p outputs of the previous layer with the multi-head self-attention mechanism.

The input to the decoder consists of the vectors for the previous p_d time steps and the next τ time steps. The electrical load for the next τ time steps is unknown and therefore set to zero. The vectors are also run through a linear layer to increase the vector size to d_{model}. Each decoder layer attends to the outputs of the previous layer with the multi-head self-attention mechanism. Masking prevents the self-attention from attending to vectors that correspond to future time steps.[2] In addition, each decoder layer attends to the outputs of the last encoder layer with the multi-head cross-attention mechanism. The last τ outputs of the decoder, which correspond to the τ next time steps, are fed into a linear layer with a single unit, resulting in the τ predictions.

A sinusoidal positional encoding is added to the input of the first encoder and decoder layer. This is used in the Transformer [2] to make use of distances and absolute positions in the time series. Since we give time information also as covariates, we learn a weight for the positional encoding and a weight for the input vectors, which are both initialized as one.

4.2 Convolutional Self-Attention Transformer

The Convolutional Self-Attention Transformer differs from the Time-Series Transformer in that it uses convolutional self-attention [5] instead of the normal self-attention [2]. Before computing the keys and queries for the self-attention heads, causal 1D convolution with stride one and kernel size k is applied to the sequence of vectors.

Li et al. [5] additionally propose LogSparse attention to reduce the time and space complexity of the attention. Since we do not notice memory issues in our experiments, we do not make use of the LogSparse attention.

[2] It would be fine to attend to future time steps, since all features are already known at prediction time. However, we kept the masking to be consistent with the standard encoder-decoder architecture.

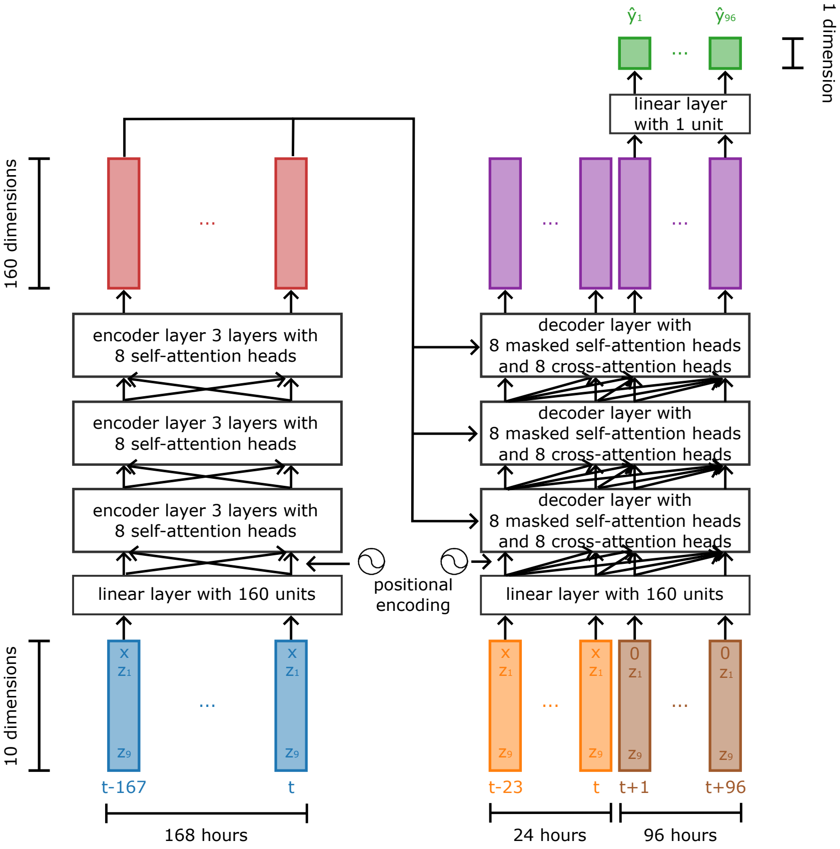
ŷ1
ŷ96
1 dimension
linear layer with 1 unit
160 dimensions
encoder layer 3 layers with 8 self-attention heads
encoder layer 3 layers with 8 self-attention heads
encoder layer 3 layers with 8 self-attention heads
decoder layer with 8 masked self-attention heads and 8 cross-attention heads
decoder layer with 8 masked self-attention heads and 8 cross-attention heads
decoder layer with 8 masked self-attention heads and 8 cross-attention heads
positional encoding
linear layer with 160 units
linear layer with 160 units
10 dimensions
x
z1
z9
0
t-167
t
t-23
t
t+1
t+96
168 hours
24 hours
96 hours

