Hertel *et al. Energy Informatics* 2023, **6**(Suppl 1):20
https://doi.org/10.1186/s42162-023-00278-z

Energy Informatics

RESEARCH Open Access

Transformer training strategies for forecasting multiple load time series

Matthias Hertel[1*], Maximilian Beichter[1], Benedikt Heidrich[1], Oliver Neumann[1], Benjamin Schäfer[1], Ralf Mikut[1] and Veit Hagenmeyer[1]

From The 12th DACH+ Conference on Energy Informatics 2023
Vienna, Austria. 4-6 October 2023. https://www.energy-informatics2023.org/

*Correspondence:
matthias.hertel@kit.edu

[1] Karlsruhe Institute of Technology, Institute for Automation and Applied Informatics, Hermann-von-Helmholtz-Platz 1, 76344 Eggenstein-Leopoldshafen, Germany
Full list of author information is available at the end of the article

Abstract

In the smart grid of the future, accurate load forecasts on the level of individual clients can help to balance supply and demand locally and to prevent grid outages. While the number of monitored clients will increase with the ongoing smart meter rollout, the amount of data per client will always be limited. We evaluate whether a Transformer load forecasting model benefits from a transfer learning strategy, where a global univariate model is trained on the load time series from multiple clients. In experiments with two datasets containing load time series from several hundred clients, we find that the global training strategy is superior to the multivariate and local training strategies used in related work. On average, the global training strategy results in 21.8% and 12.8% lower forecasting errors than the two other strategies, measured across forecasting horizons from one day to one month into the future. A comparison to linear models, multi-layer perceptrons and LSTMs shows that Transformers are effective for load forecasting when they are trained with the global training strategy.

Keywords: Load forecasting, Transformer, Global model, Time series, Smart grid

Introduction

Climate change is one of the biggest challenges facing humanity, with the risk of dramatic consequences if certain limits of warming are exceeded (Pörtner et al. 2022). To mitigate climate change, the energy system must be decarbonized. A difficulty in decarbonization is that renewable energy supply fluctuates depending on the weather. However, supply and demand must be balanced in the grid at every moment to prevent outages (Machowski et al. 1997). In addition, with the ongoing decentralization of the renewable energy supply and the installation of large consumers, such as electric vehicle chargers and heat pumps, low-voltage grids are expected to reach their limits (Çakmak and Hagenmeyer 2022). Thus, to balance the grid and to avoid congestions, advanced operation and control mechanisms must be installed in the smart grid of the future (Ramchurn et al. 2012; Haben et al. 2021). This requires accurate forecasts on various

aggregation levels, up to fine-grained low-voltage level load forecasts (Haben et al. 2021; Ordiano et al. 2018). Such fine-grained load forecasts can be used for demand-side management, energy management systems, distribution grid state estimation, grid management, storage optimization, peer-to-peer trading, peak shaving, smart electrical vehicle charging, dispatchable feeders, provision of feedback to customers, anomaly detection and intervention evaluation (Haben et al. 2021; Yildiz et al. 2017; Voß et al. 2018; Grabner et al. 2023; Werling et al. 2022). Moreover, the aggregation of fine-grained load forecasts can result in a more accurate forecast of the aggregated load (Hong et al. 2020).

With the smart meter rollout, fine-grained electrical load data will become available for an increasing number of clients. In such a scenario where load time series from multiple clients are available, different model training strategies are possible. The goal of our work is to compare training strategies for the Transformer (Vaswani et al. 2017), which was recently used for load forecasting (Zhang et al. 2022; Hertel et al. 2022a, b; Cao et al. 2022; Giacomazzi et al. 2023; Huy et al. 2022).

Task definition

We address the following multiple load time series forecasting problem: At a time step t, given the history of the electrical load of C clients $x_0^c, \dots x_t^c$ with $1 \leq c \leq C$, the goal is to predict the next h electrical load values $x_{t+1}^c, \dots, x_{t+h}^c$ for all clients $1 \leq c \leq C$, where h is called the forecast horizon.

Contribution

We compare three training strategies for the Transformer in a scenario with multiple load time series. The training strategies are depicted in Fig. 1.

1. A *multivariate* model training strategy, where a single model gets all load time series as input and forecasts all load time series simultaneously.
2. A *local* model training strategy, where a separate univariate[1] model is trained for each load time series.
3. A *global* model training strategy, where a generalized univariate model is used to forecast each load time series separately.

We compare our models with the models from related work (Zhou et al. 2021; Wu et al. 2021; Zhou et al. 2022; Nie et al. 2022), as well as with multiple baselines. In particular, we compare with the linear models used in Zeng et al. (2022), to figure out if Transformers are effective for load forecasting and which training strategy is the most promising one.

Paper structure

First, we describe the "Related work". Then, the Transformer architecture and the training strategies are described in the "Approach". This is followed by the "Experimental

[1] By 'univariate' we mean models which produce a forecast for a single time series. We still call models 'univariate' when they have multiple input variables, such as exogenous time and calendar features.

