

# Electricity load forecasting for industrial microgrid and load management

Ariane Franke  
OTH Regensburg  
Forschungsstelle für Energienetze  
und Energiespeicher  
Email: ariane1.franke@st.oth-regensburg.de

**Abstract**—As more energy stems from renewable sources and the electricity market becomes more volatile, new solutions to ensure the security of supply are tested. Microgrids offer a possibility to prevent construction downtime. In order to use the energy sources and storage facilities of the microgrid effectively, load forecasting algorithms are essential. Thus, in this study a short term load forecasting model for a construction company is designed, that will be used for managing a microgrid as well as the general load consumption. This paper proposes the use of the XGBoost algorithm for the 36-hour forecast including predictors based on past measurements as well as information extracted from the timestamp. Moreover, it is shown that including load profiles attained by conventional methods has the potential to improve the accuracy of the model.

## I. INTRODUCTION

Due to the highly fluctuating behavior of the energy market driven by demand and offer, load forecasting has been a research topic for years [1]. New developments in recent years like the surge of renewable energy sources lead to new challenges in the field. In order to integrate the volatile new energy sources, the power grid is monitored more closely [2]. The availability of sensor data enables the application of methods, that can help to ensure a high quality of supply. Moreover, renewable energy sources and smart storage solutions can help secure the supply during power outages. Local solutions or so-called microgrids are a possibility for the production industry to adapt in order to prevent financial loss due to production downtime. To achieve this, planning and managing as well as more accurate short and long term load forecasting algorithms are essential.

In this study the goals of a mid-sized construction company are considered. The company is designing a microgrid including energy sources like photovoltaic systems, wind energy plants and block heat and power systems as well as energy storage facilities. In case of a power outage the connection to the main grid will be separated and the microgrid will be used to keep the production lines going as smartly as possible until a reconnection is possible again. In case the stored energy is not sufficient to keep all machines running during the entire power outage, the machines should be shut down in a controlled manner to prevent damage. To accurately manage the load while the microgrid is in use, a load forecasting algorithm is needed. Moreover, the company also plans to use the load

forecasting to smartly manage the load consumption when taking energy from the main grid as high consumption should be avoided in times of high energy prices.

Thus, in this study a data driven load forecasting algorithm will be designed. A machine learning model using XGBoost trees will be tested and the results will be compared with a conventional method using load profiles. A focus of this paper is to show how the two methods can be merged and how load profiles can help improve the accuracy of machine learning models for electricity load forecasting.

The paper is structured as follows. In section II some related literature is presented. Afterwards in section III the data used in this study is presented and the objectives are specified. Section IV contains a review of the utilized algorithm and is followed by the presentation of the results in section V and the conclusion.

## II. RELATED WORK

Gradient Boost trees are often used for short term load forecasting in literature for different forecast horizons as well as settings. Wang et al. [3] use gradient boost decision trees to predict the power output of photovoltaic systems using historical data as well as weather data. In [4] the authors use XGBoost successfully to forecast a single time lag using Australian energy market load data. In [5] Porteiro et al. evaluate seven regression models for a 24-hour load forecast for an industrial park and show that the Gradient Boost Trees are among the three best performing algorithms. The authors used date and historic load features as input data. In [6] the XGBoost outperforms four other methods for a 168-hour forecast. In [2] Lang compares XGBoost to one dimensional convolutional neural network models for a 36-hour forecast of household loads. The features used in this study resemble the ones used by Lang. However, contrary to [2] this study examines the effect on including load profiles as predictors for the XGBoost model.

## III. DATA SET AND OBJECTIVE

The data set used in this study contains two years and six months of data from January 2019 to June 2021 and stems from a construction company based in Germany. Measurements are available in 15 minute intervals. The load data is derived from balance data between the factory and the

public grid as well as an aggregation of all energy supplying sources within the factory. It therefore represents the total load consumption of the company. The data was cleaned by domain experts and data imputation on the basis of similar days was performed. The data set is split into training and test data. While the first two years are used to fit the model and to perform hyperparameter tuning using 3-fold cross-validation, the last six months are used to assess the models.