Figure 1: Data flow in the Time-Series Transformer. The architecture consists of an encoder part (left-hand side) and a decoder part (right-hand side). The input vectors to the encoder are shown in blue, and the output of the encoder in red. The decoder receives vectors for the previous day (orange) and next four days (brown). Each decoder layer attends to the encoder output (red) with multi-head cross-attention. Additionally, each encoder and decoder layer attends to its inputs with multi-head self-attention. The decoder output (purple) corresponding to the next day is fed through a linear layer to compute the predicted electrical load (green).

4.3 Informer

Compared to the Time-Series Transformer, the Informer [6] has two additional layers after each encoder layer. The first additional layer is a convolutional layer. The second additional layer is a max-pooling layer. The additional max-pooling layers cut the length of the time series in half after each decoder layer.

Zhou et al. [6] additionally propose ProbSparse attention to reduce the time and space complexity of the attention mechanism. Since we do not notice memory problems in our experiments, and the results of the Informer were slightly worse with ProbSparse attention, we use normal attention instead.

5 Experiments

Next, we describe the dataset in Section 5.1, the baselines in Section 5.2, the models in Section 5.3, the evaluation metric in Section 5.4, and the results in Section 3.3. Limitations are discussed in Section 5.6. Finally, an analysis of the Time-Series Transformer's attention scores is presented in Section 5.7.

5.1 Dataset

The selected dataset for the experiments is the electrical load of Baden-Württemberg from the Open Power System Data time-series dataset [16].[3] It contains the electrical load in MW for every quarter of an hour from 2015 to 2019. In order to shorten the time series, we transform the data to the hourly resolution by averaging every consecutive four values. We use the data from 2015 to 2017 as the training set, 2018 as the validation set, and 2019 as the test set. This gives 26,016 examples for training, and 8,496 each for validation and test. The dataset is standardized using the mean and standard deviation from the training set.

We use the following time and calendar features as covariates: the hour of the day, the week of the year (both sine- and cosine-encoded), whether the day is a workday, whether the day is a holiday, whether the previous day is a workday, whether the next day is a workday, and whether the day is in the

[3] `https://data.open-power-system-data.org/time_series/`

Christmas period from December 24th to 27th (all binary). Overall, this makes nine covariates.

5.2 Baselines

We compare the different Transformer architectures with three baselines: a load profile baseline, a linear regression model, and multi-layer perceptrons.

Load profile baseline We create daily profiles by computing the average of the load for every hour of each combination of month and day of the week. This makes twelve times seven daily profiles, each consisting of 24 averaged hourly load values. Holidays are treated like Sundays and seven special daily profiles computed for the two weeks after Christmas. At inference time, the profile corresponding to the day in question is used as predictions.

Linear regression The second baseline is a multi-output linear regression model. It gets the last 168 values of the electrical load time series as input, together with the nine time and calendar features for the first hour to predict, and predicts the next 96 electrical load values. The model has $\tau \cdot (\mathit{number\ of\ inputs} + 1)$ parameters, which in our case is $96 \cdot 178 = 17{,}088$.

Multi-layer perceptrons The multi-layer perceptrons (MLPs) use the same inputs as the linear regression models. The MLPs consist of multiple hidden layers with ReLU activation, and an output layer with linear activation with 96 units for the 96 predicted values. Results for MLPs with one to three layers and 256 to 2048 units are reported in Table 1. We choose a small MLP with two layers and 256 units per layer and a large MLP with two layers and 2048 units per layer as baselines for the Transformer models.

