Transformer Neural Networks for Building Load Forecasting

Matthias Hertel, Simon Ott, Oliver Neumann, Benjamin Schäfer, Ralf Mikut, Veit Hagenmeyer
Karlsruhe Institute of Technology (KIT)
Institute for Automation and Applied Informatics (IAI)
Hermann-von-Helmholtz-Platz 1
76344 Eggenstein-Leopoldshafen
matthias.hertel@kit.edu

Abstract

Accurate electrical load forecasts of buildings are needed to optimize local energy storage and to make use of demand-side flexibility. We study the usage of Transformer neural networks for short-term electrical load forecasting of 296 buildings from a public dataset. Transformer neural networks trained on many buildings give the best forecasts on 115 buildings, and multi-layer perceptrons trained on a single building are better on 161 buildings. In addition, we evaluate the models on buildings that were not used for training, and find that Transformer neural networks generalize better than multi-layer perceptrons and our statistical baselines. This shows that the usage of Transformer neural networks for building load forecasting could reduce training resources due to the good generalization to unseen buildings, and they could be useful for cold-start scenarios.

1 Introduction

An increasing amount of buildings is equipped with photovoltaic modules and local batteries or flexible devices like electric vehicle chargers and heat pumps. Photovoltaic feed-in limits can result in curtailment of solar energy. To avoid curtailment, optimal building control strategies make use of flexible consumption or local battery storage, in order to consume most of the generated energy locally [1]. Such control algorithms require short-term building load forecasts to create a dispatch plan for the next hours.

Transformer neural networks [2] are state of the art in many natural language processing tasks, and promising for time-series forecasting. Our paper studies the usage of Transformer neural networks for forecasting the electrical load of buildings. The training of such models consumes energy and thereby causes carbon emissions. We therefore study whether models trained on a subset of the buildings generalize to unseen buildings. This would reduce the overall energy consumption and carbon emissions of the training, since training models for every building could be avoided. It is also a requirement for cold-start applications, where no training data is available for a building.

2 Related work

González Ordiano et al. [3] give an overview on existing energy time-series forecasting methods, including linear regression and multi-layer perceptrons, which we use as baseline methods (see Section 5.3). Haben et al. [4] review applications and methods for low-voltage level forecasting, but do not cover Transformer neural networks, which were applied to time-series forecasting [5–9] and electrical load forecasting [10–16] only recently. Li et al. [7] investigate Transformer neural networks for multiple forecasting tasks, including the dataset that we use, but only evaluate models trained on the data from all buildings, whereas we also train models for individual buildings. Zeng et al. [17] question whether Transformer neural networks are effective for time series forecasting and show that they are often outperformed by a one-layer linear model.

3 Task definition

We address the following electrical load forecasting problem: At a time step t, given a building's hourly electrical load of the previous k time steps $x_{(t-k+1):t} = (x_{t-k+1}, ..., x_t)$, m covariate sequences $z^j_{(t-k+1):t}$, and n a priori known covariate sequences $z^l_{(t+1):(t+\tau)}$, 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. $k = 168$), and a forecasting horizon of $\tau = 24$ hours. We use the following time and calendar features as covariates: hour of the day, day of the week, month (all 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 Christmas time from December 24th to 27th (all binary).

4 Approach

We use the time-series Transformer [6] architecture for our models. An overview of the architecture is shown in Figure 1. It consists of an encoder part and a decoder part, described next.

The input to the encoder is a sequence of 168 vectors, one for each hour of the preceding week. Each vector contains 12 entries: one for the electrical load and 11 for the time and calendar features for this time step. Before feeding the vectors to the encoder, we run them through a linear layer with $d_{\text{model}} = 160$ units. The encoder consists of multiple layers with multi-head self-attention with eight heads. Each encoder layer gets an input of shape $168 \times d_{\text{model}}$ and produces an output of the same shape.