The goal is to design a short-term-forecasting model that produces a 36-hour forecast. A focus should lie on the first hour, as the short term load changes are the most essential for the microgrid management. However, the 36-hour forecast is important for the general load management of the company. The objective of this study is to show how the use of load profiles can improve an electricity load forecast model utilizing the XGBoost algorithm.

#### IV. METHODS

In this section the methods used in this study are presented closer. Subsection A presents the conventional method using load profiles, while subsections B and C focus on the XGBoost tree as well as the combined model. All models will be evaluated by calculating the Mean Absolute Percentage Error (MAPE) for all 15 minute intervals of the 36-hour forecast

##### A. Load Profiles

A well established method for planning and managing energy consumption is to analyze load curves [7]. To estimate the future load a comparable measurements from a load curve formed by past values can be chosen. This approach utilizes the seasonality and the distinct patterns that are present in the data. A preliminary exploration of the data set at hand reveals strong daily and weekly patterns. They are typical for industrial load data, due to the orderly workflow in companies. As the electricity demand correlates strongly with the working hours, the daily load progression mirrors the end and start of the work day as well as lunch and coffee breaks. For the same reason the load progression of weekdays differ significantly from weekends and holidays. Therefore it is reasonable to suggest, that the future load is similar to a load measurement recorded on the same weekday at the same time of the day. To make the estimation more robust to random behavior the average of all data points recorded at the same time during the week is taken. The results of this procedure can be seen in figure 1 and will be referred to as load profiles throughout this paper.

##### B. XGBoost Trees

Popular machine learning-based methods for load forecasting are tree-based models. The XGBoost algorithm used in this study is an implementation of a gradient boost regression model, which uses an ensemble of regression trees for prediction.

Regression trees are binary decision trees predicting a numeric value by splitting the predictor space or feature space into regions [8]. Further information on feature engineering and

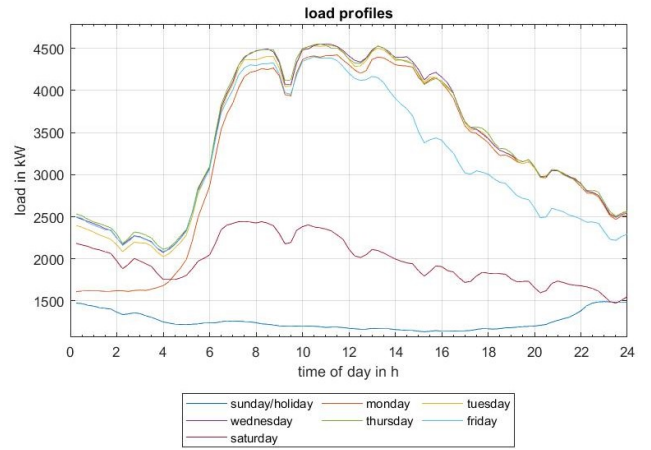


Figure 1. Load profiles attained by averaging data points from identical times of day for different weekdays.

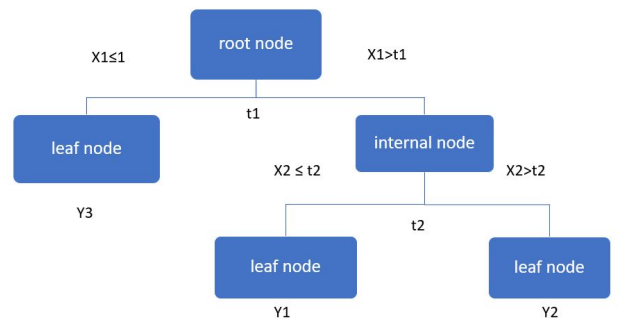


Figure 2. Schematic of a decision tree.  $Y_1, \dots, Y_3$  represent the target values. The decisions are represented by the split points  $t_1$  and  $t_2$ .  $X_1$  and  $X_2$  are the predictors that contain the information relevant for the prediction.

which features are used in this study, will follow in subsection C. Figure 2 shows the basic structure of a decision tree. The data points are divided by certain split decisions until a leaf node is reached. The regression result is the average of all data points that end up in a specific leaf node. When fitting a tree to a training data set the split decisions are optimized to minimize the prediction error [8].

