

17th International Symposium on District Heating and Cooling, DHC2021, 6–9 September 2021, Nottingham, UK

A comprehensive thermal load forecasting analysis based on machine learning algorithms

Stefan Leiprecht, Fabian Behrens, Till Faber, Matthias Finkenrath*

Kempen University of Applied Sciences, Germany

Received 6 August 2021; accepted 22 August 2021

Abstract

Precise forecasting of thermal loads is a critical factor for economic and efficient operation of district heating and cooling networks. If thermal loads are known with high accuracy in advance, use of renewable energies can be maximized, and – in combination with thermal storage units – fossil generation, in particular in peaking units, can be avoided. Machine learning has proven to be a powerful tool for time series forecasting, and has demonstrated significant advancements in recent years. This paper presents the scientific methodology and first results of the publicly funded research project “deepDHC”, which aims at a broad benchmarking of traditional and advanced machine learning methods for thermal load forecasting in district heating and cooling applications. The analysis covers autoregressive forecasting approaches, decision trees such as “adaptive boosting”, but also latest “deep learning” techniques such as the “long short-term memory” (LSTM) neural network. This work is based on data from the district heating network of the city of Ulm in Germany. First, different performance metrics for evaluating forecasting qualities are introduced. Second, approaches for data screening and results of a linear and non-linear correlation analysis are presented. Third, the machine learning tuning process is described. For thermal load forecasting, weather data are key input parameters. This work uses hourly weather forecasts from weather models provided by the German meteorological service. These weather data are updated automatically, and have been statistically corrected in order to represent very accurate point forecasts for up to ten days ahead. In addition, a user-friendly web interface has been developed for use by the district heating network operator. The performance of different machine-learning algorithms is compared based on 72 h heating load forecasts.

© 2021 The Authors. Published by Elsevier Ltd. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

Peer-review under responsibility of the scientific committee of the 17th International Symposium on District Heating and Cooling, DHC2021, 2021.

Keywords: District heating; Load forecasting; Machine learning; Decision trees; Neural networks; LSTM

1. Introduction

Precisely forecasting thermal loads is crucial for operating district heating networks efficiently, economically and in an environmentally friendly way. If precise load forecasts are available to the operator, the use of fossil-fueled

* Corresponding author.

E-mail address: matthias.finkenrath@hs-kempen.de (M. Finkenrath).

<https://doi.org/10.1016/j.egy.2021.08.140>

2352-4847/© 2021 The Authors. Published by Elsevier Ltd. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

Peer-review under responsibility of the scientific committee of the 17th International Symposium on District Heating and Cooling, DHC2021, 2021.

peaking boilers can be significantly reduced. In addition, the integration of fluctuating renewables into the grid can be maximized. A precise long-term load forecast several days ahead also simplifies ordering fuel, or planned maintenance. Hence this work focuses on thermal load forecasts throughout 72 h in advance, based on hourly data from the district heating network for Ulm, a medium-sized city in southern Germany with about 125,000 inhabitants.

1.1. State of the art of thermal load forecasting for district heating networks

Accurately predicting heat loads has become an interesting field of application for modern time-series forecasting methods. Its importance even increases with rising global energy demand, decreasing reserves of fossil fuels and the impact of using fossil fuels on climate change [1]. District heating and cooling can be an efficient way of reducing carbon dioxide emissions by optimizing fuel consumption [2]. Machine learning has also proven to be an attractive option for generating accurate thermal load predictions in the context of district heating and cooling (e.g. [3]). Different algorithms have been evaluated in recent years for this purpose, such as Adaptive Boosting (AdaBoost) [4], Long Short-Term Memory (LSTM) recurrent neural networks [5], the Seasonal Autoregressive Integrated Moving Average Exogenous model (SARIMA) [6], Echo State Networks (ESN) [7] and Trajectory Tracking [8]. Most of the work previously conducted on this topic focused on a 24-h time horizon [1,9]. This project instead focuses on an extended forecasting period of up to 72 h to further optimize dispatch planning for power plants and thermal energy storage units. After analyzing single machine learning algorithms in the first phase of the project, stacked generalization is addressed thereafter to further improve forecasting precision. This involves combining an ensemble of algorithms in order to minimize individual error rates [10,11].