The input to the decoder contains the vectors for the last 24 time steps and the next 24 time steps. The electrical load for the next 24 time steps is unknown at prediction time and therefore set to zero. The vectors are also run through a linear layer to increase the vector size to $d_{\text{model}} = 160$. The decoder consists of multiple layers. Each decoder layer attends to the outputs of the previous layer with multi-head self-attention with eight heads. Masking prevents the self-attention to attend to outputs that correspond to future time steps. In addition, each decoder layer attends to the outputs of the last encoder layer with multi-head cross-attention with eight heads. The last 24 outputs of the decoder, which correspond to the 24 future time steps, are fed into a linear layer with a single unit. This results in 24 predictions for the next 24 hours.

This architecture outperformed the statistical baselines, a linear regression model and MLPs of different sizes in previous experiments on forecasting the aggregated electrical load of the German state Baden-Württemberg for long prediction horizons [16], and we now test it on the more volatile electrical load of individual buildings. We test Transformer neural networks with one to three encoder and decoder layers, and choose the model with the best result on the validation period.

5 Experiments

5.1 Dataset

We use a public dataset from the UCI Machine Learning Repository for our experiments.[1] The dataset contains electrical load time series in kW for 370 clients in Portugal. We manually removed buildings whose time series looked erroneous, synthetic or had lots of missing data. After this step, 296 buildings remained in the dataset. The time series range from 2011 to 2014, with a time resolution of 15 minutes, which we transform to hourly resolution by averaging every four consecutive values. For some clients the first values are missing, which results in shorter time series. We use the last six months from the dataset (July to December 2014) as test period, the six months before as validation period (January to June 2014), and the remaining as training period. To evaluate on buildings that were not used for training, which we call the *unseen* buildings, we perform a five-fold

[1] `https://archive.ics.uci.edu/ml/datasets/ElectricityLoadDiagrams20112014`

cross-validation. In every fold of the cross-validation, 80% of the buildings are used for training, and 20% for testing. The validation is done on the validation period with the buildings used for training. The data is standardized with the mean and standard deviation of the training data of each model (the training data differs for local and global models, see Section 5).

5.2 Baselines

Persistence: This baseline predicts the same value as was observed one week ago. We also tested it with the value of a day ago, and one week was better.

Weekly profile: We compute a weekly profile by averaging the electrical load of each hour of the week across the training data. This results in 168 averaged values, one for each hour of the week, which are used as predictions.

Linear regression: The third baseline is a multi-output linear regression model [18]. It gets the last 168 values of the electrical load time series as input, together with the 11 time and calendar features for the first hour to predict, and predicts the following 24 electrical load values.

Random forest regression: The random forest regression [19] models consist of 100 trees with a maximum depth of 10. The input and output is the same as for the linear regression models.

Multi-layer perceptron: The multi-layer perceptron (MLPs) [18] uses the same inputs as the linear regression and random forest regression models. The MLP consist of two hidden layers with ReLU activation [20], and an output layer with linear activation with 24 units for the 24 predicted values. We compare MLPs with between 256 and 2048 units per hidden layer and choose the MLP with the best result on the validation period.

5.3 Local and global models

We distinguish between models trained on the time series from a single building, which we call *local models*, and models trained on the time series from all buildings, which we call *global models*. Local models are only evaluated on the building they were trained on.

All neural networks are trained with the AdamW [21] optimizer with batch size 128, an initial learning rate of 0.0005 which gets reduced by 90% every two epochs, and early stopping with a patience of five epochs. Training was done on a NVIDIA GeForce 2080 Ti GPU.

