Received: 15 May 2023 | Revised: 18 September 2023 | Accepted: 9 October 2023 | IET Smart Grid

DOI: 10.1049/stg2.12137

The Institution of Engineering and Technology WILEY

ORIGINAL RESEARCH

Privacy-preserving peak time forecasting with Learning to Rank XGBoost and extensive feature engineering

Leo Semmelmann | **Oliver Resch** | **Sarah Henni** | **Christof Weinhardt**

Karlsruher Institut für Technologie, Karlsruhe, Germany

Correspondence
Leo Semmelmann, Karlsruher Institut für Technologie, Kaiserstraße 89-93, Karlsruhe 76133, Germany.
Email: leo.semmelmann@kit.edu

Abstract

In modern power systems, predicting the time when peak loads will occur is crucial for improving efficiency and minimising the possibility of network sections becoming overloaded. However, most works in the load forecasting field are not focusing on a dedicated peak time forecast and are not dealing with load data privacy. At the same time, developing methods for forecasting peak electricity usage that protect customers' data privacy is essential since it could encourage customers to share their energy usage data, leading to more data points for the effective management and planning of power grids. Hence, the authors employ a dedicated Learning to Rank XGBoost algorithm to forecast peak times with only ranks of loads instead of absolute load magnitudes as input data, thereby offering potential privacy-preserving properties. We show that the presented Learning to Rank XGBoost model yields comparable results to a benchmark XGBoost load forecasting model. Additionally, we describe our extensive feature engineering process and a state-of-the-art Bayesian hyperparameter optimisation for selecting model parameters, which leads to a significant improvement of forecasting accuracy. Our method was used in the context of the final round of the international BigDEAL load forecasting challenge 2022, where we consistently achieved high-ranking results in the peak time track and an overall fourth rank in the peak load forecasting track with our general XGBoost model.

KEYWORDS

artificial intelligence and data analytics, data privacy, load forecasting

1 | INTRODUCTION

Ensuring a balance between power supply and demand is crucial for electrical grids' stable and efficient operation. In this context, forecasting future electrical loads plays an integral role [1]. Load forecasting is performed for various planning horizons, from long-term over medium-term to short-term load forecasting with annual, monthly or daily planning horizons, respectively [2]. It can additionally be classified by the aggregation level considered [3].

One essential discipline of load forecasting is peak load forecasting. Peaks are the occurrence of the maximum load in a specific timeframe (e.g. a day) and can be characterised by two dimensions: peak time and peak load quantity. Peak time describes the timestep where the maximum load occurs, while peak load quantity describes the maximum load measured in the respective timestep [4]. As peaks constitute the maximum strain on the grid, predicting the maximum load and especially the timing of peaks is crucial for the grid stability. Yet, most approaches in the discipline of electrical peak demand forecasting are either concerned with only predicting the peak load quantity [5] or the peak time forecast is inferred from an overall load forecast by considering the time when the predicted load curve is at its maximum [6, 7]. As a high general prediction accuracy does not necessarily imply a good prediction quality concerning peak loads and peak times [7], further research specifically about peak times is necessary.

Another increasingly important aspect of load forecasting is the privacy preservation of load data. By sharing the exact load data, new potential data security vulnerabilities are created. For instance, in case of a data leak, the load data of an industrial firm could allow conclusions about the current economic situation of the respective firm. Substantial industrial customers see data security as a significant barrier to participation in load-shifting programs, and thereby a barrier to sharing load data [8]. Hence, to motivate industrial customers to share their load data, participate in grid provider programs and reduce potential vulnerabilities, it is essential to work on privacy-preserving methods to work with load data.

This paper combines a focus on peak time forecasting with a privacy-preserving Learning to Rank model in the context of the BigDEAL Challenge 2022 [9]. In this international competition organised by Dr. Tao Hong, Duke Energy Distinguished Professor at UNC Charlotte and Director of the Big Data Energy Analytics Laboratory (BigDEAL), 78 international teams competed along different tracks related to peak load forecasting. The challenge consisted of a qualification and a final round. In the final round, the three tracks of the challenge were targeted at forecasting peak load quantities, peak times, and the shape of the load in a 5-h timeframe around the peaks. The findings presented in this paper stem from the approach followed by Team SGEM KIT in the final round of the BigDEAL Challenge for the track peak time forecasting.

We propose a novel approach to forecast peak times with a Learning to Rank extreme gradient boosting (XGBoost) model, also used for the peak time forecasting track of the BigDEAL Challenge. We compare our results with a naive day-before benchmark forecast and a general state-of-the-art XGBoost-based load forecasting model that delivers load forecasts for every timestep. The latter model achieved the fourth rank in the peak load forecasting track of the final round of the BigDEAL challenge and hence can be seen as a relevant benchmark. The Learning to Rank model only requires ranks of loads, instead of their actual magnitude, as input. Thus, it can be used in application areas where protecting actual load data is highly relevant, thereby, for instance, potentially encouraging customers to share their data with grid operators.

This paper shows that the Learning to Rank XGBoost model yields comparable forecasting accuracy as a well-performing baseline XGBoost model. Furthermore, we conduct extensive feature engineering. We investigate which features are most important for the general XGBoost and Learning to Rank models. Additionally, we use Bayesian Hyperparameter Optimisation to find optimal hyperparameter combinations.

In conclusion, we aim to make the following contributions:

- An extensive feature engineering process is described, including the implementation of rolling averages and type-of-day features, thereby significantly improving the peak time forecasting accuracy.
- The Learning to Rank XGBoost algorithm is used for peak time forecasting, working only with ranks of loads instead of absolute loads as a target feature, thereby offering potential privacy-preserving properties. The model is compared to a conventional XGBoost load forecasting model, from which peak time forecasts are inferred.
- The XGBoost-based models are optimised with a state-of-the-art Bayesian hyperparameter optimisation, enabling further increases in prediction accuracy.

The remainder of this paper is structured as follows. In the second chapter following the introduction, we set our study in the context of related work. In the third chapter, we describe our general methodology. We depict our feature engineering process, the utilised XGBoost models as well as the regarded metrics. In the fourth chapter, we present the BigDEAL case study and the underlying data. Subsequently, we describe the case study results and the achieved forecasting accuracy according to the previously introduced metrics in chapters five and six. Finally, in chapter seven, we discuss our results and give an outlook to further research questions.

2 | RELATED WORK

In this section, we give an overview of related peak load and peak time forecasting studies, with a special focus on studies that cover privacy-preserving features.

Most load forecasting-related studies focus on an overall load forecast, most often through neural network-based methods, Support Vector Machines (SVM) or Auto-Regressive Integrated Moving Average (ARIMA) [10-13]. The first advances in load forecasting were, amongst others, made with ARIMA-based models. In ref. [14], an hourly short-term electrical ARIMA load forecasting model is introduced. Lee et al. improve the ARIMA model by using a lifting scheme wavelet transformation to enhance the forecasting accuracy in ref. [15]. Another way of enhancing the ARIMA load forecasting is suggested by the authors in ref. [16], employing a hybrid ARIMA and SVM model, where the ARIMA model forecasts the linear basic load component and the SVM is used for non-linear components.