1.2. Project structure

Fig. 1 illustrates the basic structure and key activities of the “deepDHC” project. Weather and operating data, including smart metering data from the grid in the future, form the basis for analyzing a diverse set of machine learning-based thermal load forecasting methods. Important elements of the process are data pre-processing and statistical optimization, e.g., by imputation of missing data to increase the quality of the training data. After training and validation, the machine learning methods are optimized by tuning key parameters. Different forecasting methods are then benchmarked against one another. The best model for each method is used to make predictions for the next 72 h. In operation those predictions are updated hourly. The best predictions are then either selected based on their benchmarking performance for the current season and network section or all predictions are stacked. The processed load forecasts are made available to the operator via a web-based user interface. The experiences gained during the first phase of the project are used to develop automated machine learning. This approach would enable automatically generated load forecasts for new or changed heating and cooling networks, with minimal user input. All tools and forecasts are implemented and tested directly in the operator’s central control station in Ulm.

1.3. Characterization of analyzed district heating networks

The district heating network used for the evaluation has a total length of 40 km. It provides space heating for over 1100 households with an average annual heating demand of 75 GWh, and a heat load ranging from 2 to 22 MW. Water with a temperature between 70 °C and 100 °C is used as a heat transfer fluid. Fig. 2 shows significantly different typical heat demands in winter, summer and transition times. Therefore, in addition to a full-year analysis, the performance of the algorithms for each of these periods is evaluated. This information can be used beneficially in the second phase of the project, when stacked generalization is analyzed, to identify the best-suited combinations of forecasting models.

2. Methodology

2.1. Analyzed machine learning algorithms

Three different machine learning algorithms were analyzed, selected from three different machine learning categories, namely autoregressive models (SARIMAX), meta-algorithms (AdaBoost), and neural networks (LSTM):

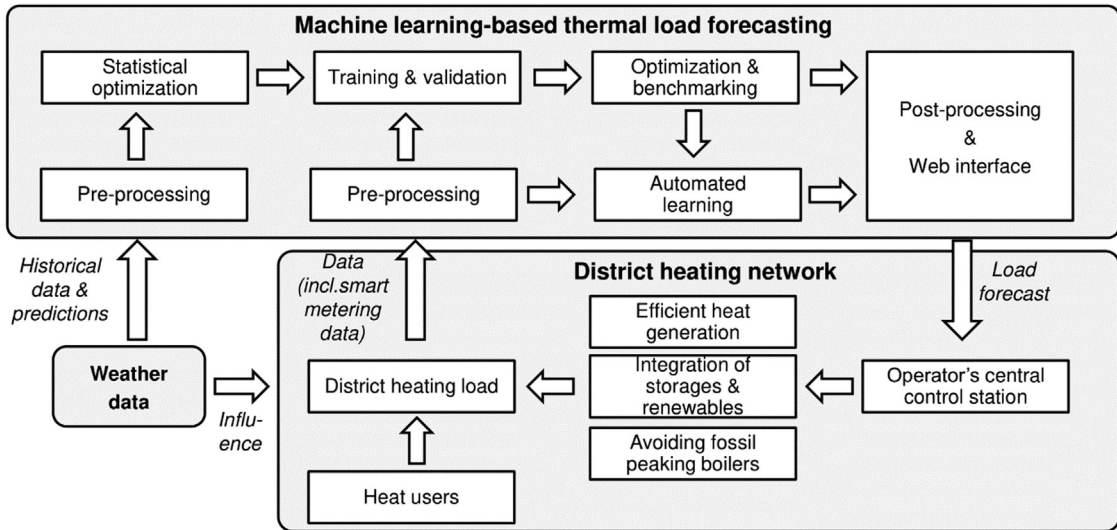


Fig. 1. Basic project structure and key activities.