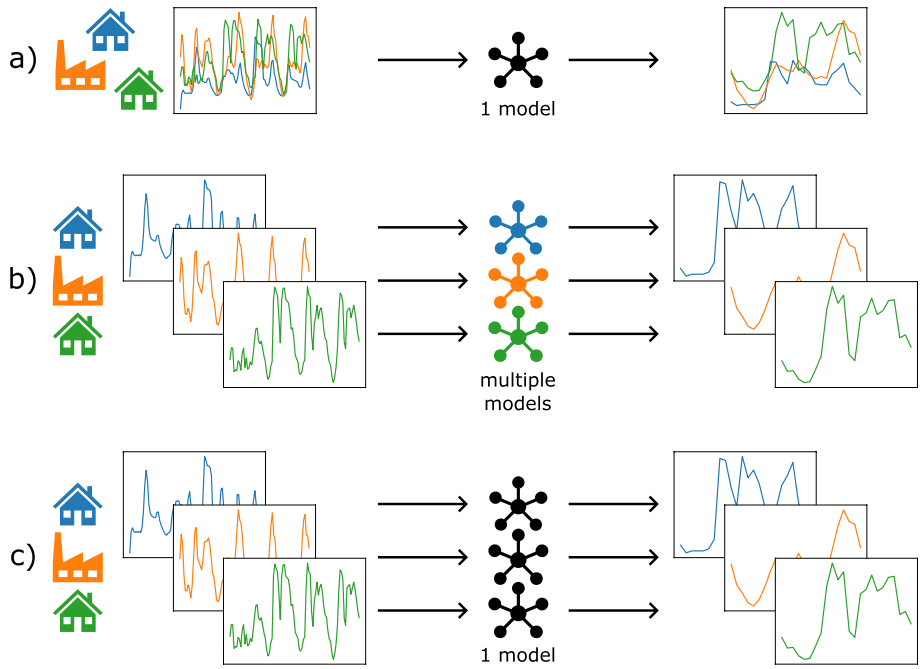

Fig. 1 The three training strategies, with models depicted as networks. An example with three load time series, four days input and one day output is shown. **a** Multivariate: one model processes all load time series simultaneously; **b** local: separate models (blue, orange, green) process each load time series; **c** global: one model (black) processes all load time series one at a time

setup", "Results" and a "Discussion". Finally, the paper concludes with the "Conclusion and future work".

Related work

This section first presents related work on long time series forecasting and load forecasting with Transformers. Most of the load forecasting literature uses local models, but few works use global models, which are presented next. The global training strategy can be understood as a transfer learning technique. We therefore discuss transfer learning in the field of load forecasting at the end of this section.

As Transformer are often used for long time series forecasting with up to one month horizon, various extensions to the Transformer architecture exist that aim to reduce the time and space complexity. This is done by the Informer using ProbSparse self-attention (Zhou et al. 2021), by the Autoformer using auto-correlation (Wu et al. 2021), by the FEDformer using frequency enhanced decomposition (Zhou et al. 2022) and by PatchTST using patching (Nie et al. 2022). The proposed models are multivariate or local, except for the global PatchTST (Nie et al. 2022). The experiments in these works are conducted on six datasets from different domains, including one load forecasting dataset, which we also use in our experiments (see section Datasets). A global linear model called LTSF-Linear (Zeng et al. 2022) gives better results than the aforementioned multivariate Transformers. Parallel to our work, global Transformers were shown to beat the aforementioned multivariate Transformers (Murphy and Chen 2023). However, this work does not optimize the model's lookback size and therefore achieves sub-optimal

results. PatchTST (Nie et al. 2022) is a global Transformer with patched inputs and is superior to LTSF-Linear (Zeng et al. 2022) on the six datasets.

Transformer architectures for short-term load forecasting are designed to use external calendar and weather features (Wang et al. 2022; Huy et al. 2022). An evaluation of different architectures is undertaken in Hertel et al. (2022a). Further work modifies the architecture for multi-energy load forecasting (Wang et al. 2022). Upstream decompositions are used to improve the forecast quality (Ran et al. 2023). These models are not compared on a common benchmark dataset, but evaluated on different datasets on city or national level. There, usually only one load time series is available, which only allows for local models. Furthermore, the models are not compared to the Transformer architectures for long time series.

Global load forecasting models are already used with convolutional neural networks (Voß et al. 2018) and N-BEATS (Grabner et al. 2023). A mixture between a multivariate and a global model is investigated in Shi et al. (2017), where a single recurrent neural network (RNN) model is trained on randomly pooled subsets of the time series. Some works cluster the time series and then train global or multivariate models for each cluster (Han et al. 2020; Yang and Youn 2021). PatchTST (Nie et al. 2022) is a global Transformer with patched inputs. We compare to this approach in our experiments.

