Benchmarking of state-of-the-art machine learning methods for highly accurate thermal load forecasting in district heating networks

Christian Pressa[a], Stefan Leiprecht[a], Fabian Behrens[a], Verena Jetzinger[a], Hendric Popma[a], and Matthias Finkenrath[a]

*[a] Kempten University of Applied Sciences, Germany,
e-mail: christian.pressa@hs-kempten.de*

Abstract:

Decarbonisation of heat generation has become a priority for district heating network operators. In order to avoid the use of fossil-fired boilers, operators need to know peaks in heat demand in advance. Accurate thermal load forecasting is playing an increasingly important role in this respect.

This paper presents the final results of the research project “deepDHC” (deep learning for district heating and cooling) funded by the German Federal Ministry for Economic Affairs and Climate Action (BMWK). The three-year project focused on systematically benchmarking thermal load forecasts for district heating networks, based on state-of-the-art machine learning methods. The analysis covers a variety of machine learning techniques, such as neural networks – including latest deep learning methods – (e.g. LSTM, TFT, ESN, RC), decision trees (random forests, adaptive boosting, XGB) and statistical methods (SARIMAX). In addition, the impact of combining methods by so-called “stacking” was investigated. Training and validation of the machine learning algorithms was based on historical operating data from the district heating network for the city of Ulm in Germany, in combination with historical weather data, and weather forecasts. Thermal load forecasts – typically for three days ahead – are presented and compared against one another. An automatic tuning routine was developed as part of the project, which enables regular re-training of the machine learning algorithms based on the latest operating data from the heating network. Furthermore, a web interface for real-time forecasting was developed and implemented at the power station.

Keywords:

District heating; load forecasting; machine learning; dispatch optimisation.

1. Introduction

Half of Europe’s entire energy consumption is used for heating and cooling, with 75 percent derived from fossil sources. Thus it is important to provide heat as efficiently as possible, also producing the lowest possible greenhouse gas emissions. District heating networks play a key role in this context, as they are currently used to supply 60 million Europeans with heat.[1] However, with total lengths often spanning several hundred kilometres, supplying district heating networks efficiently presents a considerable challenge. Typically, heating network operators maintain several power plants with different technical and economic features. District heating or cooling can be provided more cost-effectively, efficiently and with fewer emissions, the more precisely the expected load can be estimated. In addition to a comprehensive understanding of individual power plants and a dispatch optimisation strategy, this requires a precise forecast of the thermal loads that can be expected in the network during the next few days. In industrial practice, however, generally only simple forecasting methods with comparably high uncertainties are used for district heating load forecasts

Previously established methods, which are based on typical days or reference load profiles, [2] currently lead to an estimated 15-30 percent load forecasting error rate over a 72-hour horizon. In addition, heating load forecasts have been considerably less intensively studied to date than, for example, electricity load forecasts.

Accurately predicting the district heating load is essential for district heating network operators in order to optimise the utilisation of available power plants, thermal storage facilities and “power-to-heat” plants, thus boosting efficiency. The quality of district heating load forecasting directly impacts plant dispatch quality, making it a critical factor to consider. Nevertheless, quantifying the economic advantage of reducing load forecasting errors is difficult, due to the fact that different load forecasts cannot be relied on to accurately repeat the actual plant dispatch during a specific period. The economic benefit of improved thermal load forecasting was shown in in a previous publication.[3]

New methods in the field of artificial intelligence, especially in machine learning and "deep learning", offer considerable potential for improvement, in particular since machine learning techniques have become easy to deploy due to increasing amounts of training data coupled with cheaper and improved computing power.[4–7]

https://doi.org/10.52202/069564-0283

This work is based on operating data from the district heating network for Ulm, a medium-sized city in southern Germany with about 125,000 inhabitants, as provided by the local utility company “Fernwärme Ulm GmbH” (FUG), which operates a district heating network about 150 km in length supplying some 600,000 MWh of thermal energy per year, equivalent to 45 percent of the city’s heat requirements. Table 1 lists all the major heat generation units operated by FUG. In addition, a pressurised hot water thermal storage of 2,427 m^3 volume, equivalent to 150 MWh thermal energy storage capacity, and with 20 MW discharge power, is used.

Table 1. Heat generation units operated by FUG.