In contrast, recent load forecasting studies mostly focus on neural network-based methods. In ref. [10], the authors use Long Short-Term Memory (LSTM) recurrent neural networks to forecast single-residential household loads. Also, the authors in ref. [17] use a dual-stage model, based on an LSTM and an attention-based encoder, for a probabilistic load forecasting model. The temporal attention mechanism, in combination with a Convolutional Neural Network, is employed by the authors in ref. [12] as well. Another type of neural network is suggested by the authors in ref. [13] who use a transformer model, based on an encoder-decoder architecture, for a multi-energy load forecasting problem.

For the planning and operation of modern power systems and distribution grids, the forecasting and subsequent reduction of peak loads are essential [18, 19]. Besides the fact that the before-mentioned studies are focused on an overall load forecast, several studies emphasise the tendency of neural network-based approaches to underestimate peak loads [7, 20, 21]. Hence, the authors in ref. [20] adapt the LSTM cost function to penalise underestimation of the load. The authors in ref. [7] pursue another approach by combining LSTMs with a dedicated peak time and peak load XGBoost forecast. Thereby, the overall load and peak load forecasting accuracy are improved. Also the authors in ref. [22] develop a hybrid LSTM-XGBoost load forecasting model. The authors state that the XGBoost forecast could be further improved by employing a Bayesian hyperparameter optimisation method to find more suitable parameters. Our study implements the suggested Bayesian hyperparameter search approach for improving forecasting accuracy.

Haida et al. [23] were amongst the first to focus on peak load forecasting specifically. The authors combine a transformation technique to consider seasonal load changes, as well as annual load growth, with a multivariate regression analysis. Thereby, the authors reduce forecasting errors in transitional seasons such as spring and fall. In ref. [21], the importance of peak load forecasting for dispatching centres in power networks is highlighted. The authors focus on peak load forecasts with a dedicated ARIMA model alongside the overall hourly load forecast. In ref [24], the peak loads for up to 7 days ahead were forecasted with a feed-forward neural network, combined with a Principal Component Analysis for factor extraction. Besides peak loads itself, also forecasting the time of peak loads plays an essential role. The authors in ref. [25] show that significant improvements in overall load forecasting accuracy can be reached by focussing on peak time forecasting. Within the study, the load demand time series is decomposed into low-frequency components. Then, a peak load binary variable is derived from the value at risk concept, with the aim of improving forecasting accuracy during peak times. Finally, a deep belief network is trained to forecast future loads. We remark that there are—to the extent of our knowledge—no studies solely focusing on predicting peak times.

Another research stream in smart grid research deals with implementing privacy-preserving methods. In ref. [26], the authors discuss two potential privacy protection schemes for short-term load forecasting: model-distribution predictive control (MDPC) and load-level, in combination with support vector regressions. The study concludes that the MDPC has a slight negative impact on forecasting accuracy for smaller aggregations of loads, which diminishes with higher aggregation levels. On the other hand, the load-levelling approach manages to improve load forecasting accuracy. Another privacy-preserving load forecasting approach is presented in ref. [27], where load forecasting models of residential customers are trained on distributed smart metres and handled locally. Then, only the forecasting outputs are reported to the cloud through fog nodes. The authors in ref. [28] are suggesting a privacy-preserving model for electricity theft detection by adding Gaussian noise to the consumption data of customers before applying a Convolutional Neural Network, aiming to achieve a balance of customer privacy and model accuracy.

A further important concept in privacy-preserving forecasting research is differential privacy [29]. In the context of differential privacy, noise is added to the input data until it can no longer be used to confidently predict which individual delivered the underlying data. In ref. [30], differential privacy is granted by adaptively controlling the gradients of training data, combined with a framework to allocate privacy budgets. Le et al. introduce a novel, privacy-preserving adaption of the XGBoost framework for federated learning [31]. The authors use a secure matrix multiplication method and a noise perturbation approach in a separate model. In a comparable approach, in ref. [32], a combination of federated learning and differential privacy is utilised for short-term load forecasting. One outcome of the study is that increasing the number of participating consumers not only leads to enhanced forecasting results but also to potentially too high computational costs, especially for complex neural network architectures. Also, in ref. [33], a federated learning model for privacy-preserving forecasting of distributed energy resources, such as solar PV, EV storage or flexible loads, is developed. The authors validate their study with 1000 IoT nodes and show that the approach can be used for grid services such as predicting curtailment events or load swings. A recent advance of federated learning models for privacy-preserving forecasting has been made in ref. [34], where a hierarchically federated model exploits all underlying datasets while enabling information exchange of users with similar load patterns. The state-of-the-art model enables a significant improvement in forecasting accuracy over benchmark models while maintaining a high fault tolerance. For a better balance between privacy and data quality, in ref. [35] a two-step model is suggested. In the first step, a distributed perturbation method is applied on the underlying high-frequency load data. In the second step, through a private noise distribution protocol, noise elements are distributed over the smart metres of individual customers. In a case study, the authors show the utility of the data is maintained, while preserving the privacy of users. Also the authors in ref. [36] investigate the impact of differential-privacy on forecasting quality, underlining that for some methods the introduction of differential-privacy leads to significantly worse forecasts.

We can observe that many past privacy-preserving methods focus on adding noise to the underlying data, for example, the hourly loads, sometimes at the cost of worse forecasting accuracy. Another possible approach to change the underlying data could be the transformation of actual loads to less sensitive ranks of loads, which can then be used to forecast peak times, for example, through the Learning to Rank method introduced by Chapelle et al. [37]. The Learning to Rank method has been applied with the XGBoost method [38], but not in the context of peak time forecasting. The transformation of loads to ranks could yield one big advantage over more sophisticated methods: it is likely to be easier for end-users to implement and comprehend, which, as various studies have shown, is essential for the adoption of novel technologies and smart grid applications [39-42].

Our study fills two essential gaps in research. In the context of the BigDEAL Challenge, we primarily focus on the peak time forecasting task, which has only been discussed to a small extent in prior research. Furthermore, we provide a detailed examination of suitable features for the peak time forecasting problem. Second, we employ the Learning-to-Rank XGBoost algorithm to forecast peak times. We are analysing its performance compared to benchmark approaches, which use actual loads instead of ranks of loads. Thereby, we make an essential contribution to privacy-preserving peak time forecasting.

3 | METHODOLOGY

This section aims to provide a comprehensive overview of the methodology used in the BigDEAL Challenge to forecast peak load times and quantities. First, we give an overview of our general forecasting framework. The following subsections describe the respective steps, beginning with our feature engineering approach. Then, we express our utilised models and our Bayesian Hyperparameter Optimisation. Finally, we introduce the metrics used to evaluate the performance of the previously engineered models.