The authors of Pinto et al. (2022) and Himeur et al. (2022) present current literature on transfer learning in the domain of energy systems. They define a taxonomy of transfer learning methods and discuss different strategies of using transfer learning with buildings from different domains. Two works (Nawar et al. 2023; Gao et al. 2022) use transfer learning by pre-training and fine-tuning Transformers. Transferability from one building to another is tested in Nawar et al. (2023), and from one district to another in Gao et al. (2022). In contrast to these works, our transfer learning approach is to train a generalized model on the data from many clients, without fine-tuning for a target time series.

Approach

We use an encoder–decoder Transformer (Vaswani et al. 2017) as a load forecasting model. This model architecture has self-attention and cross-attention as its main components and was initially used for machine translation. It was used as a forecasting model in Wu et al. (2020) and later adopted for load forecasting (Zhang et al. 2022; Hertel et al. 2022a, b). We use the model implementation from Hertel et al. (2022a).

The encoder gets L vectors as input, which represent the last L time steps, where L is called the lookback size. Each input vector consists of one (in the case of local and global models) or C (in the case of multivariate models) load values, and nine additional time and calendar features. The features are the hour of the day, the day of the week and the month (all cyclically encoded with a sine and a cosine function), whether it is a workday, whether it is a holiday and whether the next day is a workday (all binary features). The input to the decoder consists of h vectors, which represent the following h time steps for which a forecast will be made. In the decoder input, the load values are set to zero, so that each value is forecasted independently from the previous forecasted values, allowing for a direct multi-step forecast instead of generating all values iteratively. The input vectors to the encoder and the decoder are first fed through linear layers to increase the

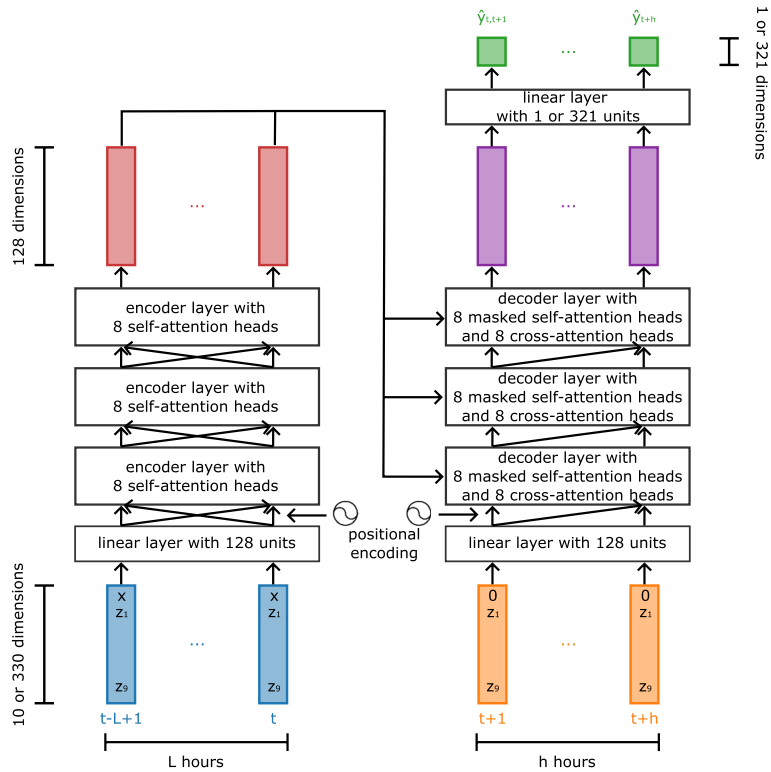

Fig. 2 Architecture of the Transformer forecasting model. The input and output dimensions differ for the multivariate model and the local and global models. The shown dimensions refer to the Electricity dataset with 321 clients

Table 1 Training strategy details for the *Electricity* dataset with 321 load time series, 2.1 years training data and nine time and calendar features

Training strategy	Models	Input size	Output size	Training data
Multivariate	1	$L \times 330$	$h \times 321$	2.1 years
Local	321	$L \times 10$	$h \times 1$	2.1 years
Global	1	$L \times 10$	$h \times 1$	$321 * 2.1$ years

For the local models, *training data* is the amount of training data per model

dimensionality to the hidden dimension of the model d_{model}. Both the encoder and the decoder consist of multiple layers with eight self-attention heads and the decoder layers have eight additional masked cross-attention heads. Finally, a linear layer transforms the h decoder output vectors into a forecast with $h \times 1$ (for local and global models) or $h \times C$ (for multivariate models) values. We varied the number of encoder and decoder layers and the hidden dimension d_{model}, and found three layers with $d_{\text{model}} = 128$ to give the best results. The full model architecture is shown in Fig. 2.

Training strategies

We compare multivariate, local and global Transformers. The training strategies are depicted in Fig. 1 and are further explained in the following. Details on the inputs,

outputs, number of models and training data size for each training strategy are given in Table 1.