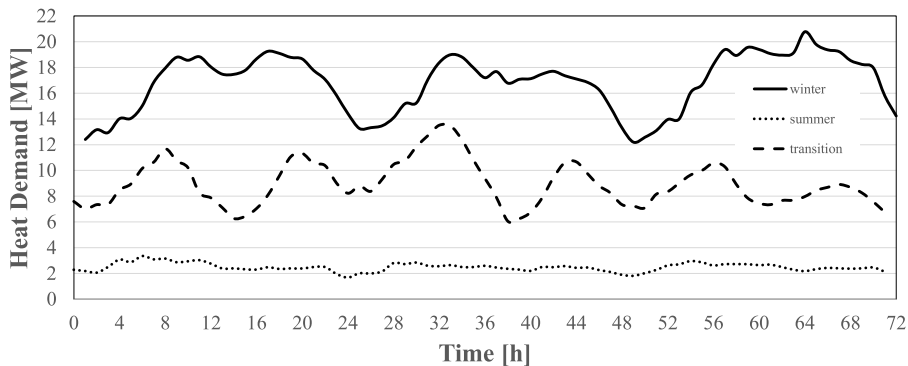


Fig. 2. Typical heat demand profiles during summer, winter and transition times.

- SARIMAX stands for seasonal auto regressive integrated moving average with exogenous factors. This stochastic process for modeling time series dating from 1970 is based on a Box–Jenkins approach that originates from economics [12]. To develop this model the *statsmodels* package was used [13].
- AdaBoost is a popular boosting strategy [4]. In each iteration, the individual training vectors are assigned a selection probability. This indicates how likely a vector is to be drawn in the training of a newly added learner. The data series that were underfitted by the previously executed learner receive a higher probability. Thus, the algorithm adapts to the regression errors in each step. As the weak learner Random Forest was used. Both, the Adaboost and the underlying Random Forest use the *Scikit-learn* library [14].
- LSTM is a gradient-based recurrent neural network with feedback connections. LSTM cells can remember values over an arbitrary amount of time, which make them an attractive option for time series prediction [5]. For this model the *Keras* implementation for LSTM layers was used [15].

2.2. Weather data

A high-quality weather forecast is indispensable for a precise load prediction. The German Weather Service (DWD¹) offers a fully automated product named MOSMIX that provides statistically corrected weather forecasts

¹ <https://www.dwd.de/>.

for the next ten days for about 5400 locations around the world, mostly in Germany and Europe. The forecasts include common meteorological parameters such as outdoor air temperature, dew point, wind speed, wind direction, surface pressure, sunshine duration, and visibility.

2.3. Correlation analysis

Correlation analysis helps to reduce the amount of possible input parameters to those that have a relation to the target figure. This improves the effectiveness, efficiency and speed of the training process [16]. The Pearson Coefficient is a method of numerically computing linear correlations [17]. However, not all correlations are linear, such as similar loads at the same days of different years. For covering such non-linear correlations, the Maximal Information Coefficient (MIC) [17] can be used. A linear and non-linear correlation analysis was performed on 24 parameters. Table 1 shows the results for some of the most important of these. Relevant parameters often have a high coefficient both with Pearson and with MIC. Although some parameters have a low linear correlation, they can still be valuable input if they show non-linear correlations. E.g., the average outdoor air temperature does not correlate linear with the load demand, but it still provides information about the general temperature trend. The same seems to be the case for seasonal parameters such as month and day of the year which provide meta information that can be connected to common upper and lower limits.

Table 1. Correlation analysis between load and all existing parameters.