Table 1: MAPE on the test set for multi-layer perceptrons with varying numbers of layers and units. The results are averaged across ten runs with different random seeds.

Layers	Units per layer	MAPE [%]
1	256	3.33
1	512	3.22
1	1024	3.17
1	2048	3.15
2	256	3.33
2	512	3.20
2	1024	3.15
2	2048	3.12
3	256	3.34
3	512	3.22
3	1024	3.16
3	2048	3.12

5.3 Models

The Transformer models are the Time-Series Transformer, the Convolutional Self-Attention Transformer and the Informer, described in Section 4. We use vectors for the previous $p = 168$ time steps (i.e. one week) as input to the encoder, and vectors for the previous $p_d = 24$ time steps together with the $\tau = 96$ next time steps as input to the decoder. Each model consists of three encoder and three decoder layers with eight heads for the attention modules. The model dimension d_{model} is set to 160. The kernel size k for the Convolutional Self-Attention Transformer is set to twelve, and the kernel size for the Informer to three. We also tested Transformer models with a different number of layers, and varying model dimensions d_{model} and kernel sizes k, and found this architecture to be optimal among all tested variants.

An overview of the model sizes is given in Table 2. Notably, the large MLP has more trainable parameters than the Transformer models. The Convolutional Self-Attention Transformer and Informer have more trainable parameters than the Time-Series Transformer due to the additional convolutional layers.

Table 2: Model sizes.

Model	Layers	Units or d_{model}	#parameters
Linear regression	-	-	17,088
MLP small	2	256	233,056
MLP large	2	2,048	5,533,792
Time-Series Transformer	3 + 3	160	1,245,605
Conv. Self-Att. Transformer	3 + 3	160	4,013,285
Informer	3 + 3	160	1,400,165

All models are trained with the mean absolute error (MAE) as the loss function using the AdamW [17] optimizer. The initial learning rate is 0.0005, which is decayed by 90% after every two epochs. The batch size is set to 32. Early stopping is used to achieve the lowest possible generalization error on the validation data, with a patience of five epochs. The model with the lowest validation error is saved and used in the evaluation. We find that due to early stopping, the MLPs improve the validation error for no longer than 18 epochs, and the Transformer models for no longer than eight epochs. A possible explanation for the short training is the high number of trainable parameters in the models compared to the few training examples, which lets the models overfit easily.

5.4 Metric

To evaluate the performance of a model, we compute its mean absolute percentage error (MAPE) for forecasts from 1 to τ hours into the future. For each hour t in the test dataset, we have a vector $\hat{y}_t \in \mathbb{R}^\tau$ with the predicted electrical load for the next τ hours, and a vector $y_t \in \mathbb{R}^\tau$ with the actual electrical load of the next τ hours. We denote the i^{th} entry in y_t as $y_{t,i}$ and the i^{th} entry in $\hat{y}_t$ as $\hat{y}_{t,i}$. We evaluate the MAPE for forecasting T hours into the future, called MAPE_T, with $1 \leq T \leq \tau$. It is computed as follows:

$$\text{MAPE}_T(y,\hat{y}) = \frac{1}{N} \cdot \sum_{t=1}^{N} |\frac{y_{t,T} - \hat{y}_{t,T}}{y_{t,T}}|,$$

where N is the number of examples in the test set.

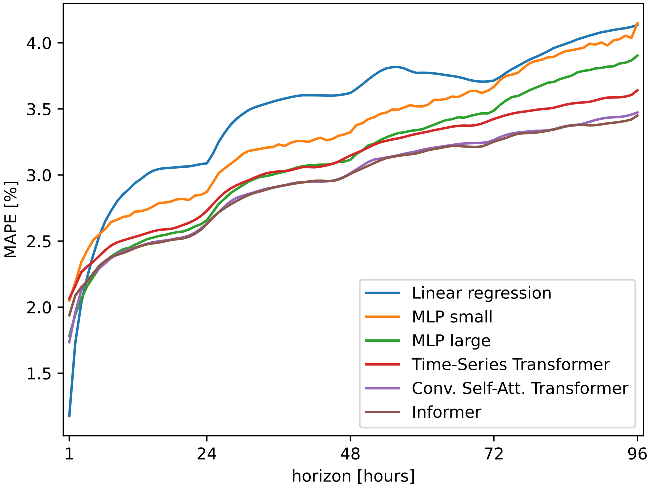

Figure 2: Results of the different models and baselines for predicting the electrical load from one to 96 hours into the future, evaluated on the test set.