3.1 | Forecasting framework

The final stage of the BigDEAL challenge consisted of several rounds, each including their respective historical load and temperature observations. For the to-be-forecasted time horizon, only temperature data was given. In the first step, we enrich the given dataset through extension with additional features, which are explained in detail in the following section. The colour of the arrows indicates which data is used as input for the respective steps. All the next steps use fully feature-engineered data. We distinguish between two model types: a dedicated peak time model and a general load forecasting model. The latter serves as a solid, well-performing benchmark and reflects the predominant approach in the existing literature to infer the peak load and peak time forecast from the whole daily forecast [5, 6]. Both model types are described in Section 3.3. For the peak time models, we differentiate between an XGBoost model with standard parameters and a model with tuned hyperparameters. The overall hyperparameter tuning approach is introduced in Section 3.4. The forecasts are then evaluated according to the metrics defined in Section 3.5.

Overall, the utilised forecasting framework can be structured as depicted in Figure 1.

3.2 | Feature engineering

Feature engineering describes the process of creating representations of the raw data that can improve the models' effectiveness. For a high prediction quality, adequate feature engineering is essential, with the effect of feature selection surpassing that of selecting different models in many cases [43]. Below, the different feature engineering techniques used to transform the input data are described:

3.2.1 | Type-of-day features

Type-of-day features are variables that are created by categorising dates, for instance, in groups of working days and non-working days. Past studies have shown that type-of-day features can improve the overall forecasting accuracy when added to the feature set [44]. Hence, we added binary variables for determining whether the day is a weekday, holiday, preceded, or followed by a holiday, respectively. In addition, the weekday as well as month and day of the month are provided to the model as input after a sine and cosine transformation.

3.2.2 | Sine and cosine transformation of cyclical features

Past studies, such as refs. [7, 45] have shown that the sine and cosine transformation of cyclical features, such as the hour or

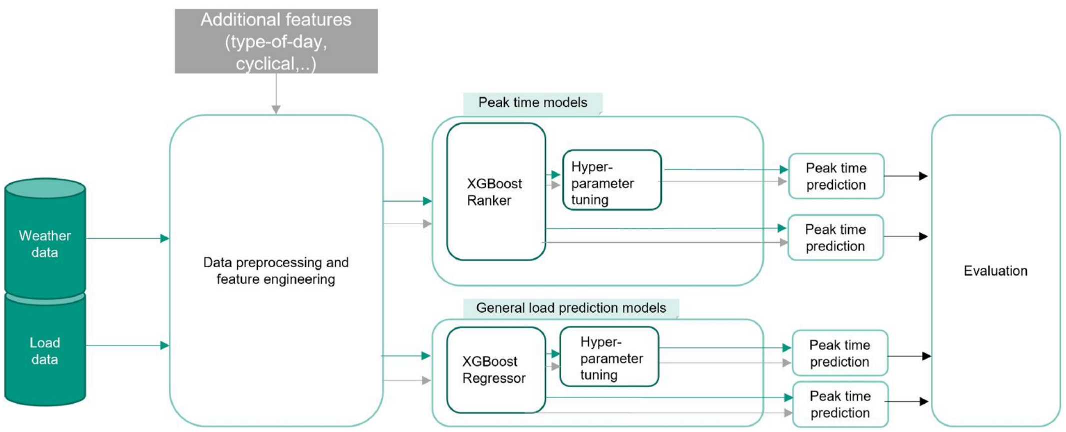

FIGURE 1 Graphical overview of the research approach.

weekday, result in high feature importance and are therefore essential features for electrical load forecasting. The advantage of the sine and cosine transformation lies in a better representation of the cyclical variables, for example, allowing the model to learn that 11 PM is closer to 2 AM than 8 AM. Hence, we implement sine and cosine features for hour, weekday, and month. For the hour, we additionally implement a 2× and 4× sine and cosine, which repeats two and four times, respectively, per day. We apply the sine and cosine transformation by calculating the number of past timesteps since the beginning of the respective seasonal period, for example, a day or month, scaled to a range between 0 and 2π.

3.2.3 | Relative changes of features

For every continuous feature, relative changes up to the past 24 h in one-hour increments are calculated and used as additional input features. This is based on the hypothesis that the rate of change in those features might affect the resulting load pattern, especially if those changes happen suddenly. The formula for the calculation of relative change $G_{\tau,T}$ of feature τ and the time frame tf in every time step t is calculated in Equation (1):

$$G_{\tau,tf} = \frac{\tau_t - \tau_{t-tf}}{\tau_{t-tf}} \tag{1}$$

3.2.4 | Rolling averages

Since the selected model predicts the respective timesteps independently of each other, one potential drawback could be its difficulty in considering dependencies over multiple timesteps. Furthermore, as described in ref. [46], it is also essential to include lagged temperature features in the load forecasting model due to the thermal inertia of buildings. Therefore, rolling averages are calculated for all temperature features over different time frames ranging from 1 to 192 h. For up to 10 h, this is done for every interval length. Above, only the rolling average for 12, 15, 18, 24, 36, 48, 96 and 192, respectively, are calculated. The calculation of the rolling average $RA_{\tau,tf,t}$ for temperature feature τ and time frame tf at timestep t is described in Equation (2).

$$RA_{\tau,tf,t} = \frac{1}{tf}\sum_{n=1}^{tf} \tau_{t-n} \tag{2}$$

Rolling average features have shown a high feature importance in previous research where ensemble-based prediction models were used on time series data, for example, in ref. [47]. We, therefore, calculate a "DiffToRollingAverage" (dRA) data point for every rolling average to detect deviations of the underlying temperature feature from the rolling average, as described in Formula (3):

$$dRA_{\tau,tf,t} = RA_{\tau,tf,t} - \tau_t \tag{3}$$

3.3 | Prediction models

As mentioned in Section 3.1, two different XGBoost-based models are trained to predict the load in every timestep of the test set. The first model is a general load prediction model that forecasts the load of every hour in the respective period. From the forecasted load patterns of the general model, the peak times are inferred to serve as a benchmark for the second model to compete against. The second model is a dedicated peak prediction model, which is based on a novel approach to forecast peak times by employing a Learning to Rank XGBoost model. First, the XGBoost algorithm is presented in general since both used models are based thereon. The following two sections describe both peak load forecasting models in greater detail.

3.3.1 | Extreme gradient boosting (XGBoost)

XGBoost was introduced by the authors in ref. [48] and has been proven as a highly efficient and accurate model for regression and classification tasks. In the load forecasting context, approaches based on XGBoost have often outperformed other models, as shown in refs. [49, 50]. The model is based on an ensemble of classification and regression tree weak learners. Furthermore, the quadratic objective function is simplified through a second-order Taylor expansion, which yields enhanced runtimes and limits overfitting.

3.3.2 | General XGBoost load prediction model

The general load prediction model consists of an XGBoostRegressor that predicts the load for each timestep of the test set individually. Then, for every day d in the test set, the timestep t of the highest load $P_{\max}$ is taken as peak time prediction $t_{d,P_{\max}}$, as depicted in Equation (4)

$$t_{d,P_{\max}} = \max(P_{d,t}, \ldots, P_{d,T}) \tag{4}$$

Hereafter, this model is referred to as XGBP (XGBoost Pattern). We are also considering a hyperparameter-optimised version, which is called XGBPH hereafter. The XGBPH model also served as a model for the peak load and shape prediction tracks of the BigDEAL challenge.