5.4 Metric

To evaluate our models, we compute the normalized mean absolute error (NMAE) on each building. The NMAE is the mean absolute error (MAE) divided by the mean observed value. For every hour i in the test set, we have two vectors $\hat{y}_i \in \mathbb{R}^{24}$ and $y_i \in \mathbb{R}^{24}$, containing the predicted and actual electrical load for the next 24 hours. The NMAE is computed as

$$NMAE(y, \hat{y}) = \frac{\sum_{i=1}^{n} \sum_{j=1}^{24} |y_{i,j} - \hat{y}_{i,j}|}{n \cdot 24} \cdot \frac{1}{\bar{y}},$$

where n is the number hours in the test set, and $\bar{y}$ the mean observed electrical load for the building during the test period. The NMAE values of the global models are averaged across all buildings. For the local models, we average the NMAE values of all local models evaluated on the building they were trained on (evaluations are done on the test period, which was not used for training).

5.5 Results

Table 1 shows the results of the different models, separately evaluated on the buildings in the training set and the unseen buildings in the generalization set. Figure 2 shows a bar plot of the NMAE results. The local models have a better NMAE than their global counterparts, except for the Transformer neural networks, where the global model is better than the local models. The local MLP has the lowest NMAE, followed by the local linear regression and the global Transformer neural network. No single model is best on all buildings. The local MLP is the best model on the majority of the buildings, and the global Transformer neural network is best on more than one third of the buildings. The global Transformer neural network has the lowest NMAE among the global models

Table 1: Results on buildings that were seen during training and buildings that were not seen during training, evaluated on the test period. The normalized mean absolute error (NMAE) is averaged across the buildings. ”#best” indicates on how many buildings a model is the overall best model, best global model and best local model. Training times of the local models are summed up for all buildings.

Method	**Seen buildings**				**Unseen buildings**		**Training**
	NMAE	#best	#best	#best	NMAE	#best	time
	[%]	overall	local	global	[%]	overall	[minutes]
Persistence	10.41	3	6	-	10.41	24	-
Weekly profile	18.28	0	0	-	-	-	-
Linear regression local	8.04	4	14	-	-	-	1.0
Random forest local	9.23	0	0	-	-	-	81.4
MLP local	7.15	161	264	-	-	-	29.6
Transformer local	8.30	11	12	-	-	-	454.5
Linear regression global	13.44	1	-	3	13.41	6	1.2
Random forest global	21.29	0	-	0	22.22	0	186.2
MLP global	8.57	1	-	54	8.60	48	21.2
Transformer global	8.10	115	-	239	8.53	218	338.3

both on buildings seen during training and on unseen buildings, but the difference to the global MLP is small. The Transformer neural network is the best global model on 80% of the buildings seen during training and the best model on 73% of the unseen buildings. However, the global MLP is better on more buildings during the validation period (see Table 2).

6 Conclusion and future work

We found that Transformer neural networks generalize well to unseen buildings during the six-months test period, and therefore could be used to build generally applicable forecasting models for household demand. Transformer neural networks are promising for cold-start scenarios, where little or no training data is available for a building, and for scenarios where training separate models for all buildings is not feasible. However, we also found that the Transformer neural networks need more time to train than the MLPs.

There are buildings where the local MLP is the best model, and others where the global Transformer neural network is the best. In the future, we want to characterize buildings for which the Transformer neural networks are the best model, and give advice to practitioners on when to use which model.

We see room for improved forecasts. External features such as weather data, other Transformer architectures [5, 7–9, 14] and data augmentation [22] could improve the results. Another possible improvement is to make use of dependencies between buildings, e.g. with transfer learning [23], by clustering similar buildings [24], or by using the time series of other buildings as covariates.

In the future, the carbon emissions from training and using a model should be evaluated and compared with the potential benefit of more accurate forecasts.

7 Acknowledgements

We thank Kumar Saurav (IBM Research) for being the mentor for this work within the CCAI mentorship program. This work is funded by the Helmholtz Association’s Initiative and Networking Fund through Helmholtz AI, and the Helmholtz Association under grant no. VH-NG-1727.

References

[1] Yannick Riesen, Christophe Ballif, and Nicolas Wyrsch. "Control algorithm for a residential photovoltaic system with storage". In: *Applied Energy* 202 (2017), pp. 78–87.