5.5 Results

The results of the different models evaluated on the test set for forecasting horizons from one to 96 hours are shown in Figure 2. The results are averaged across ten runs with different random initializations of the neural networks' parameters.

All models are better than the load profile baseline, which has a constant MAPE value of 4.92% (not shown in the figure). For short prediction horizons of one to three hours, the linear regression is the best model. However, its MAPE value increases rapidly and after seven hours it is already the worst model. The large MLP is always better than the small MLP and is the best model for prediction horizons of four and five hours. For prediction horizons of six hours and more, either the Convolutional Self-Attention Transformer or the Informer is the best model. The Time-Series Transformer is worse than the other two Transformer variants, but it is also better than the MLPs for prediction horizons

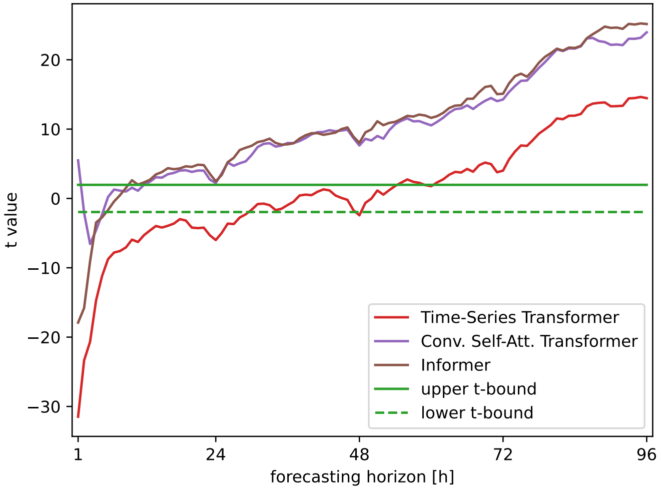

Figure 3: Results of Welch's t-tests with $\alpha = 0.025$ for each forecasting horizon. The Time-Series Transformer, Convolutional Self-Attention Transformer and Informer are compared to the MLP with two layers and 2048 units per layer.

longer than two days. Doing a Welch's t-test with $\alpha = 0.025$, we find that the Informer and the Convolutional Self-Attention Transformer are significantly better than the large MLP after 10 and 12 hours respectively, and the Time-Series Transformer is significantly better than the large MLP after 61 hours (see Figure 3). The Convolutional Self-Attention Transformer and the Informer are significantly better than the Time-Series Transformer for all forecasting horizons.

5.6 Discussion

In our experiments, Transformers beat the baselines for long prediction horizons, which shows their potential in electrical load forecasting. However, a comparison to other machine learning methods, such as random forests, support vector regression, long-short-term memories, convolutional neural networks and profile neural networks [14], must be made in the future. Also, some of the methods could benefit from feature engineering, feature selection and inclusion of external features such as weather data more than others, which could change the results. We notice that our models have many trainable parameters compared to the small number of training examples, and train only for a few epochs because of early stopping. Other training hyperparameters and more training data could lead to better results. We have compared three Transformer architectures in our work, and would like to include more in the future, for example Temporal Fusion Transformer [4], Autoformer [11] and FEDformer [18]. We have only used one dataset in our work, which contains the electrical load of the state Baden-Württemberg. An evaluation on more datasets, for example on the less aggregated and therefore more volatile load of buildings, and on other forecasting tasks, such as renewable energy generation forecasting, would help to analyze the usefulness of Transformers for energy time-series forecasting.