Parameter	Pearson	MIC
Last load	0.9905	0.9309
Average load of the last 6 h	0.9718	0.9010
Average load of the last 12 h	0.9622	0.8801
Average load of the last 24 h	0.9605	0.8765
Outdoor air temperature	−0.8304	0.5958
Season sine	−0.8218	0.7948
Dewpoint	−0.8073	0.5541
Month	−0.1891	0.7175
Day of the year	−0.1864	0.8006
Average outdoor air temperature of the last 6 h	−0.0042	0.6940
Average outdoor air temperature of the last 12 h	−0.0037	0.7387
Average outdoor air temperature of the last 24 h	−0.0011	0.7867

2.4. Data pre-processing, training and validation approach

Measured data from the last six years were used for model training and validation. Each hourly data point consists of 20 parameters, i.e., measurement data from the district heating network, or weather information. 45,000 data points (5.3 years) were used for training the forecasting models, and the remainder for validation. Four periods were used for a consistent comparison of the different algorithms:

- Summer period (27.07.2020–09.08.2020)
- Winter period (30.11.2020–15.12.2020)
- Transition period (19.10.2020–01.11.2020)
- Overall period (21.07.2020–15.12.2020)

In order to ensure that errors caused by inaccurate weather forecasts are taken into account in the training, recorded historical weather forecasts were used as input for the trained model.

3. Benchmarking results

3.1. Benchmarking metrics

Benchmarking the forecasts is not an easy task, since no standardized metric is available. The Mean Absolute Percentage Error (MAPE) is probably the most used metric for measuring forecast accuracy. The MAPE is

dimensionless and independent of the magnitude of the values considered. At the same time, it can be clearly interpreted. A MAPE of zero corresponds to the perfect forecast [18,19].

$$\text{MAPE} = \frac{1}{n} \sum_{i=0}^n \frac{|e_i|}{d_i} \quad (1)$$

The normalized Mean Absolute Error (nMAE) avoids the problem of interpretability, as does the MAPE, through a load-related weighting. Since the reference value of the nMAE is not averaged, but summed up, errors with a higher reference are weighted more heavily than errors with a smaller reference. This kind of weighting is optimal for a load forecast, since errors in times of peak loads have a significant influence on plant dispatch planning.

$$\text{nMAE} = \frac{\frac{1}{n} \sum_{i=0}^n |e_i|}{\frac{1}{n} \sum_{i=0}^n d_i} = \frac{\sum_{i=0}^n |e_i|}{\sum_{i=0}^n d_i} \quad (2)$$

In addition, the benchmarking metrics ‘Mean Squared Error’ (MSE) and ‘Mean Absolute Error’ (MAE) were explored for training and evaluation purposes.

3.2. Comparison of load forecasting methods

To find the best model tuning hyperparameters, a broad search with randomly selected parameter settings was performed for each training model. The results were used as a starting point for further tuning. Then, all hyperparameters were kept constant except for one, which was explored over a broad range again to evaluate its sensitivity and find an optimum value. This process was repeated for each hyperparameter.

SARIMAX does not have many hyperparameters. For that reason, autocorrelation function (ACF) and partial autocorrelation function (PACF) plots were used. It was found that a $(0,1,2) \times (0,1,1)_{24}$ configuration gives the best results. As an exogenous parameter, the outdoor air temperature showed the greatest influence on heat demand.

For AdaBoost, the analysis showed that it is necessary to have a sufficiently large amount of data, spanning at least two years (17,520 data points). In addition, the number of implemented features plays a very important role. The best training was achieved when all available input parameters were used.

For LSTM, a low number of epochs was found to provide better results. Generally, the neural network tended to overfit when it was trained for too long. Also using MSE rather than MAE and MAPE as error functions during training resulted in a lower nMAE value during the evaluation process.

Table 2 shows the nMAE forecasting errors of the best models in each category. LSTM achieved the best overall results. While AdaBoost performed slightly better during hot months, LSTM had less deviations compared to the measured load during colder months. SARIMAX showed the worst overall performance, mostly due to larger forecasting errors, for warmer periods with lower heat demands. The error during winter was similar for all methods.

Table 2. Errors of different models for each evaluation period.