Ensemble learners are based on the notion, that the model accuracy can be improved by aggregating several simpler models [9]. In other words, many weak predictors form a stronger one. One way of building an ensemble of regression trees is called boosting. Contrary to other aggregation models, like for example bagging, boosting is a sequential algorithm, where each fitted weak predictor depends on the results of the previous ones. This means in each new step the model puts the most effort into reducing the biggest errors made by the previous model [8]. The most often used boosting algorithm for regression tasks is gradient boosting. Each new tree is built using the prediction errors or pseudo-residuals of the previous ensemble of trees. Next the tree is scaled by multiplying its results with a parameter called learning rate or

shrinkage. Adding the scaled tree to the ensemble represents a small step along the negative gradient of the loss function [8]. After updating the model the training data is evaluated, the new pseudo-residuals are computed and the next tree is built. XGBoost (Extreme Gradient Boosting) is a framework that effectively implements gradient boost regression by adding features like awareness of sparse data and parallelization [10]. It is known for the speed of the computations. Further advantages of gradient boost algorithms are, that the single decision trees can be visualized and have a comprehensible structure. Moreover, they can work with numeric as well as categorical predictors, that do not need to be standardized in any way [8].

When using XGBoost several hyperparameters have to be set. For the analysis in this study a set of parameters that was successful for a similar application [2] is used. To justify this approach a GridSearch is performed for a few key parameters, namely the number of trees, the maximum tree depth of each regression tree, the learning rate and the subsampling rate, that determines the number of training data points that are used to build each tree. The examined settings are the following:

- number of trees: 100, 200, 400
- maximum tree depth: 3, 9, 12
- learning rate: 0.01, 0.1, 0.3
- subsampling rate: 0.4, 0.5, 0.7, 0.9

All other parameters are left at their default as described in the documentation of the XGBoost framework [11].

### C. Feature Engineering

In order to build meaningful tree structures and to prevent overfitting there should be little redundant information in the training data [2]. Instead of using the original time series data as the input, the goal is to extract characteristic and discriminative properties also called features from the data. Therefore, a key aspect of successfully using the XGBoost algorithm for electricity load forecasting, is to convert the input time series to a feature vector that summarizes the most important information for the prediction and is the input for the decision trees.

There are several different groups of features that can be used. To include information about the load progression some data from not too distant time stamps are used as features. As in this study the desired forecast horizon is 36 hours, the load from 48 hours ago as well as from 1 week, 2 weeks and 3 weeks ago can be used for the entire forecast. These particular features were chosen as they represent load data from similar times in the day or week. Furthermore, the last recorded load data point is added to the feature vector, as it holds important information about the most recent changes. Thus, for the first forecast point the load from 15 minutes prior is included, while for the last forecast point the load from 36 hours back is used. To gain further insight into the historic load consumption, features of the last fully recorded day are added. There are many possible options for features, that could be utilized as for example presented in

the framework tsfresh [12]. Corresponding to [2] in this study statistical features such as the mean, maximum and minimum load as well as the 0.25 and 0.75-percentiles are included.

For time series data with strong seasonality, date features that are extracted from the time stamp of the data points can be useful. In this study the month, the day of the week and the hour of the day are added to the feature selection. Furthermore binary features that state whether the day is a holiday and whether it is a weekday or not are included. Many studies also include weather features such as temperature or sunshine duration. The idea is to capture information about seasonal changes as well as more volatile day to day changes, as the load consumption is postulated to correlate with the weather. This is feasible as generally more electricity is needed for lighting and heating when it is dark and cold. However, this is not necessarily the case in the industrial setting considered in this paper, as the lighting in production halls is switched on regardless of the weather and the main electricity consumption stems from the usage of machines for manufacturing. Thus, weather features are not included in this study.

One goal of this study is to test the idea of merging the conventional approach presented in subsection A, with the machine learning approach outlined in subsection B. To achieve this the load profiles are used as input features for the regression trees. In order to identify the effect of the load profile feature, models with and without it are evaluated.