Plant type	Thermal load, MW	Electrical load, MW	Fuel type
Biomass CHP plant 1	58	9	Waste wood
Biomass CHP plant 2	25	5	Waste wood
Waste incineration CHP plant	30	11	Waste
Peaking boiler (w/ connection to turbine)	230	15	Coal, gas or oil

2. DeepDHC system structure

The DeepDHC system structure in Figure 1 below shows the developed process for load forecasting. The basis for training the models is historical data on the heat load of the district heating networks, as well as historical and current weather forecasts, which are obtained from the German Meteorological Service (Deutscher Wetterdienst, DWD) and stored in a database. In addition, the weather data currently measured for the Ulm site are obtained from the DWD’s Climate Data Center (CDC) and saved together with the actual district heating load in an hourly resolution. Operational data is rarely available in the correct format or ideal quality. Deviations such as missing measured values, outliers, or data with a different interval (time frames) must be corrected or optimised with statistical methods as part of a pre-processing step. During the machine learning process, features are generated, datasets are divided into test and training data, and the corresponding models are initially trained and optimised. Based on the trained model, a live forecast of the heat load is generated and made available to the power plant operator via a web interface. In addition, an automated learning routine was developed in the project. This routine helps to always provide the best model with optimal parameters, including retraining the forecasting models whenever new training data become available.

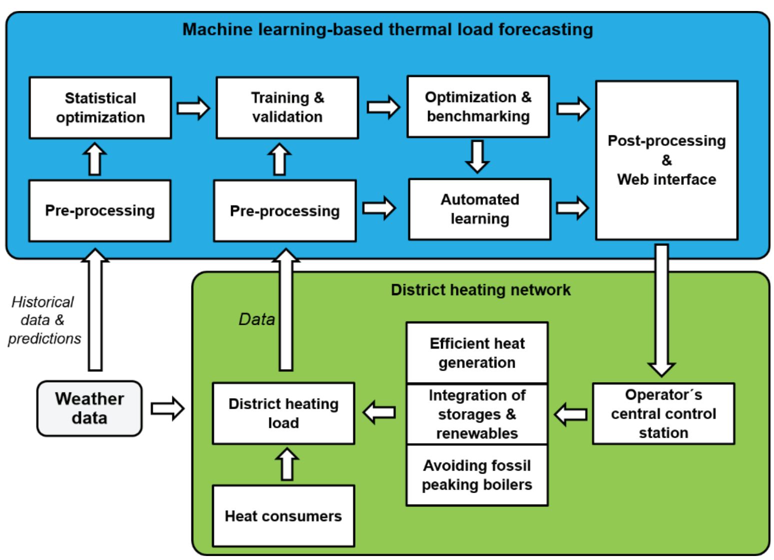

Figure 1. The deepDHC machine learning process for thermal load forecasting

3. Analysed machine learning methods

The objective of this chapter is to provide a concise overview of the analysed machine learning methods, including their characteristics, and to specify the software packages that were used for their implementation.

3.1. Seasonal Auto-Regressive Integrated Moving Average (SARIMAX)

In contrast to neural networks or decision trees, SARIMAX (“Seasonal Auto-Regressive Integrated Moving Average)” is based on statistical calculation models. SARIMAX extends the ARIMA model by adding a

https://doi.org/10.52202/069564-0283

seasonal counterpart to each component.[8] Compared to ARIMA, SARIMIX offers a “seasonal order”, which offers advantages for the type of forecasts examined. There are terms that capture the pattern of the data over a seasonal period, such as a week, a month, or a year. SARIMAX was implemented in this study using the “statsmodels” python library.[9]

3.2. Echo-State Networks (ESN)

Echo-State Networks (ESN) are one of the simplest implementations of a neuron-based reservoir computer. Reservoir computing is a machine learning approach to training recurrent neural networks (RNN). At its core, RNN cells contain an internal memory state which acts as a compact summary of past information. A unique characteristic of the ESN model is its efficiency in terms of the required computing power.[10, 11] Modern machine learning methods are becoming increasingly complex and require increasingly powerful hardware. In contrast, the development of reservoir computers focuses on reducing complexity and solving large machine learning tasks with minimal computing power. Recently, reservoir computers have proven to be a powerful and resource-efficient alternative in the field of classic forecasting of complex dynamic systems.[12] The Python package ReservoirPy was used in implementing this project.[11]