- *Multivariate training strategy:* In the input to the model, each time step is represented by a vector of size $C + f$, where C is the number of load time series and f is the number of calendar features. The model forecasts C values for the next h time steps, i.e. its output consists of h vectors of size C. A single model is used to forecast all time series simultaneously.
- *Local training strategy:* Local models get only one time series as input and generate a forecast for this time series. In the input, each time step is represented by a vector with $f + 1$ entries for the f calendar features and the electrical load value. C separate models are trained for the C time series, each using the training data from one time series.
- *Global training strategy:* The global approach is a single model that generalizes for all load time series. The model gets one load time series as input and generates a forecast for that load time series. In contrast to the local models, only one global model is trained on samples from all load time series, and this model is used to forecast all load time series. This results in C times as many training data for the global model as for a local model. To generate forecasts for all C time series, the global model is used C times with the history of one load time series as input.

Experimental setup

Datasets

As recommended in recent literature reviews on load forecasting (Haben et al. 2021; Hong et al. 2020; vom Scheidt et al. 2020), we conduct experiments on multiple datasets, namely the *Electricity* and the *Ausgrid solar home* datasets. For both datasets we make a temporal split and use the first 70% of each time series for training, the next 10% for validation, and the last 20% for testing, as in related work (Wu et al. 2021; Zhou et al. 2022; Nie et al. 2022; Zeng et al. 2022).

The *Electricity* dataset[2] is published in Lai et al. (2018) and used in related work on long-term forecasting (Zhou et al. 2021; Wu et al. 2021; Zhou et al. 2022; Nie et al. 2022; Zeng et al. 2022). It is a subset of the UCI Electricity Load Diagrams dataset[3] first presented in Rodrigues and Trindade (2018), only containing the time series without missing values. The dataset contains hourly electrical load data from 321 clients of a Portuguese energy supplier. The clients are from different economic sectors, including offices, factories, supermarkets, hotels, restaurants, among others (Rodrigues and Trindade 2018). The time series range from 2012 to 2014.

The *Ausgrid solar home* dataset[4] contains solar generation and electrical load data from 300 clients[5] of an Australian energy supplier. The clients are private houses with rooftop solar systems. The time series range from July 2010 to June 2013. We only use the electrical load data transformed into hourly resolution.

[2] https://github.com/laiguokun/multivariate-time-series-data.

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

[4] https://www.ausgrid.com.au/Industry/Our-Research/Data-to-share/Solar-home-electricity-data.

[5] We use 299 of the clients because one client had missing data.

Comparison methods

We compare our models with models from related work (Zhou et al. 2021; Wu et al. 2021; Zhou et al. 2022; Nie et al. 2022; Zeng et al. 2022), as well as with a persistence baseline, linear regression models, multi-layer perceptrons and long short-term memory networks.

- *Models from related work:* For Informer (Zhou et al. 2021), Autoformer (Wu et al. 2021), FEDformer (Zhou et al. 2022), PatchTST (Nie et al. 2022) and LTSF-Linear (Zeng et al. 2022), we take the results reported in the publications where applicable, and run the code published with the papers otherwise. All parameters except for the forecast horizon are left unchanged.
- *Persistence baseline:* The persistence baseline takes the value from one week before the predicted hour as a forecast for the 24 h and 96 h horizons, and the value from 1 month before the predicted hour as the 720 h forecast.
- *Linear regression:* For each load time series, we train a linear regression model with h outputs. The input consists of the last 336 load values and the nine time and calendar features for the current hour when the prediction is made (see "Approach" for a description of the features). The main difference to LTSF-Linear (Zeng et al. 2022) is that the linear regression models are local models, but LTSF-Linear is a global model. Furthermore, the two approaches use different training algorithms and LTSF-Linear does not use time and calendar features.
- *Multi-layer perceptron (MLP):* As for the linear regression, we train a local MLP for each load time series. The MLPs get the last 168 load values and the nine time and calendar features of the current hour as input. Using more than 168 load values as input did not improve the results. Each MLP has two hidden layers with ReLU activation (ReLU 2023) and 1024 neurons per layer.
- *Long short-term memory (LSTM):* We train multivariate, local and global LSTM (Hochreiter and Schmidhuber 1997) models. We use the same architecture as in Kong et al. (2017), consisting of two LSTM layers with 20 units each and a linear prediction layer. Using larger models did not improve the results.

Training details

All models are trained with the AdamW optimizer (Loshchilov and Hutter 2019) using the mean squared error loss. We use a batch size of 128 and a learning rate of 0.0001 with 1000 warm-up steps and cosine decay with $\gamma = 0.8$. When testing different lookback sizes L, we find one week to be optimal for the multivariate Transformer and the local Transformers. For the global Transformer, the results improve with increasing lookback size until $L = 336$ (two weeks), and stay almost the same for $L = 720$ (one month). For Transformer models with two weeks input and one month output, the batch size has to be reduced to 64 due to the quadratic memory consumption of the model. For the multivariate Transformer, the batch size is set to 32 as in related work (Zhou et al. 2021; Wu et al. 2021; Zhou et al. 2022). The validation error is evaluated every 10,000 training steps and at the end of every epoch. We use early