3.3.3 | Learning to Rank XGBoost peak time model

The idea of the proposed Learning to Rank XGBoost peak prediction model is the essential characteristic of a peak, which is the highest load in the considered timeframe, such as a day. In other words, if the loads of a day were ranked by descending load, the peak would always have rank one. Thus, we propose a model that learns to rank the timesteps of a day. Since the Learning to Rank model is working with day-wise ranks as a target variable instead of loads, it requires less sensitive data than traditional approaches. The initial idea of the Learning to Rank model was first described by Chapelle et al. [37].

The Learning to Rank model requires a transformation of the target variable from load to rank. Every load P on a day d can be mapped to a rank r, which ranges from 1 to 24:

$$\left\{ \left(\mathbf{P}_d^t, r_d^t \right) \right\} \tag{5}$$

A rank r of 1 represents the peak load, while rank 24 stands for the lowest load occurring on a certain day d.

In Figure 2, the load is transformed into day-wise rankings of the respective timesteps by descending load for an exemplary time series consisting of 2 days with three timesteps per day.

The previous target variable, "Load," is discarded and not used by the model. The peak time model thus learns with a different target variable than the general load prediction model. The input features remain unchanged. For the prediction of ranks, for every possible rank R a score is calculated based on comparisons of the timesteps in each day. For a detailed description of the score calculation methodology, we refer to ref. [37]. The scores can, in turn, be sorted and turned into rankings for each day.

The prediction of our proposed models is thus day-wise rankings for all timesteps in the test set. In the final transformation step, the timestamp associated with rank one is selected as the peak time for each day. As stated in the overview, we do not only consider one single Learning to Rank peak time model but different variations of it. This leads to two dedicated peak time models that are investigated: XGBR (XGBoost Ranker), a plain XGBoost Ranker without hyperparameter tuning and XGBRH (Learning to Rank XGBoost hyperparameter tuned), where hyperparameter tuning using the methodology defined in the next section is applied to improve the model.

3.4 | Bayesian hyperparameter tuning

Hyperparameter tuning is a crucial task in machine learning and describes the practice of optimising the parameters of the selected model in order to obtain a higher prediction quality. It has been shown that in several cases, baseline models could be improved by hyperparameter adjustments of existing models than by inventing new models [51]. There are several approaches to hyperparameter tuning, including using Bayesian optimisation, as proposed in ref. [52], which belongs to the class of automated hyperparameter tuning. In automated hyperparameter tuning, the model is considered to be a black box function that, given validation data, returns a score, which generally is the chosen error metric to be optimised. The goal of the optimissation is to find hyperparameters that minimise this error. Contrary to the Bayesian optimisation method, traditional optimisation approaches are not suited for this kind of optimisation problem. Bayesian optimisation leverages Bayes theorem for selecting parameters to be evaluated in the true objective function by using a probability model of the objective function, which is, in turn, based on sample data from previous iterations. For a more comprehensive introduction to the general principles of Bayesian optimisation for hyperparameter tuning, we refer to ref. [53].

Bayesian optimisation leads to a significant improvement of hyperparameters with only a few iterations [53], making it both effective and time efficient. Especially time efficiency played a role in selecting this approach as the time for obtaining a prediction and, thus, also for hyperparameter tuning was limited in the competition the approach was developed for. Furthermore, several studies in the load forecasting field have shown that more accurate predictions can be reached by using Bayesian optimised model parameters [54-56].

Our implementation of the Bayesian optimisation hyperparameter selection model is depicted in Figure 3. The approach is based on two major parts: an optimiser and a hyperparameter evaluation function that is minimised over the course of the iterations.

The optimiser describes the Bayesian optimisation model with its parameters as well as the search space. For the optimisation model itself, we use the Bayesian optimisation package [57]. Table 1 depicts the chosen parameters of the optimisation model. The model is initialised with 10 random points and runs for 100 iterations. The parameters *alpha*, which is used for the internal Gaussian process, and *kappa*, which controls the relation between exploitation and exploration, are set as specified in the table. For reproducibility, a random seed is used.

Timestamp	Load (kW)
2018-01-01 12:00am	12
2018-01-01 08:00am	50
2018-01-01 04:00pm	10
2018-01-02 12:00am	20
2018-01-02 08:00am	40
2018-01-02 04:00pm	25

Timestamp	Rank
2018-01-01 12:00am	2
2018-01-01 08:00am	1
2018-01-01 04:00pm	3
2018-01-02 12:00am	3
2018-01-02 08:00am	1
2018-01-02 04:00pm	2

FIGURE 2 Exemplary visualisation of the target transformation for 2 days with only three timesteps.

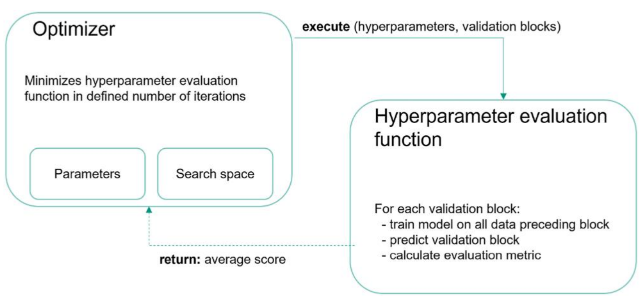

FIGURE 3 Structure of hyperparameter optimisation.

In addition to selecting a model, a search space needs to be set for its hyperparameters. As all of our models are XGBoost-based, they have a shared search space. It is presented in Table 2, where for each hyperparameter that is optimised, the upper and lower boundaries are defined. The parameter max *_depth* is discreet and thus rounded before use. We derive the parameters from past XGBoost-based load forecasting studies [7, 58, 59].

As depicted in Figure 3, in each iteration, the optimiser calls the hyperparameter evaluation function with a set of hyperparameters and validation blocks. The hyperparameters are selected by the optimisation function and thus differ in each iteration. The validation blocks constitute the time frames used by the hyperparameter evaluation function for calculating the model's evaluation metric. We use the whole year 2017 as a validation block to get parameters for an overall robust model.

In the second part, the hyperparameter evaluation function is used to determine the performance of the model for the hyperparameters of the current iteration. For each validation block, the respective model is trained on all available data with time stamps preceding the validation block. Subsequently, the model's performance is evaluated on the validation block. This is performed for each validation block individually, leading to three-fold cross-validation. The function returns the average score over the three blocks. The evaluation metric employed varies depending on which model type is evaluated. For general prediction models, the mean absolute percentage error for the true and predicted loads for each timestep is used. For the dedicated peak time prediction models, the score is based on a daily MAE-based metric, defined in Section 3.5. In the last step, the hyperparameters are selected based on the best scores. For every different Local Distribution Company (LDC), a dedicated hyperparameter tuning run has been conducted.

TABLE 1 Configuration used for the Bayesian optimisation.

Parameter	Value
init_points	5
Nitre	100
gp_alpha	10^{-10}
Kappa	1.5
Seed	112

TABLE 2 Hyperparameter search space.

Hyperparameter	Lower bound	Upper bound
max_depth	3	10
learning_rate	0.01	1.0
Subsample	0.5	1.0
min_child_weight	0.5	5.0
colsample_bytree	0.5	1.0

3.5 | Metrics

In the following, we present the used metrics for the evaluation of our peak time forecasting models.