3.3. Next Generation Reservoir Computing (NGRC)

Like the “classic” ESN model, the NGRC model belongs to the group of neuron-based reservoir computers.[12] [13] In the background, a nonlinear vector autoregression (NVAR) algorithm automatically delivers the best possible classic ESN model with the associated hyperparameters. Like the classic ESN model, the NGRC model is very efficient and very easy for the programmer to implement. The Python package ReservoirPy was used in implementing this project.[11]

3.4. Temporal Fusion Transformer (TFT)

In the area of machine learning, transformers are a subtype of neural network. A common obstacle for machine learning algorithms is the inability to handle data in the dimension of time. Commonly, the algorithms weight the relevance of data at each location equally, thus preventing the extraction of usable knowledge in the time dimension. The architecture of the Temporal Fusion Transformer is specialised for performing such time-series prediction tasks. To learn temporal relationships on different scales, TFT uses recurrent layers for local processing and interpretable self-attention layers for long-term dependencies.[14] TFT offers powerful forecasts with customisation options, and unlike other machine learning methods it is not a “black box”. TFT is able to independently recognise dependencies within a data set and also includes tools for selecting relevant or irrelevant features. Originally developed by Google, several TFT implementations are now available. Further details of the TFT-related work in this study have been published separately.[14] The TFT used in this project was implemented using the PyTorch API.[15]

3.5. Long Short-Term Memory (LSTM)

LSTM is a gradient-based, recurrent neural network with feedback connections. LSTM neurons can hold values over any length of time, making them an attractive option for time-series prediction.[16] The standard recurrent neural network (RNN) architecture has difficulties in handling long-term dependencies, more precisely with the ability to adapt to recalling information late in the sequence. LSTM models address this limitation by introducing a memory cell, an internal state that can be updated and read by the network, and "gates" that control the flow of information in and out of the cell. This structure allows the LSTM to selectively store and update information across many time steps, making it well suited for time series forecasting.[16] The LSTM model was implemented using Keras LSTM layers.[17]

3.6. Adaptive Boosting (Adaboost)

Adaptive boosting, also known as AdaBoost, is a machine learning algorithm that belongs to the family of ensemble learning methods. Ensemble methods combine multiple models to improve their predictive power. The basic idea behind AdaBoost is to combine several “weak” learning models into a single “strong” model by iteratively learning from the mistakes of the weak learners. The algorithm can handle non-linear relationships between the input features and the target variable. It is also able to cope with missing data and outliers.[18] AdaBoost was implemented using the scikit-learn library.[19]

3.7. Extreme Gradient Boosting (XGBoost)

XGBoost is an open-source implementation of the decision tree gradient-boosting algorithm. The basic idea of gradient boosting is to train a series of weak models, where each model is trained to correct the previous model’s errors. By iteratively fitting the trees, XGBoost is able to build a highly accurate ensemble of decision trees that can capture complex nonlinear patterns in the data.[20] This ensemble of decision trees is based on gradient tree boosting, i.e. the trees grow sequentially with the knowledge gained from their predecessor. But this method is prone to overfitting, which XGBoost reduces by applying regularisation objects, shrinkage, and feature subsampling. XGBoost was implemented using the open-source python library XGBoost.[21]

3.8. Stacking

With the stacking method, the strengths of multiple individual models are harnessed by combining them into a single meta-model.[22] In this project, different combinations of base forecasting models such as Random Forest, ESN, TFT and LSTM models were analysed. The individual base models' predictions are fed as features into a meta-model, which then makes the final prediction.[22] This can be done in different ways. The stacking methods examined in this project were k-fold variation, bagging and averaging. All three methods were implemented via the scikit-learn library.[19]

4. Data characteristics and their effects on forecasting performance

When assessing a model's ability to make predictions, it is not only the quality of the training data that must be considered, but also its relevance to the current situation. In the context of district heating supply, certain factors influence demand independently of the events that can be planned and modelled. The outbreak of the Covid-19 pandemic was a major unforeseeable event that impacted energy needs, shifting the demand from industrial areas and educational institutions to residential areas. This presents a challenge for cities that operate separate networks for different urban areas, since previous regularities in the data are no longer valid. Additionally, it is uncertain whether the end of an event will restore the previous conditions, or whether changes in energy requirements caused by increased working from home during lockdowns will persist. Old training data may no longer be suitable for training models based on weight adjustments when networks expand or new urban areas are connected, leading to an increase in demand. Seasonal effects, along with the energy consumption of industrial plants that require hot water for their processes, must also be taken into account.