Table 2 MAE results on the two datasets, with 24, 96 and 720 h forecast horizon

Model	Strat-	Input	Electricity			Ausgrid		
	egy	(days)	24h	96h	720h	24h	96h	720h
Informer (Zhou et al. 2021)	MV	4	0.399	0.407	0.450	0.582	0.607	0.645
Autoformer (Wu et al. 2021)	MV	4	0.289	0.317	0.361	0.579	0.569	0.592
FEDformer (Zhou et al. 2022)	MV	4	*0.284*	*0.297*	*0.343*	*0.560*	*0.566*	0.609
LSTM	MV	7	0.400	0.402	0.407	0.611	0.618	0.613
Transformer	MV	7	0.366	0.384	0.382	0.584	0.586	*0.576*
Persistence	L	–	0.279	0.279	0.447	0.647	0.647	0.717
Linear regression	L	14	0.203	*0.233*	*0.296*	*0.496*	*0.524*	*0.565*
MLP	L	7	*0.199*	0.236	0.308	0.499	0.532	0.567
LSTM	L	7	0.263	0.283	0.337	0.517	0.541	0.573
Transformer	L	7	0.256	0.289	0.354	0.535	0.563	0.583
LTSF-Linear (Zeng et al. 2022)	G	14	0.209	0.237	0.301	0.490	0.515	0.553
PatchTST (Nie et al. 2022)	G	14	0.190	**0.222**	**0.290**	**0.468**	**0.494**	**0.522**
LSTM	G	7	0.207	0.239	0.302	0.491	0.525	0.559
Transformer	G	14	**0.184**	0.225	0.312	0.482	0.514	0.533

MV = multivariate, L = local, G = global. The best results are highlighted in bold and the best results per training strategy are highlighted in italic

stopping to end the training when no more improvement on the validation set is seen for ten evaluations. For the MLPs, the initial learning rate is set to 0.001 and decayed with $\gamma = 0.5$ after every epoch.

Metric

As in related work (Zhou et al. 2021; Wu et al. 2021; Zhou et al. 2022; Nie et al. 2022; Zeng et al. 2022), every load time series is standardized by subtracting its mean and dividing by its standard deviation and the metrics are computed on these standardized time series. For every hour $t \in T_{\text{test}}$ in the test set, a forecasting model predicts the next h hourly loads $\hat{y}^c_t = \hat{y}^c_{t,t+1}, \ldots, \hat{y}^c_{t,t+h}$ for time series c. Then, the mean absolute error (MAE) between the predictions $\hat{y}^c = \{\hat{y}^c_i \; \forall \, i \in T_{\text{test}}\}$ and the ground truth $y^c = y^c_1, \ldots, y^c_{T_{\text{test}}}$ is computed. As the final result, the MAE averaged across all C load time series, the T_{test} evaluation time points and the h forecasting steps is reported.

$$\text{MAE}(y, \hat{y}) = \frac{1}{C \cdot |T_{\text{test}}| \cdot h} \sum_{c=1}^{C} \sum_{t \in T_{\text{test}}} \sum_{i=1}^{h} |y^c_{t+i} - \hat{y}^c_{t,t+i}|.$$

The mean squared error (MSE) is computed analogously, using the squared residuals instead of the absolute residuals.

Results

Forecast accuracy

Table 2 shows the MAE results on the two datasets[6]. On the Electricity dataset, the global Transformer is the best model for the 24 h horizon, and PatchTST is the best model for longer horizons. On the Ausgrid solar home dataset, PatchTST is the best model for all three horizons. The global Transformer beats the local Transformers and

[6] The MSE results show a similar pattern and can be found on GitHub.

Table 3 Training times in hours, measured on a machine with a Nvidia 3090 RTX GPU

Model	Electricity			Ausgrid		
	24h	96h	720h	24h	96h	720h
Linear regression (local)	0.02	0.03	0.08	0.02	0.03	0.07
MLP (local)	0.42	0.42	0.36	0.40	0.39	0.39
LSTM (multivariate)	0.06	0.08	0.30	0.03	0.04	0.10
LSTM (local)	8.25	7.61	7.20	4.27	3.55	3.49
LSTM (global)	1.09	0.82	0.98	1.11	0.84	0.71
Transformer (multivariate)	0.19	0.23	0.88	0.10	0.09	0.39
Transformer (local)	14.20	16.82	102.09	8.33	9.74	62.53
Transformer (global)	3.42	2.00	9.86	3.85	2.85	6.77

the multivariate Transformer across all tested horizons. On average, it reduces the error by 21.8% compared to the multivariate Transformer and by 12.8% compared to the local Transformers. Compared to the best local model, the linear regression, it reduces the error by 2.9%. Compared to the best multivariate model, FEDformer, it reduces the error by 15.4%. All multivariate models, including Informer, Autoformer, FEDformer and the multivariate Transformer, perform poorly and do not beat the persistence baseline with a lag of one week. The local linear regression models are slightly better than the global linear model, LTSF-Linear, on the Electricity dataset, but it is vice versa on the Ausgrid solar home dataset. The MLP is in five out of six cases a bit worse than the linear regression, with a 1.5% larger error on average. The local LSTMs are better than the local Transformers, but the Transformer is better as a multivariate model and as a global model (except for the one month horizon on the Electricity dataset). The forecast errors are lower on the Electricity dataset than on the Ausgrid dataset which is a more fine-grained dataset containing single private houses.