5.7 Attention Scores

Figure 4 shows an exemplary plot of the input and output time series and the attention scores of the Time-Series Transformer. The last observed hours before the prediction get a high attention when the model predicts the next few hours (see number 1 in the figure). The previous day is attended mostly when the model predicts the next day (2). A diagonal pattern can be seen, which means the model attends the embeddings from about 24 hours before the prediction. The valleys in the time series are attended when the model predicts the electrical load at night (3). Peaks in the time series are attended when the model predicts the electrical load at daytime (4). The patterns for valleys (3) and peaks (4) are similar, but shifted along the y-axis (that is, shifted with the prediction horizon). Similar patterns are seen for the other weekdays of the blue curve. The lowest values at Sunday mornings are always attended, even when the model does not make a prediction for a weekend (5).

We can use the attention scores as a plausibility check for the model. It is reasonable that the last observed hours are important to predict the next few hours, since they will often have similar values. The attention on peaks and valleys can be understood, because the rest of the time series can be inferred

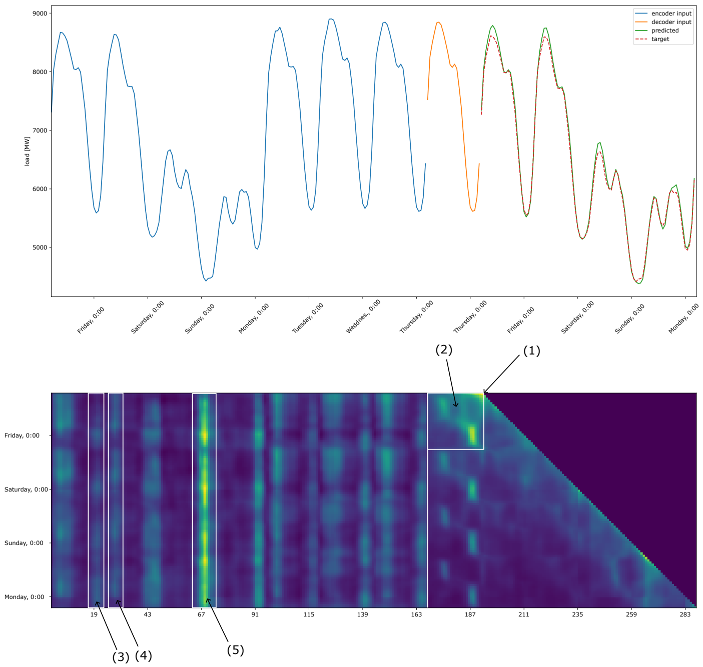

Figure 4: Visualization of the decoder's attention scores averaged across all Thursdays at 5 a.m. that appear in the test set. The upper part shows four averaged time series: the input to the encoder (blue), the previous day fed to the decoder (orange), the predictions (green) and expected values (red). The lower part shows the cross-attention on the left and self-attention on the right. Each row corresponds to one prediction time step, from top to bottom in chronological order. The lighter the color, the higher the attention score. The upper right triangle of the self-attention consists of zeros because of the masked self-attention, that prevents the model from attending future time steps.

from its highest and lowest values. Peaks are important at daytime when the predicted values are high, and valleys at night when the predicted values are low. However, we had expected that the model would attend the previous weekend only while making a forecast for the weekend. In addition, we had expected more of a diagonal pattern in the cross-attention scores, meaning that the model would attend the same weekday a week ago.

In the future, we want to apply Transformers with multiple time series as input, such as additional weather data, and use the attention scores to estimate feature importance. Another possible direction of future research is to adapt the attention scores such that the Transformer attends more on the previous weekday and less on the previous Sunday, and investigate whether this improves or deteriorates the performance. A third option is to select the features that received high attention, such as the peaks and valleys, and use them in another model. The resulting model could achieve the best of two worlds: the good performance of the Transformer, and the fast training and inference of simpler methods.

6 Conclusion and Future Work

Our experiments showed that Transformers give better electrical load forecasts for the state Baden-Württemberg than multiple statistical baselines and multilayer perceptrons. In the future, a comparison to other strong machine learning methods must be made. We plan to integrate the Transformers into the Python Workflow Automation Tool for Time-Series (pyWATTS) [19] and evaluate them against the models already included in the package. In addition, we want to incorporate external features such as weather data, and evaluate if the model ranking remains the same. Transformers could also be useful for other energy forecasting tasks, such as forecasting the more volatile electrical loads of individual buildings or forecasting renewable energy generation.