To account for these changes in the model, data must be collected for the altered consumption profile and used to train a new model. In addition to sudden changes such as grid expansion, primary energy shortages, and shifts in user behaviour, long-term global transformations such as climate change must also be considered. These changes occur continuously and can be compensated for with appropriate methods, as described in section 6.

In summary, numerous factors strongly influence the demand presented by a supply network. This project only utilised quantifiable influences to train the models. While it would be possible to incorporate additional data e.g. from climate models, economic forecasts, political events, crisis models, stock prices, and other forms of news into a model, it would be a time-consuming process that would not guarantee long-term improvement. Nevertheless, these factors should be taken into consideration when evaluating and using a forecasting technique. While machine learning models can recognise patterns in data that humans cannot see, it is important to note that there are no conscious trains of thought behind these mathematical models and that only information contained in the machine learning features used can be taken into account.

5. Benchmarking results

Fifteen years of operating and weather data in hourly resolution were used to train the machine learning algorithms. All eight subnetworks in the city of Ulm were analysed, each providing different challenges in terms of demand profiles, data quality and consumer structure. The data used in the following analysis came from a subnetwork of Fernwärme Ulm GmbH spanning a total length of 40 km and representing an annual heat demand of 75 GWh or 1,100 households. Hence a total of 131,400 data sets were used, each consisting of twenty parameters, i.e. measurement data from the district heating network and weather information such as air temperature, wind direction and wind speed. The correlation between individual parameters and the associated district heating load was previously checked by means of a correlation analysis.[23]

When optimising and comparing the quality of forecasting algorithms, the characteristics of the metrics used must be taken into account. In this project, mainly the mean absolute percentage error (MAPE) was investigated because it is easy to interpret and dimensionless. However, a weakness of the MAPE becomes apparent when the forecast values are very low, as during the summer months. The metric is skewed at values close to zero.[24] Therefore, the mean average error (MAE) is used in addition. In the following analysis, the forecasting methods for the heating period winter 2021 (December 2020 and January and February 2021) are examined and compared using the MAPE. This period is also the economically most interesting part of the year for heating network operators. If precise forecasts make it possible to avoid operating fossil-fuelled peaking boilers, the largest proportion of fossil fuels and thus CO_2 emissions can be saved during this period. The bar chart in Figure 2 below shows a final comparison of the forecasting errors of all analysed machine learning algorithms.