Computational cost

The training times are given in Table 3. The local Transformer models need by far the longest time to train. Their training time increases sharply with longer forecast horizons. The multivariate Transformer trains fast and is even faster than the MLPs for short horizons. Training a global Transformer is much faster than training the many local Transformers but takes longer than the linear regression, MLP and the multivariate Transformer. The LSTM always trains faster than the Transformer with the same training strategy.

Discussion

Best Transformer training strategy: On the two datasets, the global Transformer is superior to the multivariate and local Transformers. We hypothesize that this is a result of the larger number of training samples for the global model (see Table 1). The Transformer benefits from more training data, even if the training data comes from different sources. The multivariate models on the other hand are prone to overfitting.

Best Transformer architecture: PatchTST is the best model in five out of six cases. However, the difference to the global Transformer is small. This shows that the success

of PatchTST is mainly a result of its global training strategy. Its improvement upon the global Transformer can be due to the patching mechanism, a better hyperparameter configuration, or the encoder-only architecture. Among the multivariate models, Autoformer (Wu et al. 2021) and FEDformer (Zhou et al. 2022) give better results than the multivariate Transformer. It remains an open question whether these architectures are also better global models than the standard Transformer and PatchTST (Nie et al. 2022). Another promising architecture is the Temporal Fusion Transformer (Huy et al. 2022). In previous work with just one aggregated time series, the Informer (Zhou et al. 2021) also gave better results than the Transformer (Hertel et al. 2022a).

Comparison with the state of the art: The global Transformer achieves a better result for short-term forecasting on the Electricity dataset than related work (Zhou et al. 2021; Wu et al. 2021; Zhou et al. 2022; Nie et al. 2022; Zeng et al. 2022), and achieves close results to the best results from PatchTST (Nie et al. 2022) for longer horizons and on the Ausgrid solar home dataset. However, to establish a state of the art for short-term and medium-term load forecasting, a comparison to other forecasting models must be undertaken, including models that are not based on the Transformer architecture and that are more sophisticated than our baselines. Using weather data could improve the forecasts, because some electrical load patterns, such as the usage of electrical heating, are weather-dependent. Weather features could affect which model gives the best results, because some models might be better in capturing these dependencies than others.

Linear models: As in related work (Zeng et al. 2022), we observe that linear models are strong baselines. The linear regression is in five out of six cases the best local model and only outperformed by the local MLP for the one day horizon on the Electricity dataset. No general answer can be given on whether the local linear regression models are better or the global LTSF-Linear is better, because each variant is better on one dataset.

Task complexity: For longer horizons, the global Transformer's performance compared to the linear models deteriorates. This can be due to the increasing complexity when the model forecasts many values simultaneously. We chose a direct multi-step forecasting model because good results were achieved with this procedure before (Nie et al. 2022; Zeng et al. 2022). However, other multi-step forecasting procedures, such as iterative single-step and iterative multi-step forecasting (An and Anh 2015; Sahoo et al. 2020), could be beneficial for long-term forecasting because they reduce the number of forecasted values per model run.

Transfer learning: According to the definition of transfer learning in Pinto et al. (2022), the global training strategy can be seen as a transfer learning method, because the model must transfer knowledge between different types of buildings with different consumption patterns. Pre-training on other tasks than forecasting or on less similar data from domains other than electricity, as well as fine-tuning for a time series of interest, could improve the results. An advantage of the global model is that it can be applied to new time series without retraining. In Hertel et al. (2022b) it was shown that the Transformer generalizes better to new time series than other approaches, but the forecasts are still better when training data from the target time series is available.

Other forecasting tasks: The Transformer model and the different training strategies are not designed for load forecasting in particular, but can also be applied to other forecasting tasks. We hypothesize that the global training strategy can also be beneficial for other datasets containing multiple time series with similar patterns.

Conclusion and future work

We compare three Transformer training strategies for load forecasting on two datasets with multiple years of data for multiple hundred clients. We show that the multivariate training strategy used in related work on forecasting with Transformers (Zhou et al. 2021; Wu et al. 2021; Zhou et al. 2022) is not optimal, and it is better to use a global model instead. This shows that the right training strategy is crucial to get good results from a Transformer. Our approach achieves better results than related work (Zhou et al. 2021; Wu et al. 2021; Zhou et al. 2022), and comes close to the best results from PatchTST (Nie et al. 2022). In particular, our approach gives better results than the linear models from Zeng et al. (2022) for one day to four days forecasting horizons, which shows that, with the right training strategy, Transformers are effective for load forecasting. However, simple linear models give decent results for both short-term and medium-term horizons and train much faster than the Transformers.