## V. RESULTS AND DISCUSSION

When using the load profiles to predict the load, the rough progression of the daily load changes can be approximated. Moreover, as pictured in figure 3 the major changes between the load consumption during holidays, such as the beginning of January or April, and normal workdays are identified and forecasted correctly. However, the more subtle day-to-day changes are not followed by the model. This leads to regular prediction errors of about 1000 kW. The maximum prediction error amounts to 3500 kW. The MAPE of the model is calculated to 17%. This result is valid for the entire 36-hour forecast, because the available information does not change with longer forecast horizons.

The hyperparameter search for the XGBoost model showed that a learning rate of 0.01 is not suitable as it increases the forecast error drastically. However, all other combinations of parameters only have a minor effect on the results. Thus, the following set of parameters was maintained throughout the examinations:

- number of trees: 100
- maximum tree depth: 9
- learning rate: 0.1
- subsampling rate: 0.5

The prediction results of the XGBoost model follow the true load progression more closely, justifying the application of the machine learning model. The first model includes all features except for the load profile feature. Figure 4 shows the

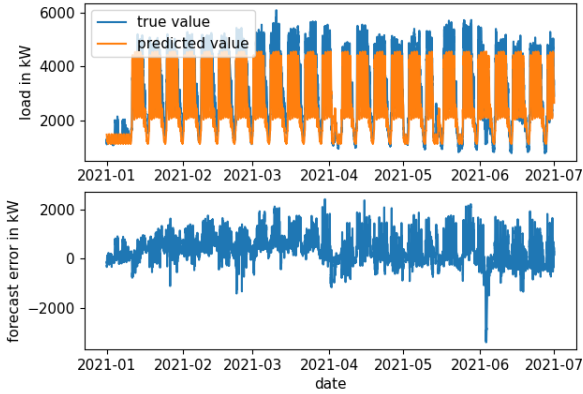


Figure 3. Forecast results using the load profiles for prediction. The true load progression as well as the predicted value and the forecast error for the 15-minute forecast are pictured.

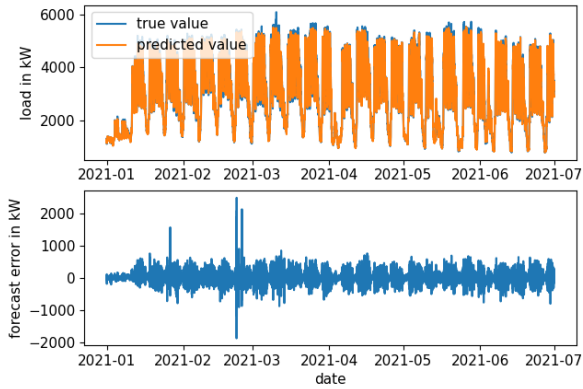


Figure 4. Forecast results using XGBoost without using the load profile as a feature. The progressions for the 15-minute forecast are pictured.

progression as well as the prediction errors for the 15-minute forecast. As the algorithm takes into account the most recent load measurement, it more accurately models fluctuations. The prediction error ranges between 700 and -700 kW with two outliers in the end of January at 1600 kW and the end of February at 2500 kW. The MAPE for the 15-minute forecast is calculated to 3.5 %. To assess the model for the whole 36-hour forecast, the MAPE is calculated for each 15-minute interval and pictured in figure 5. For the first six hours of the forecast horizon a rapid increase of the MAPE is observed. For the seven hour to 35-hour predictions the MAPE is constant at about 12 % with a minor dip at the 24 hour-forecast. Also illustrated in figure 5 is the result of a model that uses the load profile feature instead of the most recent measurement. Its MAPE calculates to 11 %. It can be observed that this second model outperforms the first one for forecast horizons longer than 5 hours. However, for more short-term predictions up to five hours the first model succeeds.