The exact details of the Transformer architecture are important to get the best results, as in our experiments the Convolutional Self-Attention Transformer and the Informer were better than the Time-Series Transformer. More architectures from the literature [4, 11, 18] could be added to the comparison, and new architectures developed.

Promising research directions are to use the Transformer's attention scores to better understand the models and get closer towards explainable AI methods, or to use the gained insights to develop new models.

Acknowledgement

This project is funded by the Helmholtz Association's Initiative and Networking Fund through Helmholtz AI.

References

[1] Jan Machowski et al. "Power system dynamics and stability". John Wiley & Sons, 1997.

[2] Ashish Vaswani et al. "Attention is all you need". In: *Advances in Neural Information Processing Systems* 30 (2017).

[3] Neo Wu et al. "Deep transformer models for time series forecasting: The influenza prevalence case". In: *arXiv preprint arXiv:2001.08317* (2020).

[4] Bryan Lim et al. "Temporal Fusion Transformers for Interpretable Multi-horizon Time Series Forecasting". In: *CoRR abs/1912.09363* (2019).

[5] Shiyang Li et al. "Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting". In: *Advances in Neural Information Processing Systems* 32 (2019).

[6] Haoyi Zhou et al. "Informer: Beyond efficient transformer for long sequence time-series forecasting". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 35. 12. 2021, pp. 11106–11115

[7] Guangqi Zhang et al. "Short-Term Electrical Load Forecasting Based on Time Augmented Transformer". In: *International Journal of Computational Intelligence Systems* 15.1 (2022), pp. 1–11.

[8] Shichao Huang et al. “Short-Term Load Forecasting Based on the CEEMDAN-Sample Entropy-BPNN-Transformer”. In: *Energies* 15.10 (2022), p. 3659.

[9] Alexandra L’Heureux, Katarina Grolinger, and Miriam AM Capretz. “Transformer-Based Model for Electrical Load Forecasting”. In: *Energies* 15.14 (2022), p. 4993.

[10] Chen Wang et al. “A Transformer-Based Method of Multienergy Load Forecasting in Integrated Energy System”. In: *IEEE Transactions on Smart Grid* 13.4 (2022), pp. 2703–2714.

[11] Haixu Wu et al. “Autoformer: Decomposition Transformers with Auto-Correlation for Long-Term Series Forecasting”. In: *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*. Ed. by Marc’Aurelio Ranzato et al. 2021, pp. 22419–22430.

[12] Zezheng Zhao et al. “Short-Term Load Forecasting Based on the Transformer Model”. In: *Inf.* 12.12 (2021), p. 516

[13] Jorge Ángel González Ordiano et al. “Energy forecasting tools and services”. In: *WIREs Data Mining Knowl. Discov.* 8.2 (2018).

[14] Benedikt Heidrich et al. “Forecasting energy time series with profile neural networks”. In: *Proceedings of the Eleventh ACM International Conference on Future Energy Systems*. 2020, pp. 220–230.

[15] Ailing Zeng et al. “Are Transformers Effective for Time Series Forecasting?” In: *CoRR abs/2205.13504* (2022).

[16] Frauke Wiese et al. “Open Power System Data–Frictionless data for electricity system modelling”. In: *Applied Energy* 236 (2019), pp. 401–409.

[17] Ilya Loshchilov and Frank Hutter. “Decoupled Weight Decay Regularization”. In: *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*.

[18] Tian Zhou et al. “FEDformer: Frequency Enhanced Decomposed Transformer for Long-term Series Forecasting”. In: *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*. Ed. by Kamalika Chaudhuri et al. Vol. 162. Proceedings of Machine Learning Research. PMLR, 2022, pp. 27268–27286.

[19] Benedikt Heidrich et al. “PyWATTS: Python workflow automation tool for time series”. In: *arXiv preprint arXiv:2106.10157* (2021).