In the future, more sophisticated Transformer architectures could be tested with the global training strategy. A comparison to other forecasting methods could be undertaken, and weather data could be incorporated into the models to see how it affects the results. Experiments with other datasets and varying amounts of training data could show under which circumstances the global Transformer model is better than other approaches. Additionally, transfer learning from other tasks and datasets could be tested. Future work could experiment with different datasets with varying amounts of data to see how much training data is needed for the global model to surpass the local models. A compromise between local and global models could be established by first clustering similar time series and then training one global model per cluster. The cluster-specific models would have less training data than the global model, but could benefit from the training data being more similar. Potentially, the global training strategy could also be beneficial for other forecasting tasks than load forecasting.

Acknowledgements
We thank the anonymous reviewers for their helpful comments.

About this supplement
This article has been published as part of Energy Informatics Volume 6 Supplement 1, 2023: Proceedings of the 12th DACH+ Conference on Energy Informatics 2023. The full contents of the supplement are available online at https://energyinformatics.springeropen.com/articles/supplements/volume-6-supplement-1.

Author contributions
MH: Conceptualisation, Investigation, Methodology, Software, Validation, Visualisation, Writing—original draft. MB: Writing—original draft, Writing—review and editing. BH, ON: Writing—review and editing. BS, RM, VH: Funding acquisition, Supervision, Writing—review and editing.

Funding
This project is funded by the Helmholtz Association's Initiative and Networking Fund through Helmholtz AI, the Helmholtz Association under the Program "Energy System Design", and the German Research Foundation (DFG) as part of the Research Training Group 2153 "Energy Status Data: Informatics Methods for its Collection, Analysis and Exploitation".

Availibility of data and materials
See section Datasets for the sources of the public datasets. Code is available on GitHub via https://github.com/KIT-IAI/transformer-training-strategies.

Declarations

Competing interests

The authors declare that they have no competing interests.

Published: 19 October 2023

References

A gentle introduction to the rectified linear unit (ReLU). https://machinelearningmastery.com/rectified-linear-activation-function-for-deep-learning-neural-networks/. Accessed 28 Apr 2023

An NH, Anh DT (2015) Comparison of strategies for multi-step-ahead prediction of time series using neural network. In: 2015 International Conference on Advanced Computing and Applications (ACOMP), pp. 142–149

Çakmak HK, Hagenmeyer V (2022) Using open data for modeling and simulation of the all electrical society in eASiMOV. In: 2022 Open Source Modelling and Simulation of Energy Systems (OSMSES)

Cao Y, Dang Z, Wu F, Xu X, Zhou F (2022) Probabilistic electricity demand forecasting with transformer-guided state space model. In: 2022 IEEE 5th International Conference on Automation, Electronics and Electrical Engineering (AUTEEE), pp. 964–969. IEEE

Gao J, Hu W, Zhang D, Chen Y (2022) TgDLF2.0: Theory-guided deep-learning for electrical load forecasting via transformer and transfer learning. arXiv:2210.02448

Giacomazzi E, Haag F, Hopf K (2023) Short-term electricity load forecasting using the temporal fusion transformer: Effect of grid hierarchies and data sources. arXiv preprint arXiv:2305.10559

Grabner M, Wang Y, Wen Q, Blažič B, Štruc V (2023) A global modeling framework for load forecasting in distribution networks. IEEE Trans Smart Grid (Early Access)

Haben S, Arora S, Giasemidis G, Voss M, Greetham DV (2021) Review of low voltage load forecasting: methods, applications, and recommendations. Appl Energy 304:117798

Han F, Pu T, Li M, Taylor G (2020) Short-term forecasting of individual residential load based on deep learning and k-means clustering. CSEE J Power Energy Syst 7(2):261–269

Hertel M, Ott S, Schäfer B, Mikut R, Hagenmeyer V, Neumann O (2022) Evaluation of transformer architectures for electrical load time-series forecasting. In: Proceedings 32. Workshop Computational Intelligence

Hertel M, Ott S, Schäfer B, Mikut R, Hagenmeyer V, Neumann O (2022) Transformer neural networks for building load forecasting. In: Tackling Climate Change with Machine Learning: Workshop at NeurIPS 2022

Himeur Y, Elnour M, Fadli F, Meskin N, Petri I, Rezgui Y, Bensaali F, Amira A (2022) Next-generation energy systems for sustainable smart cities: roles of transfer learning. Sustain Cities Soc 85:104059

Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9(8):1735–1780

Hong T, Pinson P, Wang Y, Weron R, Yang D, Zareipour H (2020) Energy forecasting: a review and outlook. IEEE Open Access J Power Energy 7:376–388

Huy PC, Minh NQ, Tien ND, Anh TTQ (2022) Short-term electricity load forecasting based on temporal fusion transformer model. IEEE Access 10:106296–106304