To combine the advantages of the first two models a third

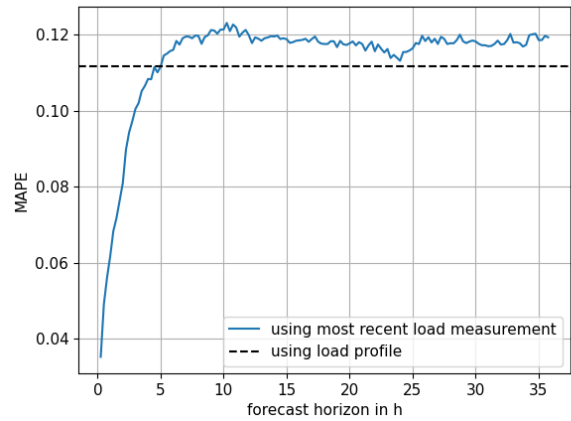


Figure 5. MAPE for the 36-hour forecast horizon for the XGBoost model. The results show the effect of switching out the feature including the most recent measurement for the load profile feature.

model containing the most recent measurement as well as the load profile as features is created. Figure 6 shows that the general progression of the MAPE over the forecast horizon is similar to the first model. However, it does not exceed a MAPE of 11 % which is the value of the second model. It can further be observed that the slope of the MAPE up to the 5-hour forecast horizon is steeper than in the first model. Hence, to predict the load about 3 to 5 hours in the future the combined model performs slightly worse.

The strong increase of the MAPE in models one and three is explainable as for a shorter forecast horizon the model has information about very recent load changes. When predicting a load further in the future, the changes that might happen in the meantime cannot be incorporated as they are unknown, which leads to bigger prediction errors. The small dip in figure 5 confirms the approach of using similar times of the day as features, as it shows that using the load data from exactly 24 hours prior, which is a measurement from the exact same time of day, reduces the prediction error. In general the results show that using the last measured load data points helps improve the predictions for the first hours drastically. Moreover, it is shown that for the 36-hour forecast using the load profile as a feature for the machine learning algorithm improves the accuracy especially for the day ahead. This demonstrates that merging conventional and machine learning models is a promising approach.

## VI. CONCLUSION

This study presented the development of a model for a 36-hour load forecast in an industrial setting. The data set is provided by a construction company, with the goal to use the load forecasting algorithm for managing the company microgrid as well as the general load consumption. The results show that the XGBoost algorithm is suitable for the task as it outperforms conventional methods. It was demonstrated that the model can be improved by proper feature selection.

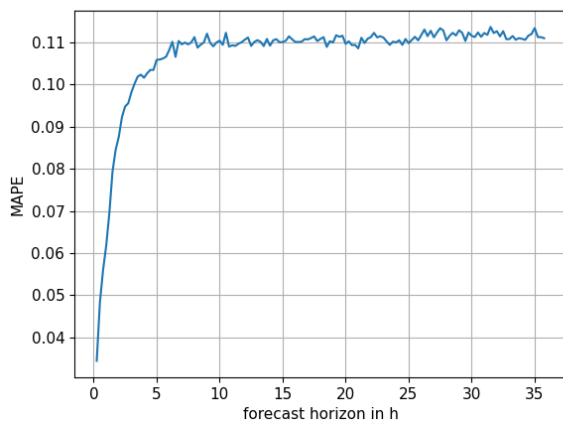


Figure 6. MAPE for the 36-hour forecast for the XGBoost model. The results for the combined model using the most recent measurement as well as the load profile as features is pictured.

Especially the effects of using the load profile built for the conventional method as a feature were highlighted. For the day-ahead forecast the combined model significantly improved the results. On the one hand this confirms the potential of combining conventional and machine learning methods. On the other hand it shows that for electricity load forecasting the load profiles contain significant information that can help enhance the forecast. In future work the effect of using the load profile with other machine learning algorithms could be tested.