https://doi.org/10.52202/069564-0283

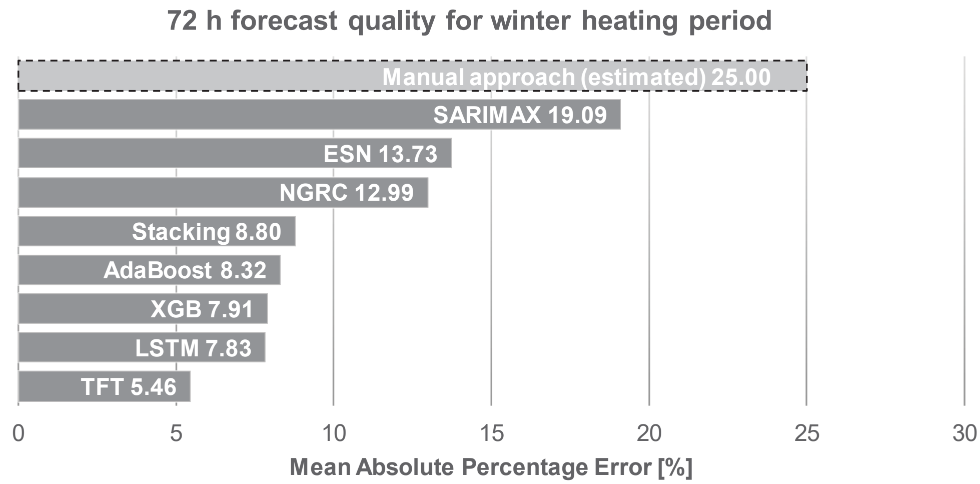

Figure 2. Final results of the 72 h forecast machine learning benchmarking

In the graph above, an estimated error of traditional forecasting methods where no machine learning algorithms is indicated for comparison. Although the SARIMAX model performs only slightly better than the estimated manual approach, it has the advantage of requiring only a small amount of data and still being able to produce a relatively good forecast (3.1 %-8.25 %) for a short time horizon of 12-24 hours (not shown in the graph). However, when predicting over a longer time horizon of 72 hours, the SARIMAX model performs significantly worse, with a MAPE of 19.09%. The NGRC and ESN recurrent neural networks show a similar performance, with NGRC slightly outperforming ESN with a MAPE of 12.99%. Similarly to SARIMAX, these algorithms also have other advantages besides a competitive forecasting error, such as the lower calculation power required and faster training time, which is briefly discussed at the end of this chapter. Throughout the project, it was found that stacking does not necessarily lead to better forecasts than the best individual method. Among the various stacking methods that were examined, such as k-fold variation, bagging, and averaging, simple averaging led to the lowest error value of 8.8%. However, stacking was found to be computationally expensive and required careful tuning of hyperparameters in order to avoid overfitting. Of the examined learning methods, AdaBoost, XGB, and LSTM are amongst the best and most stable, with a forecasting error of around eight percent. The LSTM model, in particular, achieves excellent forecasting quality and has lower memory requirements compared to XGB and AdaBoost, making it one of the most attractive models, but it requires significant implementation effort due to the many hyperparameter options and complex data pre-processing. However, the LSTM model demonstrated excellent adaptability when it was faced with new data. For example in scenarios that were previously unknown to the model, such as the Covid-19 pandemic or the increase in gas prices in 2022, which caused a reduction in heat demand, the LSTM model showed consistent results.

Overall, TFT delivers the best result for the examined subnetwork, with approximately 5.5% MAPE. Figure 3 illustrates the forecasting quality by showing the actual thermal load, the load prediction of the best TFT algorithm, and the percentage deviation for this method over a 72-hour period.

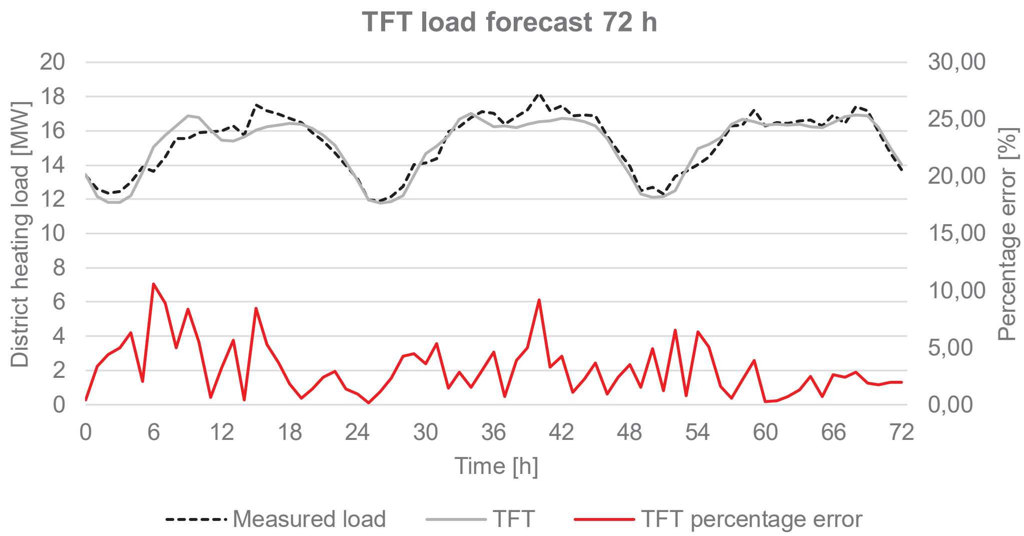

Figure 3. Example of a 72h thermal load forecast using the TFT model.

During live operation at a different grid operator, similar successful forecasting results have been observed. TFT is also one of the most complex algorithms to implement. But on the other hand, TFT offers the ability to predict multiple time series with a single model. If a district heating supplier operates several networks, which is usually the case, a separate model would have to be trained for each network when using a classic machine learning model. With the TFT model, this would no longer be necessary, since the model can assign features to different target predictions by classifying features into groups.[25] As a result, it is only necessary to train a single model for several use cases, which ideally saves time and money.

Benchmarking in terms of calculation power and training time is not straightforward as it depends heavily on the complexity of the model and various other factors such as the size and quality of the data and the hardware used for computation. However, the ESN and NGRC models proved to be exceptionally efficient, even if they do not provide the best forecasting quality. A monthly forecast with approx. 60,000 training data sets can be carried out using a simple office notebook (Intel Core i7-6600U, 16GB RAM, Windows 10) in a period of a few seconds. In comparison, the TFT model computed on a server with comparatively powerful hardware (NVIDIA Quadro RTX 6000 graphics card) needs about 30 minutes for the same training and forecasting process. In addition, the ESN model is very easy to implement and is one of the most user-friendly algorithms analysed.