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

[3] Jorge Ángel González Ordiano et al. "Energy forecasting tools and services". In: *WIREs Data Mining Knowl. Discov.* 8.2 (2018). DOI: 10.1002/widm.1235. URL: https://doi.org/10.1002/widm.1235.

[4] Stephen Haben et al. "Review of low voltage load forecasting: Methods, applications, and recommendations". In: *Applied Energy* 304 (2021), p. 117798.

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

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

[7] 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).

[8] 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.

[9] 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. URL: https://proceedings.mlr.press/v162/zhou22g.html.

[10] 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.

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

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

[13] 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.

[14] 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. URL: https://proceedings.neurips.cc/paper/2021/hash/bcc0d400288793e8bdcd7c19a8ac0c2b-Abstract.html.

[15] Zezheng Zhao et al. "Short-Term Load Forecasting Based on the Transformer Model". In: *Inf.* 12.12 (2021), p. 516. DOI: 10.3390/info12120516. URL: https://doi.org/10.3390/info12120516.

[16] Matthias Hertel et al. "Evaluation of Transformer Architectures for Electrical Load Time-Series Forecasting". In: *32. Workshop Computational Intelligence* (2022).

[17] Ailing Zeng et al. "Are Transformers Effective for Time Series Forecasting?" In: *CoRR* abs/2205.13504 (2022). DOI: 10.48550/arXiv.2205.13504. arXiv: 2205.13504. URL: https://doi.org/10.48550/arXiv.2205.13504.

[18] Trevor Hastie et al. *The elements of statistical learning: data mining, inference, and prediction*. Vol. 2. Springer, 2009.

[19] Leo Breiman. "Random Forests". In: *Mach. Learn.* 45.1 (2001), pp. 5–32. DOI: 10.1023/A:1010933404324. URL: https://doi.org/10.1023/A:1010933404324.

[20] *A gentle introduction to the rectified linear unit (ReLU)*. https://machinelearningmastery.com/rectified-linear-activation-function-for-deep-learning-neural-networks/. Accessed: 2022-09-16.

[21] 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*. OpenReview.net, 2019. URL: `https://openreview.net/forum?id=Bkg6RiCqY7`.

[22] Benedikt Heidrich et al. “Boost short-term load forecasts with synthetic data from transferred latent space information”. In: *Energy Informatics* 5.1 (2022), pp. 1–20.

[23] Marcus Voß, Christian Bender-Saebelkampf, and Sahin Albayrak. “Residential Short-Term Load Forecasting Using Convolutional Neural Networks”. In: *2018 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids, SmartGridComm 2018, Aalborg, Denmark, October 29-31, 2018*. IEEE, 2018, pp. 1–6. DOI: `10.1109/SmartGridComm.2018.8587494`. URL: `https://doi.org/10.1109/SmartGridComm.2018.8587494`.

[24] Heng Shi, Minghao Xu, and Ran Li. “Deep learning for household load forecasting—A novel pooling deep RNN”. In: *IEEE Transactions on Smart Grid* 9.5 (2017), pp. 5271–5280.