Forecast model	AdaBoost	LSTM	SARIMAX
nMAE summer [%]	9.1	9.5	14.7
nMAE transition [%]	9.1	7.8	12.3
nMAE winter [%]	4.7	4.1	4.7
nMAE overall [%]	8.5	7.5	11.5

Fig. 3 shows the measured heat demand for a typical 72-h summer, winter and transition period in comparison to 72-h load predictions for all three trained forecasting models. In addition, the Absolute Percentage Error (APE) for each time step is given. The graphs show that the APE for the winter scenario is noticeable smaller than during summer and transition times. Also, the prediction seems to fit small fluctuations during the day better in winter. This might be due to a strong correlation between colder outdoor temperatures and the need to heat offices and living space. Heating big buildings is often automated and happens at the same time every day. The absolute deviation during summer seems small with less than 1 MW on average. Nonetheless it is still important for the operator to predict peaks to meet the demand without overproducing heat. It should be mentioned that the evaluated network is only one of eight networks supplying the city. First tests shown significant differences depending on the network. Besides different area types, e.g., residential, industrial, commercial, the quality of training data can differ for each

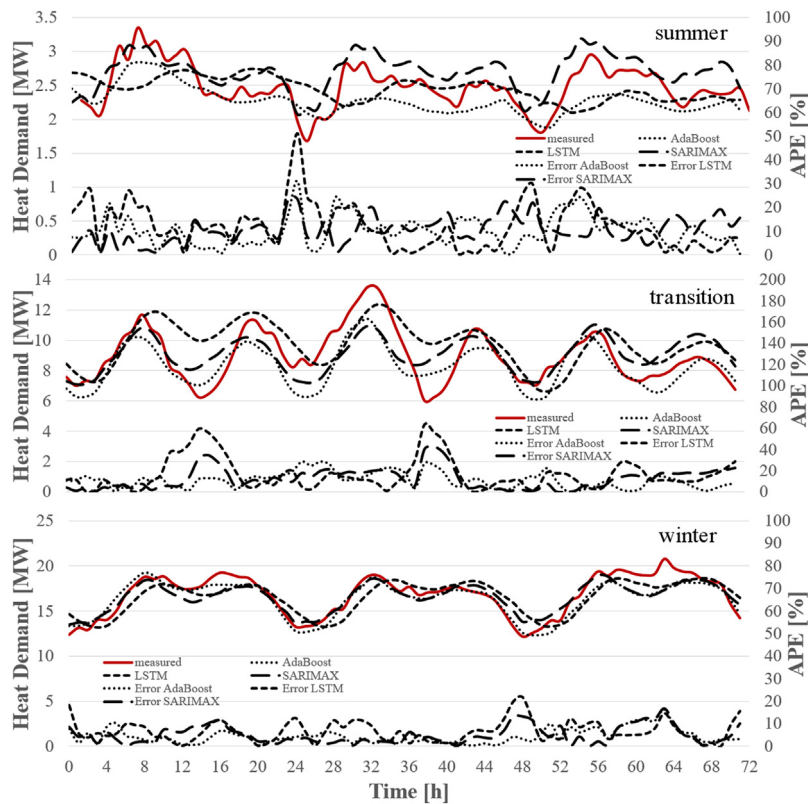


Fig. 3. Measured and predicted heat demand for different evaluation periods and forecasting models.

network. Some networks have less data available due to long maintenance periods. Overall, it seems like each model needs adjustment for each network to achieve the best possible result.

3.3. Discussion of results

The performance of forecasting models can vary significantly depending on the season. All in all, AdaBoost and LSTM were able to produce the most reliable and accurate thermal load predictions. In particular, the SARIMAX approach exhibited significantly higher deviations for most evaluation periods. There is potential to further improve forecasting accuracy by specifically optimizing models for different seasons, or by stacking different forecasting methods. Optimizing colder months is of particular interest due to the high peaks and fluctuations from day to day. Predicting those peaks precisely would have the biggest potential to reduce CO₂ emissions and operating costs [20].