6. Automated optimisation routine

During the initial setup of each model, various parameters need to be determined such as the choice of subnetwork, training timeframe, and the suitable dataset features. Adjusting the model-specific hyperparameters is usually part of "fine tuning" process, which enables the model to be adapted to the specific network in question. Each district heating network presents its own issues and characteristic properties, such as consumer structure, data quality and consumption patterns (households, industrial customers). A model that delivered very good results for one network might possibly deliver poor to unusable results for a network unknown to the model. Hence a network-specific approach with an automated optimisation routine was developed.

The developed routine is based on a random search algorithm.[26] The best model up to that point is loaded from the database and used as a reference. With the random search method, an n-dimensional parameter space is searched intelligently. By considering the effects of individual changes, parameters with a high impact are examined more intensively. This results in greater efficiency in contrast to a structured grid search. After a comparison, the best model is saved in the database and the process repeated. The system can use this routine to react to changes (network expansions, intensive savings measures, lengthy sub-network failures), in a limited scope, within a few intervals.

https://doi.org/10.52202/069564-0283

6. Live operation and web interface

In addition to benchmarking thermal load forecasting itself, a software system and web interface for live forecasting at the operator's site was developed. This covered a process of data aggregation, including a fully automated routine for regularly fetching required data from different sources, and combining them in a data warehouse. It included a step of screening and cleaning invalid data using methods such as imputation or other replacement strategies. Besides robust backend systems aiding the training and usage of machine learning models, a means of presenting the predictions and additional information to the operator of the district heating network is required. Thus a user-friendly web interface, capable of displaying 72-hour time frames of predictions together with corresponding data for any given date, was developed in cooperation with the district heating system operator. A screenshot of the thermal load forecasting web interface, which was installed at the operator's facilities in summer 2021 and has been in operation since then, is shown in Figure 4 below.

Figure 4. Web-based user interface developed for live operation of load forecasting.

7. DeepDHC user guide

A short guide was created to show the basic steps necessary to implement any of the described models. It is accessible via the code hosting platform GitHub under the following link:

https://github.com/deepDHC/deepDHC-user-guide

This guide will go through an example scenario that will use the historic thermal load and weather data to train a LSTM model. This model will then be saved and loaded from disc to do load predictions for another timeframe. Moreover, it shows how to calculate errors of the predictions compared to the real load demand, thereby providing a brief overview about all basic components needed to create models for district heating demand predictions.

8. Conclusion

The research project deepDHC systematically benchmarked thermal load forecasts using state-of-the-art machine learning methods, such as neural networks, decision trees and statistical methods. The study utilised long-term historical operating data from the district heating network in Ulm, Germany, combined with historical weather data and weather forecasts. Thermal load forecasts were predicted for three days ahead, in comparison against one another, and an automatic tuning routine was developed to retrain machine learning algorithms based on the latest operating data.

A key takeaway from the analysis is that a proper database has at least the same influence on forecasting quality as the model selection. Each and every district heating network is different and requires a model that is tailored to the specific network and database (see "No Free Lunch" theorem [27]).

Over a time period of 72 hours in advance, the developed forecasting tools were able to predict the thermal load within a mean average percentage error of five to eight percent for the investigated district heating subgrid. In summary, the TFT model performs best in the examined subnetwork with respect to forecasting accuracy. An attractive option is also the LSTM model, since it turned out to be computationally more efficient compared to the TFT model, whilst still ranking amongst the best and most stable algorithms in this study.

As part of a plant dispatch optimisation process, the forecasting methods that were analysed in this study can help to operate district heating networks more efficiently and cost-effectively.[3] By developing a web interface, the best analysed forecasting models could be made available for use in live operation and optimisation by the district heating system operator.

Acknowledgments

This work was funded by the German Federal Ministry for Economic Affairs and Climate Action under the funding code 03EN3017, with additional financial and in-kind support from the project partners AGFW, Fernwärme Ulm GmbH and ZAK Energie GmbH. The responsibility for the content of this publication lies with the authors. The authors would like to thank Till Faber and Jonas Brantl for their support during the study.