Appendix

A Transformer neural network architecture

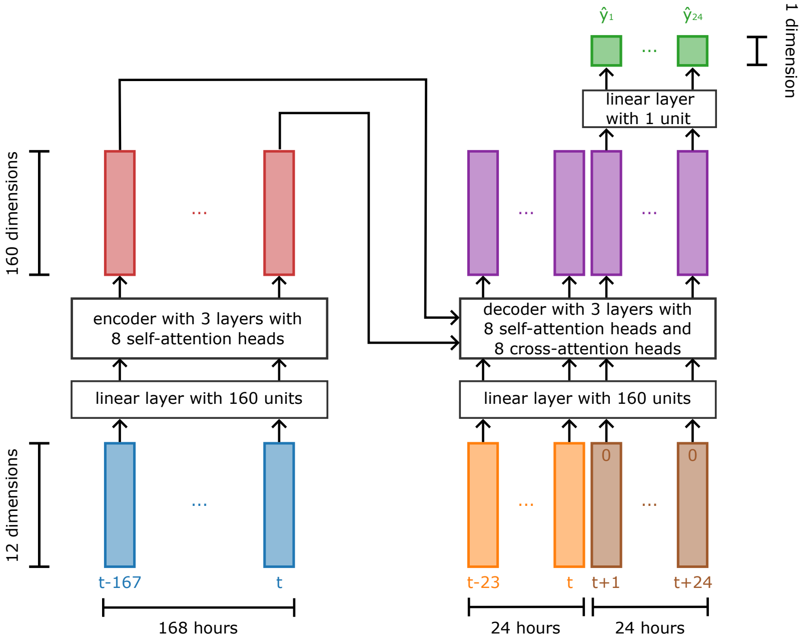
$\hat{y}_1$
...
$\hat{y}_{24}$
1 dimension
linear layer with 1 unit
160 dimensions
...
...
...
encoder with 3 layers with 8 self-attention heads
decoder with 3 layers with 8 self-attention heads and 8 cross-attention heads
linear layer with 160 units
linear layer with 160 units
12 dimensions
...
...
0
...
0
t-167
t
t-23
t
t+1
t+24
168 hours
24 hours
24 hours

Figure 1: Data flow in the Transformer neural network. 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 day (brown). Each decoder layer attends to the encoder output (red) with multi-head cross-attention (arrows from left to right). Additionally, each encoder and decoder layer attends to its inputs with multi-head self-attention (not shown in the figure). The decoder output (purple) corresponding to the next day is fed through a linear layer to compute the predicted electrical load (green).

B Bar plots

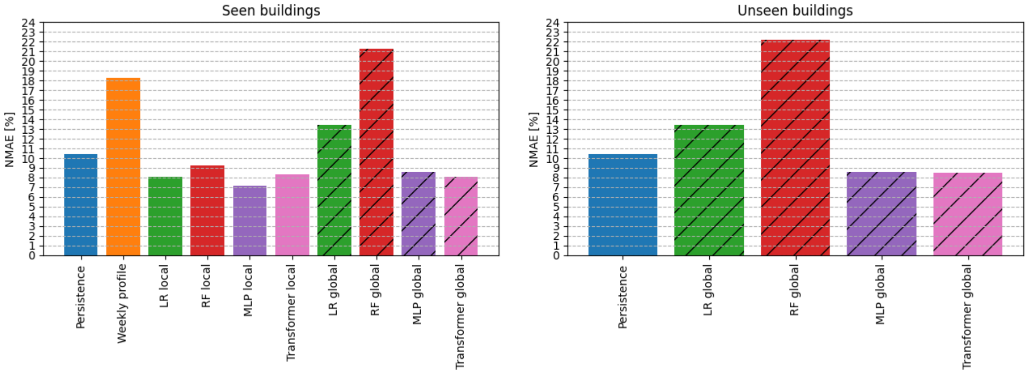

Figure 2: NMAE on the seen buildings (left) and unseen buildings (right).

C Validation results

Table 2: Results on the validation period.

Method	**Seen buildings**				**Unseen buildings**	
	NMAE [%]	#best overall	#best local	#best global	NMAE [%]	#best overall
Persistence	12.90	0	1	-	12.90	6
Weekly profile	19.71	0	0	-	-	-
Linear regression local	8.82	10	21	-	-	-
Random forest local	10.03	0	0	-	-	-
MLP local	7.97	220	263	-	-	-
Transformer local	9.07	10	11	-	-	-
Linear regression global	12.97	1	-	4	12.86	15
Random forest global	21.60	0	-	0	22.94	0
MLP global	8.88	23	-	199	9.02	176
Transformer global	9.26	32	-	93	10.10	99