## REFERENCES

- [1] C. Wang, T. Back, H. H. Hoos, M. Baratchi, S. Limmer, and M. Olhofer, "Automated Machine Learning for Short-term Electric Load Forecasting," in *2019 IEEE Symposium Series on Computational Intelligence (SSCI)*. Xiamen, China: IEEE, Dec. 2019, pp. 314–321. [Online]. Available: <https://ieeexplore.ieee.org/document/9002839/>
- [2] C. Lang, "Machine Learning Approaches for Energy Forecasting," 2021, publisher: Universität Regensburg. [Online]. Available: <https://epub.uni-regensburg.de/id/eprint/46447>
- [3] J. Wang, P. Li, R. Ran, Y. Che, and Y. Zhou, "A Short-Term Photovoltaic Power Prediction Model Based on the Gradient Boost Decision Tree," *Applied Sciences*, vol. 8, no. 5, p. 689, Apr. 2018. [Online]. Available: <http://www.mdpi.com/2076-3417/8/5/689>
- [4] R. A. Abbasi, N. Javaid, M. N. J. Ghuman, Z. A. Khan, S. Ur Rehman, and Amanullah, "Short Term Load Forecasting Using XGBoost," in *Web, Artificial Intelligence and Network Applications*, L. Barolli, M. Takizawa, F. Khafa, and T. Enokido, Eds. Cham: Springer International Publishing, 2019, vol. 927, pp. 1120–1131, series Title: Advances in Intelligent Systems and Computing. [Online]. Available: [http://link.springer.com/10.1007/978-3-030-15035-8\\_108](http://link.springer.com/10.1007/978-3-030-15035-8_108)
- [5] R. Porteiro, S. Nesmachnow, and L. Hernández-Callejo, "Short Term Load Forecasting of Industrial Electricity Using Machine Learning," in *Smart Cities*, S. Nesmachnow and L. Hernández Callejo, Eds. Cham: Springer International Publishing, 2020, vol. 1152, pp. 146–161, series Title: Communications in Computer and Information Science. [Online]. Available: [http://link.springer.com/10.1007/978-3-030-38889-8\\_12](http://link.springer.com/10.1007/978-3-030-38889-8_12)
- [6] E. Aguilar Madrid and N. Antonio, "Short-Term Electricity Load Forecasting with Machine Learning," *Information*, vol. 12, no. 2, p. 50, Jan. 2021. [Online]. Available: <https://www.mdpi.com/2078-2489/12/2/50>
- [7] C. Mihai, I. Lepadat, E. Helerea, and D. Calin, "Load curve analysis for an industrial consumer," in *2010 12th International Conference on Optimization of Electrical and Electronic Equipment*. Brasov,

- Romania: IEEE, May 2010, pp. 1275–1280. [Online]. Available: <http://ieeexplore.ieee.org/document/5510494/>
- [8] J. Elith, J. R. Leathwick, and T. Hastie, "A working guide to boosted regression trees," *Journal of Animal Ecology*, vol. 77, no. 4, pp. 802–813, Jul. 2008. [Online]. Available: <https://onlinelibrary.wiley.com/doi/10.1111/j.1365-2656.2008.01390.x>
- [9] R. E. Schapire, "The Boosting Approach to Machine Learning: An Overview," in *Nonlinear Estimation and Classification*, P. Bickel, P. Diggler, S. Fienberg, K. Krickeberg, I. Olkin, N. Wermuth, S. Zeger, D. D. Denison, M. H. Hansen, C. C. Holmes, B. Mallick, and B. Yu, Eds. New York, NY: Springer New York, 2003, vol. 171, pp. 149–171, series Title: Lecture Notes in Statistics. [Online]. Available: [http://link.springer.com/10.1007/978-0-387-21579-2\\_9](http://link.springer.com/10.1007/978-0-387-21579-2_9)
- [10] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. San Francisco California USA: ACM, Aug. 2016, pp. 785–794. [Online]. Available: <https://dl.acm.org/doi/10.1145/2939672.2939785>
- [11] xgboost developers. Xgboost documentation. [Online]. Available: <https://xgboost.readthedocs.io/en/latest/>
- [12] M. Christ, N. Braun, J. Neuffer, and A. W. Kempa-Liehr, "Time Series Feature Extraction on basis of Scalable Hypothesis tests (tsfresh – A Python package)," *Neurocomputing*, vol. 307, pp. 72–77, Sep. 2018. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S0925231218304843>