References

1. Corscadden J., Möhring P, Krasatsenka A. Renewable Energy Sources in District heating and Cooling: EU Level Survey. 2021.
2. VDI 4655: Reference load profiles of residential buildings for power, heat and domestic hot water as well as reference generation profiles for photovoltaic plants 2021.
3. Finkenrath M, Faber T, Behrens F, Leiprecht S. Holistic modelling and optimisation of thermal load forecasting, heat generation and plant dispatch for a district heating network. Energy 2021. doi:10.1016/j.energy.2022.123666.
4. Geysen, D., De Somer, O., Johansson, C., Brage, J., Vanhoudt, D. Operational thermal load forecasting in district heating networks using machine learning and expert advice. Energy and Buildings;2018:144–53.
5. Huang, S., Tang, W., Wu, Q., Li, C. Network constrained economic dispatch of integrated heat and electricity systems through mixed integer conic programming. Energy;2019:464–74.
6. Idowu, S., Saguna, S., Ahlund, C., Schelen, O. Applied machine learning: Forecasting heat load in district heating systems. Energy and Buildings;2016:478–88.
7. Xue, P., Jiang, Y., Zhou, Z., Chen, X., Fang, X., Liu, J. Multi-step ahead forecasting of heat load in district heating systems using machine learning algorithms. Energy. 2019.
8. Fang T, Lahdelma R. Evaluation of a multiple linear regression model and SARIMA model in forecasting heat demand for district heating system. Applied Energy. 2016:544–52.
9. Seabold, Skipper, Perktold J. Statsmodels: Econometric and Statistical Modeling with Python. Proceedings of the 9th Python in Science Conference. 2010.
10. Cortes C., Lawrence N, Lee D., Sugiyama M., Garnett R. Advances in neural information processing systems: Hidden Technical Debt in Machine Learning Systems. NIPS Proceedings. 2015.
11. Trouvain N., Pedrelli L. ReservoirPy. An Efficient and User-Friendly Library to Design Echo State Networks. 2020. https://hal.inria.fr/hal-02595026.
12. Jaurigue L, Lüdge K. Connecting reservoir computing with statistical forecasting and deep neural networks. Nature Communications. 2022.
13. Bollt E. On explaining the surprising success of reservoir computing; 2020.
14. Behrens F, Leiprecht S, Brantl J, Finkenrath M, Behrens F, Leiprecht S, et al. Temporal Fusion Transformer for thermal load prediction in district heating and cooling networks. SIMS Proceedings;2022:333–8. doi:10.3384/ecp192047.
15. Paszke A. et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library. Advances in Neural Information Processing Systems 32. 2019:8024–35.
16. Hochreiter S., Schmidhuber J. Long Short-Term Memory. Neural Computation;1997:1735–80.
17. Chollet F. et al. Keras. Journal of Data Analysis and Information Processing. 2016.
18. Raschka S, Mirjalili V. Machine Learning mit Python und Keras, TensorFlow 2 und Scikit-learn: Das umfassende Praxis-Handbuch für Data Science, Deep Learning und Predictive Analytics. 2021st ed. Frechen: MITP; 2021.
19. Pedregosa et al. Scikit-learn: Machine Learning in Python. JMLR. 2011:2825–30.
20. Chen T, Guestrin C. XGBoost: A Scalable Tree Boosting System. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2016:785–94. doi:10.1145/2939672.2939785.
21. xgboost developers. XGBoost Documentation. 2022. xgboost.ai.

 https://doi.org/10.52202/069564-0283

22. Wolpert D. Stacked Generalization. Neural Networks. 1992:241–59. doi:10.1016/S0893-6080(05)80023-1.
23. Faber T, Groß J, Finkenrath M. Innovative Lastprognosen mit Deep Learning-Methoden,. EuroHeat & Power;2018:1–2.
24. Barrot C. Methodik der empirischen Forschung. Prognosegütemaße. Wiesbaden: Gabler Verlag; 2007.
25. Lim B., Arik S., Loeff N., Pfister T. Temporal Fusion Transformers for interpretable multi-horizon time series forecasting. International Journal of Forecasting. 2021:1748–64.
26. Mantovani G: et al. Effectiveness of random search in SVM hyper-parameter tuning. International Joint Conference on Neural Networks (IJCNN). 2015:5–8.
27. Wolpert DH. Ubiquity symposium: Evolutionary computation and the processes of life: what the no free lunch theorems really mean: how to improve search algorithms. Ubiquity. 2013;2013:1–15. doi:10.1145/2555235.2555237.