4. Practical implementation

Apart from the machine learning activities described above, a tool was developed that gives the operator real load predictions in a production environment. Such a system should be robust, fast, and intuitive to use for people that do not have any knowledge about the prediction processes running in the background. The final architecture that fulfills these requirements is illustrated in Fig. 4.

The core component of the system is a database, containing both historical data and predictions made by the system itself. It receives data from two other systems, namely the Preprocessor and the Predictor. The Preprocessor handles the gathering and filtering of any external data, such as historical weather data, current loads and weather forecasts. It then adds the processed data to the database. The Predictor can query the data relevant to the current point in time for predicting the loads for the next 72 h, which are then added again to the database. A Web-Interface can query these predicted loads and any other additional data and present it to the operator in a visually appealing

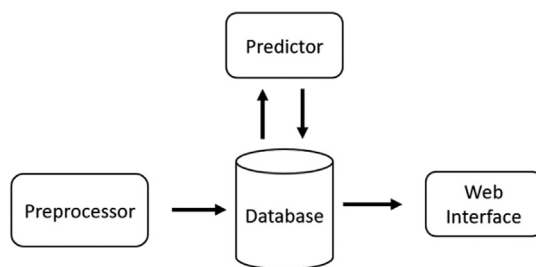


Fig. 4. Components of an end-user system.

way. It gives the operator the ability to manipulate the data visualization, e.g., turning on and off certain parameters, to focus only on the data currently needed.

This multi-system architecture closely resembles the architecture of Self-Contained Systems (SCS) proposed by Tilkov and Ghadir [21], but some parts, e.g., a web interface for each system, were not required in this project. Still, the proposed configuration shares many advantages with SCS architectures, such as the ability to make local decisions that affect only one system, faster iteration time during development, and an easy replaceability of systems. This allows improvements to the quality of predictions without continuously deploying slightly changed versions to the end user. The decoupling also makes the system more robust against failures, in that if one of the systems crashes, all the others can continue their work. There is no communication between the systems, which allows for either running all on one server or distributing them onto multiple machines.

Besides a robust backend, the application must be easy to use for the operator. Designing a good user experience requires gathering feedback from actual operators from the outset and developing the interface iteratively [22]. The interviews conducted in this study showed that the operators preferred a more complex solution over a simple integration of the predicted load into their current workflow. Based on this information, the currently implemented user interface displays graphs showing the actual load and the predicted load for a longer timeframe, in order to give the operator the opportunity to reason about the predictions made by the system. Additionally, the user interface also provides a means of inspecting older load and prediction data to compare them to the current situation and to help make educated guesses based on past results.

5. Conclusions and outlook

Using machine learning techniques could contribute to significantly improving thermal load forecasting. This paper presents the initial results of a broader research activity that focuses on thoroughly analyzing and benchmarking different machine learning-based forecasting approaches for district heating and cooling networks. As a starting point, both typical seasonal characteristics of measured thermal load profiles as well as correlations between key parameters and the load itself were analyzed. All measured data, including statistically optimized point weather forecasts, were automatically pre-processed prior to the actual training and validation steps. Based on detailed historical and actual operating and weather data, different types of machine learning algorithms were used to predict thermal loads 72 h in advance. The algorithms were benchmarked against one another by means of different standardized load regimes. Live operation of machine learning-based load forecasting is going to be demonstrated at one of the heating network operator's facilities. To allow robust and flexible operation, a multi-component software architecture and a user interface showing the actual and predicted loads for a longer timeframe were implemented.

As a next step, more advanced machine learning approaches, including stacking of different techniques, is planned. In addition, integration of smart metering data and automated training and generation of load forecasts will be investigated. Furthermore, the web interface will be made available to selected persons within the power plant. Thereby it is possible to gather vital information about the general performance, unexpected edge cases and the usability of the interface. If those tests are satisfactory it is planned to also integrate the interface in the control center.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was funded by Germany's Federal Ministry for Economic Affairs and Energy under the funding code 03EN3017, with additional financial and in-kind support from the project partners AGFW, Germany, Fernwärme Ulm GmbH, Germany and ZAK Energie GmbH, Germany. The responsibility for the content of this publication lies with the authors.