3.5.1 | Accuracy

In many studies, for example, in ref. [60], peak time forecasting is considered to be a classification task. The reasoning for this is that a peak time forecast is only of use if it predicts the exact time of the peak load event. Thus, accuracy, which is in many cases used to evaluate binary classifications, is a popular metric to evaluate peak time forecasts. The accuracy metric $\mathbb{P}$ is defined as in Equation (6):

$$\mathbb{P}(\text{Actual} = \text{Predicted}) = \frac{Tp + Tn}{N} \tag{6}$$

with Tp and Tn being the amount of correctly predicted positive and negative labels, respectively. N constitutes the total amount of predictions. Hence, accuracy measures the share of correct predictions.

3.5.2 | Mean Absolute Error (MAE)

As a second error metric, we calculate the Mean Absolute Error (MAE), which punishes wrong predictions linearly to the distance to the true prediction of the respective day. It is calculated as the mean of day-wise the absolute deviations of the predicted from the true peak time, as depicted in Equation (7):

$$MAE = \frac{1}{D}\sum_{i=1}^{D} \left| t_{d,P_{\max},pred} - t_{d,P_{\max}} \right| \tag{7}$$

with D being the considered amount of days and $t_{d,P_{\max},pred}$ and $t_{d,P_{\max}}$ being, respectively, the predicted and the actual peak time of day i.

3.5.3 | BigDEAL Peaktime Metric (BDPM)

We also evaluate a dedicated metric, which was introduced in the context of the BigDEAL challenge, called BigDEAL Peaktime Metric (BDPM). The BPDM is a modified version of a cumulative absolute error that punishes higher deviations more strongly by introducing a punishment factor. At the same time, it is capped for deviations greater than 5 h. In its original form, the error is cumulated over the whole considered timeframe. To ensure comparability over test blocks with differing lengths, we decided to norm this error by the number of days considered. Our normed version of the BDPM is defined in Equations (8) and (9):

$$BDPM = \frac{1}{D}\sum_{i=1}^{D} f\left(\left|t_{d,P_{\max},pred} - t_{d,P_{\max}}\right|\right) \tag{8}$$

with

$$f(r) = \begin{cases} r & \text{for } r \leq 1 \\ 2 * r & \text{for } 2 \leq r \leq 4 \\ 10 & \text{for } r \geq 5 \end{cases} \tag{9}$$

4 | CASE STUDY

The previously described methodology was applied on the data set provided by the initiators of the BigDEAL Challenge 2022. The data set comprises historical load data of three U.S. neighbouring local distribution companies (LDCs), along with temperature data from six weather stations in the same region. Initially, data from 2015 to 2017 was provided in hourly resolution. Then, during the final round, data for 2018 was provided in six subsequent iterations, which serve as test blocks for the model evaluation and as basis for the hyperparameter optimisation.

4.1 | Exploratory data analysis

The given data set is structured as an hourly time series. Each row has a timestamp containing a year, a date and hour. The feature variables consist of the weather data columns T1 up to T6 and the timestamps. When analysing the weather data, it becomes visible that the pairwise correlation between the columns is extremely high and in no case smaller than 0.95 which fits to the assumption that the weather stations are located close together.

For the load of the LDCs, three different target variables, LDC1, LDC2 and LDC3 are given. No unit of measurement is provided for the target variables. The respective loads of the LDCs are forecasted separately. In Figure 4, the distribution of the values of the target variables is depicted. We can observe that the loads of the different LDCs vary significantly in magnitude, with LDC3 having by far the largest loads. Despite these differences in magnitudes, the LDC columns are highly correlated. The lowest pairwise correlation observed is 0.91. As there still are variations between the different LDC load profiles, predicting and evaluating multiple LDCs can be considered some form of additional cross-validation of the model.

Apart from the timestamps, weather features are the only feature variables that are initially provided. Figure 5 depicts the relationship between the observed average LDC loads and the average temperature features. The relationship between temperature and load appears to be non-linear, with both high and low temperatures being associated with a high load. This indicates the use of electricity for both heating when cold temperatures occur as well as cooling when the temperature is high. We can underline that observation also from a statistical point of view: the overall Pearson correlation between the average temperature measurements and average LDC load measurements is quite low at 0.063. However, when we only regard all observations during temperature measurements below 60, the correlation is strongly negative at −0.87: the lower the temperature, the higher the loads. When we only regard the remaining observations at temperature measurements above 60, the correlation amounts to 0.87.

On the described data set, feature engineering following the methodology introduced in Section 3.2 was performed. This increased the number of input features to over 350. 16 of those features are related to the timestamp, that is, cyclical features, while the remaining features constitute various transformations of the respective temperature features.

4.2 | Train-test splits

The train-test splits in this work are based on the iterations of the final round of the BigDEAL Challenge 2022. As training data, all observations preceding the first date of the respective test set are used. Notably, the time frames of the test sets vary in length, requiring forecasts ranging up to 3 months ahead. We

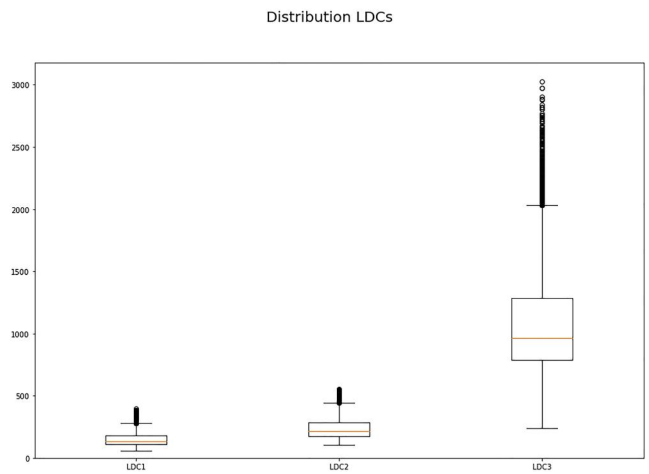

FIGURE 4 Distribution of Local Distribution Company (LDC) loads, which serve as target variables.

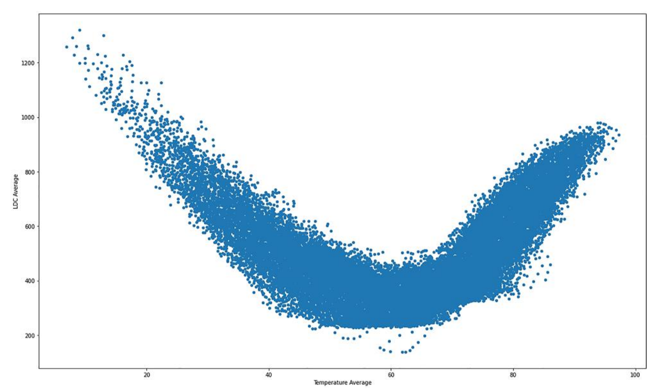

FIGURE 5 Relation between average temperature measurements and LDC loads.

consider six different test sets, four of which have a length of 2 months, while the others have a length of 1 month and 3 months, respectively. All test sets are in the year 2018. The exact time frames are depicted in Table 3.