Kong W, Dong ZY, Jia Y, Hill DJ, Xu Y, Zhang Y (2017) Short-term residential load forecasting based on LSTM recurrent neural network. IEEE Trans Smart Grid 10(1):841–851

Lai G, Chang W-C, Yang Y, Liu H (2018) Modeling long-and short-term temporal patterns with deep neural networks. In: The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, pp. 95–104

Loshchilov I, Hutter F (2019) Decoupled weight decay regularization. In: 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6–9, 2019

Machowski J, Bialek J, Bumby JR, Bumby J (1997) Power system dynamics and stability. Wiley, USA

Murphy WMJ, Chen K (2023) Univariate vs multivariate time series forecasting with transformers. https://openreview.net/forum?id=GpW327gxLTF

Nawar M, Shomer M, Faddel S, Gong H (2023) Transfer learning in deep learning models for building load forecasting: Case of limited data. arXiv:2301.10663

Nie Y, Nguyen NH, Sinthong P, Kalagnanam J (2022) A time series is worth 64 words: long-term forecasting with transformers. arXiv:2211.14730

Ordiano JÁG, Waczowicz S, Hagenmeyer V, Mikut R (2018) Energy forecasting tools and services. WIREs Data Mining Knowl Discov 8(2)

Pinto G, Wang Z, Roy A, Hong T, Capozzoli A (2022) Transfer learning for smart buildings: a critical review of algorithms, applications, and future perspectives. Adv Appl Energy 100084

Pörtner H-O, Roberts DC, Adams H, Adler C, Aldunce P, Ali E, Begum RA, Betts R, Kerr RB, Biesbroek R et al (2022) Climate change 2022: impacts, adaptation and vulnerability. IPCC Geneva, Switzerland

Ramchurn SD, Vytelingum P, Rogers A, Jennings NR (2012) Putting the "smarts" into the smart grid: a grand challenge for artificial intelligence. Commun ACM 55(4):86–97

Ran P, Dong K, Liu X, Wang J (2023) Short-term load forecasting based on CEEMDAN and transformer. Electric Power Syst Res 214:108885

Rodrigues F, Trindade A (2018) Load forecasting through functional clustering and ensemble learning. Knowl Informat Syst 57(1):229–244

Sahoo D, Sood N, Rani U, Abraham G, Dutt V, Dileep A (2020) Comparative analysis of multi-step time-series forecasting for network load dataset. In: 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT), pp. 1–7

Shi H, Xu M, Li R (2017) Deep learning for household load forecasting—a novel pooling deep RNN. IEEE Trans Smart Grid 9(5):5271–5280

Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser L, Polosukhin I (2017) Attention is all you need. In: NIPS, pp. 5998–6008
vom Scheidt F, Medinová H, Ludwig N, Richter B, Staudt P, Weinhardt C (2020) Data analytics in the electricity sector—a quantitative and qualitative literature review. Energy AI 1:100009
Voß M, Bender-Saebelkampf C, Albayrak S (2018) Residential short-term load forecasting using convolutional neural networks. In: 2018 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm), pp. 1–6
Wang C, Wang Y, Ding Z, Zheng T, Hu J, Zhang K (2022) A transformer-based method of multienergy load forecasting in integrated energy system. IEEE Trans Smart Grid 13(4):2703–2714
Werling D, Heidrich B, Çakmak HK, Hagenmeyer V (2022) Towards line-restricted dispatchable feeders using probabilistic forecasts for PV-dominated low-voltage distribution grids. In: Proceedings of the Thirteenth ACM International Conference on Future Energy Systems, pp. 395–400
Wu N, Green B, Ben X, O'Banion S (2020) Deep transformer models for time series forecasting: The influenza prevalence case. arXiv preprint arXiv:2001.08317
Wu H, Xu J, Wang J, Long M (2021) Autoformer: decomposition transformers with auto-correlation for long-term series forecasting. In: NeurIPS, pp. 22419–22430
Yang E, Youn C-H (2021) Individual load forecasting for multi-customers with distribution-aware temporal pooling. In: IEEE INFOCOM 2021-IEEE Conference on Computer Communications, pp. 1–10
Yildiz B, Bilbao JI, Dore J, Sproul AB (2017) Recent advances in the analysis of residential electricity consumption and applications of smart meter data. Appl Energy 208:402–427
Zeng A, Chen M, Zhang L, Xu Q (2022) Are transformers effective for time series forecasting? arXiv:2205.13504
Zhang G, Wei C, Jing C, Wang Y (2022) Short-term electrical load forecasting based on time augmented transformer. Int J Comput Intell Syst 15(1):67
Zhou T, Ma Z, Wen Q, Wang X, Sun L, Jin R (2022) FEDformer: frequency enhanced decomposed transformer for long-term series forecasting. In: International Conference on Machine Learning, pp. 27268–27286
Zhou H, Zhang S, Peng J, Zhang S, Li J, Xiong H, Zhang W (2021) Informer: Beyond efficient transformer for long sequence time-series forecasting. In: AAAI, pp. 11106–11115

Publisher's Note