References

- [1] Benalcazar P, Kamiński J. Short-term heat load forecasting in district heating systems using artificial neural networks. *IOP Conf Ser: Earth Environ Sci* 2019;214.
- [2] Werner Sven. International review of district heating and cooling. *Energy* 2017;137:618–29.
- [3] Saloux Etienne, Candanedo José A. Forecasting district heating demand using machine learning algorithms. *Energy Procedia* 2018;149:59–68.
- [4] Freund Y, Schapire RE. Experiments with a new boosting algorithm. In: *Machine learning: Proceedings of the thirteenth international conference*. 1996, p. 48–56.
- [5] Hochreiter Sepp, Schmidhuber Jürgen. Long short-term memory. *Neural Comput* 1997;9:1735–80.
- [6] Fang Tingting, Lahdelma Risto. Evaluation of a multiple linear regression model and SARIMA model in forecasting heat demand for district heating system. *Appl Energy* 2016;179:544–52, Elsevier.
- [7] Jaeger Herbert. The echo state approach to analysing and training recurrent neural networks-with an erratum note. In: *German national research center for information technology GMD technical report*, Vol. 148. 2001.
- [8] Ling Guo, Weiwei Du, Jiancheng Yang, Liping Wang, Ping Wan, Ling Liu. A thermal load forecasting algorithm based on trajectory tracking. *Math Probl Eng* 2020;1–12.
- [9] Xue Puning, Jiang Yi, Zhou Zhigang, Chen Xin, Fang Xiumu, Liu Jing. Multi-step ahead forecasting of heat load in district heating systems using machine learning algorithms. *Energy* 2019;188.
- [10] Wolpert David H. Stacked generalization. *Neural Netw* 1992;5(2):241–59.
- [11] Zhao Kaiyi, Li Li, Cai Saihua, Sun Ruizhi. An optimized time series combined forecasting method based on neural networks. *J Supercomput* 2020;76:2986–3012.
- [12] Box George EP, Jenkins Gwilym M, Reinsel Gregory C, Ljung Greta M. *Time series analysis forecasting and control*. *J Time Series Anal* 2016;37.
- [13] Seabold Skipper, Perktold Josef. *Statsmodels: Econometric and statistical modeling with python*. In: *Proceedings of the 9th python in science conference*. 2010, pp. 57–61.
- [14] Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, et al. *Scikit-learn: Machine learning in Python*. *J Mach Learn Res* 2011;12:2825–30.
- [15] Chollet F, et al. *Keras GitHub*. 2015, Retrieved from <https://github.com/fchollet/keras>.
- [16] Géron A. *Hands-On machine learning with Scikit-learn and TensorFlow: concepts, tools, and techniques to build intelligent systems*. 1st ed.. O'Reilly, Beijing Boston Farnham Sebastopol Tokyo; 2017.
- [17] Clark Michael. *A comparison of correlation measures center for social research*. University of Notre Dame; 2013.
- [18] Armstrong JS, Collopy F. Error measures for generalizing about forecasting methods: Empirical comparisons. *Int J Forecast* 1992;8:69–80.
- [19] Myttenaere Arnaud De, Golden Boris, Grand Bénédicte Le, Rossi Fabrice. Mean Absolute Percentage Error for regression models. *Neurocomput Adv Artif Neural Netw Mach Learn Comput Intell* 2015;192:38–48.
- [20] Faber Till, Finkenrath Matthias. Load forecasting in district heating systems using stacked ensembles of machine learning algorithms. *Atlantis Highlights Eng* 2020;6.
- [21] Tilkov S, Ghadir P. *Softwarearchitektur im Großen*. In: *Online-themenspecial architekturen*. 2010.
- [22] Green P. *Iterative design*. In: *Industrial and operations engineering Vol. 436 (Human Factors in Computer Systems)*. Ann Arbor, MI: University of Michigan; 2008, Lecture presented.