4.3 | Model variations

In the following, we compare different variations of the XGBoost model, as described in our methodology. We compare our results with a naïve benchmark model that simply takes the day-before peak time as forecast, as in ref. [61]. In Table 4, the setup of the different models is shown. First, we introduce a baseline, regression-based XGBoost model called *XGBP**, which is only trained with given data, without any feature engineering or hyperparameter tuning. Second, we introduce the baseline XGBoost model *XGBP*, which is trained with enriched feature-engineered data, but only with standard parameters instead of Bayesian-optimised parameters per test. Third, we add to the *XGBP* model the Bayesian-hyperparameter optimisation, which yields model *XGBPH*. The first three models are all delivering a general load forecast for every timestep in the respective test set, from which the peak time forecast is inferred, as described in the Methodology. The *XGBPH* model served as a model for the daily peak load forecast and the overall load forecast during the BigDeal Peak Time challenge. For the daily peak load forecast, the model achieved an overall fourth rank amongst all competitors.

The two last models, *XGBR* and *XGBRH* are based on the previously introduced Learning-to-Rank XGBoost algorithm, which uses as input ranks of loads instead of absolute loads and which yields a forecast of daily ranks of loads. The *XGBRH* utilises Bayesian-optimised hyperparameters.

4.4 | Bayesian-optimised hyperparameters

As described in our methodology, we conduct a Bayesian hyperparameter optimisation, based on test data of 2017, for every LDC. In Table 5, we show the resulting hyperparameters for the *XGBPH* and *XGBRH* models for each LDC. We can observe that within the respective models, hyperparameters tend to go in the same direction. However, comparing the general load XGBoost model *XGBPH* and the Learning to Rank *XGBRH* model, we see significant differences. The max depth is around 5 for every general LDC model, while it is between 8 and 9 for the ranked models. The learning rate, as well as the subsample, are lower for the ranked models.

5 | RESULTS

We evaluate the different model variations, based on the six test blocks, for all three LDCs. For every scenario, we evaluate the peak time forecasting performance according to the Accuracy, the Mean Absolute Error (MAE) and BigDeal Peak Time Metric (BPDM).

In Figure 6, the actual load of LDC3 for an exemplary day in the first test set is depicted alongside the forecasts of the three general load prediction models *XGBP**, *XGBP* and *XGBPH*. We can observe that all three models roughly match the shape of the daily load curve, with two peaks, one in the morning and one in the evening. The higher peak lies in the evening at 23:00. The plot shows a tendency that is later also confirmed in absolute results: *XGBP**, the model without an enriched, feature-engineered data set, has difficulties in forecasting accurate absolute load values, whereas the two remaining models match the load pattern better. From each general load forecast, the highest

TABLE 3 Test sets used in the case study.

Test set no.	Start date	Length (months)
1	2018/01/01	2
2	2018/03/01	3
3	2018/06/01	2
4	2018/08/01	1
5	2018/09/01	2
6	2018/11/01	2

TABLE 5 Hyperparameters for XGBPH and XGBRH and respectively LDC1/2/3.

Hyperparameter	XGBPH	XGBRH
max_depth	5/5/5	8/9/8
learning_rate	0.15/0.11/0.07	0.019/0.03/0.046
Subsample	0.97/0.96/0.95	0.53/0.63/0.54
min_child_weight	3.58/3.49/3.64	1.63/4.81/4.36
colsample_bytree	0.77/0.72/0.80	0.52/0.82/0.62

TABLE 4 Model variations investigated in this study.

Model	Feature engineering	Regression-based	Rank-based	Hyperparameter
Baseline (day-before peak time)	No	No	No	No
XGBP*	No	Yes	No	No
XGBP	Yes	Yes	No	No
XGBPH	Yes	Yes	No	Yes
XGBR	Yes	No	Yes	No
XGBRH	Yes	No	Yes	Yes

forecasted load is inferred as peak time forecast. Here, only the hyperparameter optimised model *XGBPH* manages to forecast the actual peak at 23:00 accurately.

In contrast to the general load prediction models, the ranked models are not delivering hourly load forecasts, from which the point of time of the highest load is inferred as peak time. Hence, they are first plotted in Figure 7, where for every day in the first test period from January to February 2018, the respective peak times and peak time forecasts are plotted for LDC3. The plot follows a certain colour scheme: real peak times are plotted in green, Learning to Rank-based forecasts are plotted in tones of blue and general load forecasts are plotted in tones of red and velvet. The hyperparameter-tuned models *XGBPH* and *XGBRH* are plotted with higher opacity. First, we can observe that, in general, all forecasting models deliver a solid performance, mostly forecasting the peak time at least at hours around the peak, if not predicting it correctly. We assume that one reason for this is the high aggregation level of local distribution companies, as well as the high data quality, and that the training data covers multiple years. We can also observe the tendency of peaks either occurring in the morning hours around 9:00 or in the evening hours around 21:00 o'clock. We note that all XGBoost-based forecasting models manage quite well to forecast the peak times, even when there

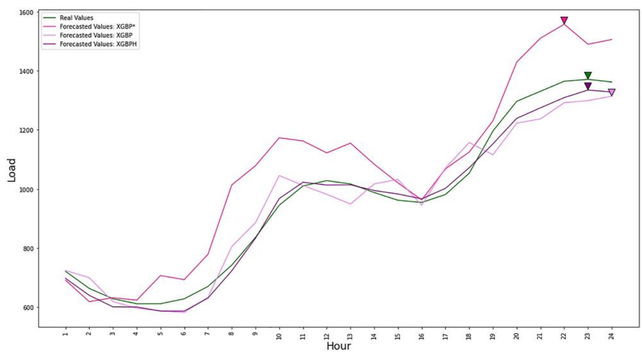

FIGURE 6 Exemplary load forecast with XGBP*, XGBP and XGBPH on the 13 January 2018, for LDC3. Inferred peak times marks with triangles.

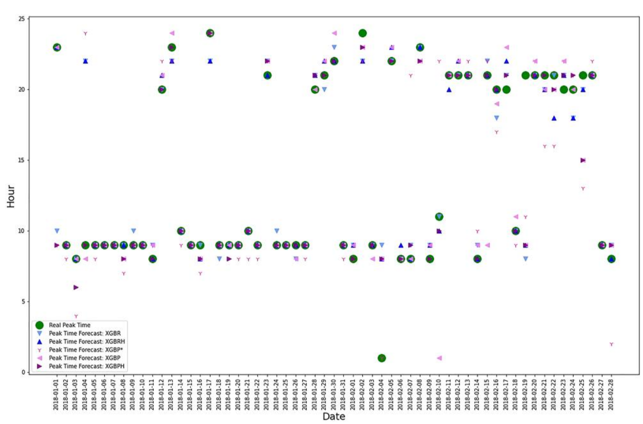

FIGURE 7 Exemplary peak time forecasts and real peak times in first test block for LDC3.

is a switch from periods of morning peaks to evening peaks. Using a recency-based model that always utilizes the day-before or week-before peak time for our forecast would lead in case of switches from morning-peaks to evening-peaks to significant losses in the MAE metric and accuracy. We also note that the Bayesian hyperparameter-optimised models are consistently predicting the peak time more accurately than the models with standard parameters. The worst performing XGBoost-based model is the one without a feature-engineered data set and without hyperparameter tuning, *XGBP**.

In Table 6, the accuracies for the different models are depicted, respectively for each LDC. First, we can observe that all models with the feature-engineered data are outperforming the model with the base data set, *XGBP**, by far. Second, both Bayesian-optimised models yield the best accuracies on average.

The same picture occurs when analysing the resulting MAEs in Table 7. On average, the Hyperparameter-optimised models outperform the models without hyperparameter optimisation. For LDC1, the *XGBRH* models yield the best MAEs on average; for LDC2 and LDC3, the *XGBPH* yields the best results. All MAE values for the models based on the feature-engineered data set are around 1, which can be interpreted as a mean deviation of the forecasted peak time from the true peak time of 1 hour.

Similarly, for the BDPM in Table 8, the Bayesian hyperparameter-tuned models mostly outperform the standard models, and *XGBPH* delivers the best results for LDC2 and LDC3, while the *XGBRH* model delivers the best BDPM results for LDC1. The average monthly BDPM values obtained through the *XGBPH* model are significantly better than the ones obtained with the *XGBP** model ($p = 0.016$). Whereas the average monthly BDPM results achieved by the *XGBRH* model are not significantly different than the ones achieved through the *XGBRH* model ($p = 0.90$).

This observation underlines two integral findings in our study. First, the peak time forecasting quality is significantly increased by our feature engineering process and the Bayesian hyperparameter optimisation. Overall, our *XGBPH* and *XGBRH* models have achieved an exceptional level of peak time forecasting quality. With a Mean Absolute Error of just 1 h, our performance is by far superior to the baseline case, where the day-before peak time is used for the forecast, resulting in a Mean Absolute Error of approximately 3 h. Second, transforming the actual load values to ranks of loads and employing a Learning to Rank XGBoost model does not significantly lowers the peak time forecasting quality compared to the well-performing, regular XGBoost model with feature engineering and hyperparameter optimisation, from which peak times are inferred. Thereby, we show that a more privacy-preserving peak time forecasting approach does not necessarily negatively influence the overall forecasting quality. Nonetheless, we note that the information on the ranks of loads still contains some information about the load data providers and could be used to identify them. However, industrial customers could be more open to sharing ranks of loads instead of actual load values with the grid operator since they do not contain information about machine utilisation and company activity.

TABLE 6 Accuracy in percent per test set for all considered Models for LDC1/LDC2/LDC3.

Test set	Baseline	XGBP*	XGBP	XGBPH	XGBR	XGBRH
1	27/34/36	36/42/37	36/51/47	46/69/68	41/59/54	58/69/61
2	31/47/36	38/29/35	44/53/58	52/66/54	58/51/55	57/54/55
3	44/45/32	41/52/34	46/56/34	52/67/61	59/69/67	59/77/69
4	41/52/32	32/32/35	39/58/58	55/55/71	52/63/68	48/74/58
5	44/43/51	39/39/48	41/54/39	49/54/49	52/46/52	48/46/51
6	30/31/36	30/31/30	36/33/49	41/48/56	38/39/46	43/46/53
Average	36/42/37	36/38/37	41/51/48	49/60/60	50/55/57	52/61/58
Std	7.01/7.3/6.4	3.90/7.93/5.56	3.34/8.27/8.92	4.60/7.86/7.73	7.93/10.23/6.30	6.09/12.48/5.96

TABLE 7 MAE per test set for all considered Models for LDC1/LDC2/LDC3.

Test set	Baseline	XGBP*	XGBP	XGBPH	XGBR	XGBRH
1	4.85/4.86/5.29	2.23/1.49/1.85	1.88/1.46/1.31	1.29/1.46/0.90	1.69/0.78/1.22	1.14/0.75/0.93
2	4.32/3.54/3.64	2.07/2.37/1.26	1.25/1.36/0.97	1.26/1.02/0.72	1.13/1.60/1.22	1.20/1.70/0.95
3	1.02/0.95/1.02	0.79/0.74/0.87	0.80/0.54/0.79	0.65/0.41/0.49	0.54/0.46/0.41	0.51/0.36/0.36
4	1.13/0.68/1.12	1.16/0.84/1.03	0.84/0.45/0.55	0.68/0.55/0.35	0.81/0.48/0.42	0.74/0.35/0.65
5	2.02/1.98/2.14	1.57/1.79/1.85	1.75/0.56/1.67	1.21/0.67/1.59	1.62/1.36/1.80	1.31/1.16/1.61
6	5.54/4.44/5.26	2.18/2.22/2.16	2.39/1.54/1.69	2.08/1.59/1.08	1.86/1.86/1.39	2.02/1.61/1.26
Average	3.10/2.72/3.07	1.67/1.52/1.50	1.49/1.09/1.16	1.16/0.95/0.86	1.27/1.09/1.08	1.15/0.99/0.97
Std	1.76/1.63/1.78	0.54/0.71/0.47	0.57/0.45/0.43	0.41/0.48/0.41	0.48/0.55/0.51	0.30/0.54/0.41

TABLE 8 BDPM per test set for all considered Models for LDC1/LDC2/LDC3.

Test set	Baseline	XGBP*	XGBP	XGBPH	XGBR	XGBRH
1	4.19/4.36/4.25	2.24/1.53/1.93	1.86/1.46/1.47	1.41/1.47/0.98	1.78/0.90/1.32	1.17/0.81/1.00
2	4.11/3.33/3.52	2.58/2.58/1.74	1.56/1.37/1.25	1.43/0.97/0.88	1.27/1.53/1.32	1.36/1.61/1.14
3	1.72/1.55/1.78	1.13/1.16/1.18	1.22/0.74/0.95	0.98/0.54/0.59	0.82/0.64/0.54	0.67/0.56/0.43
4	2.03/1.03/1.84	1.97/1.06/1.52	1.26/0.52/0.74	1.06/0.71/0.48	0.96/0.71/0.58	1.16/0.52/1.00
5	2.74/2.18/2.26	2.07/1.78/2.00	1.90/0.72/1.80	1.52/0.92/1.64	1.87/1.46/1.85	1.61/1.33/1.69
6	4.77/4.27/4.60	2.48/2.69/2.38	2.14/1.86 1.64	2.07/1.87/1.08	1.80/2.11/1.56	1.90/1.80/1.36
Average	3.26/2.79/2.95	2.07/1.80/1.79	1.65/1.11/1.31	1.41/1.08/0.94	1.42/1.22/1.20	1.31/1.11/1.10
Std	1.16/1.29/0.18	0.47/0.64/0.37	0.34/0.48/0.37	0.36/0.46/0.37	0.52/0.38/0.48	0.39/0.50/0.38

6 | FEATURE IMPORTANCES

In the previous section, we show that our feature engineering process reaches significant accuracy improvements. Hence, we are interested in investigating the average overall feature importances for the *XGBPH* and *XGBRH* models through the XGBoost feature weights. The XGBoost feature importance weight can be explained as the number of times that a certain feature is used in the trees of the model. The weight is then calculated as a share of the sum of all feature weights [62].

In Figure 8, the averaged feature importance weights are depicted for the *XGBPH* model. One striking observation is that out of the 15 most important features, 14 features are rolling averages of temperatures instead of the temperature measurements themselves. As mentioned before, we assume that the reason for the high importance of temperature rolling averages is the thermal inertia of buildings. Moreover, we observe that the most important rolling temperature features are those which cover time spans of three to 5 h. We can also see that some temperature measurements are relatively more important for the models of the respective LDCs. For instance, $T5$ temperature measurements seem to be more relevant for LDC2, while $T1$ and $T2$ measurements are relatively more important for LDC1. This might be connected to the distance of the temperature measurement stations to the respective LDCs: the closer the measurement stations are to the LDCs,

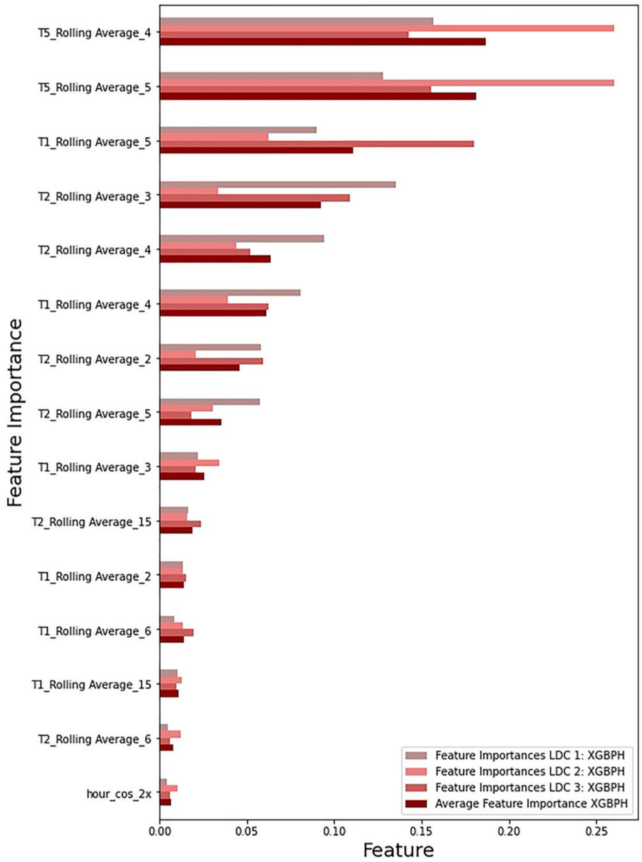

FIGURE 8 Averaged XGBoost feature importances for XGBPH models.

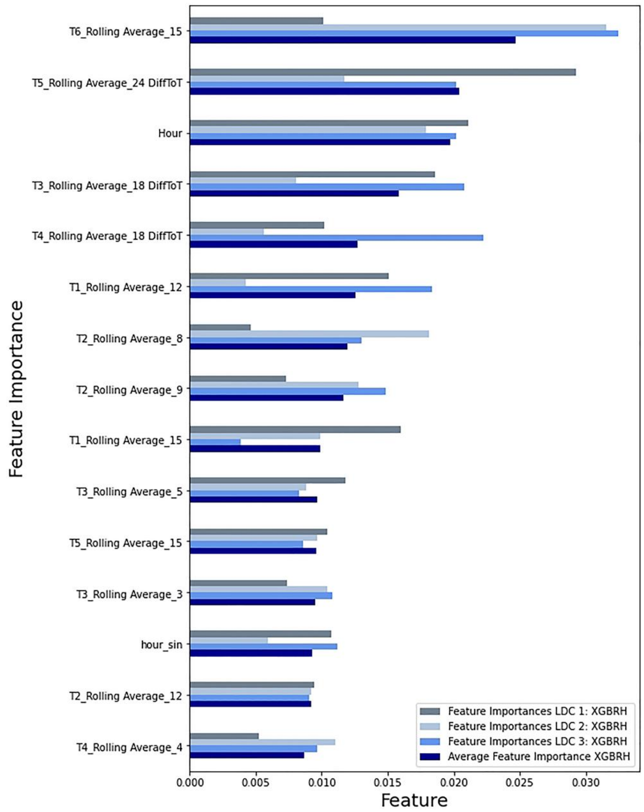

FIGURE 9 Averaged XGBoost feature importances for XGBRH models.

the more relevant the measurements are likely to be. We see this point especially relevant in light of our previous observation of the strong negative correlation of loads and temperature for lower temperatures and the strong positive correlation for higher temperatures. In addition, it underlines the relevance of considering appropriate temperature measurements in peak time and peak load forecasting tasks, in settings where the temperature is connected to the respective loads. The 15th most important feature is the 2× cosine of the hour, which supports the claim of the authors in refs. [7, 45] that it is reasonable to calculate the sine and cosine of cyclical features.

For comparison, we depict the feature importance weights of the *XGBRH* model in Figure 9. Again, we can see the high feature importance of rolling average-related features. However, for the Learning to Rank-based models, the important rolling average features cover longer time spans of up to 15 h. Furthermore, we can see the high feature importance of the "Difference of Temperature to Rolling Average" features, especially for the 24-h rolling average. High values of this feature indicate temperature peaks, which could lead to its high importance in the Ranked XGBoost model. Furthermore, we can observe relatively lower feature importances per feature, indicating that a wider array of features is used in the Learning to Ranked XGBoost trees.

7 | CONCLUSION

This paper offers a novel privacy-shielding approach to the peak time forecasting problem for local distribution companies by leveraging the Learning to Rank XGBoost algorithm. The Learning to Rank model is based on ranks of loads instead of absolute magnitudes of loads, requiring less confidential data. To analyse the accuracy of our approach, we conducted a case study in the context of the BigDEAL load forecasting challenge, where the peak times of three LDCs had to be forecasted. Furthermore, we conducted extensive feature engineering and selected model parameters through a Bayesian hyperparameter optimisation. Finally, we analyse the importance of the respective engineered features.

We show that the hyperparameter-tuned Learning to Rank XGBoost model delivers the highest average accuracy for two LDCs and the highest MAE and Big Deal Peak Time Metric for one LDC. For the remaining cases, the hyperparameter-tuned general load prediction model, which serves as a baseline in this work and achieved the fourth rank for the peak load forecasting track of the BigDEAL challenge, delivers the best results. Furthermore, we show that all XGBoost-based models are significantly outperforming a day-before recency-based benchmark model, thereby highlighting the value of XGBoost models for peak time forecasting. Also, we show a strong increase in forecasting accuracy by adding additional features, such as rolling averages of temperature measurements. Future works in this field should apply the rank-based forecasting approach to other methods, such as neural networks, and compare the results with the XGBoost-based Learning to Rank model introduced in this study. Leveraging neural network models, such as recursive neural networks or convolutional neural networks, might show superior performance in the peak time ranking task while still working with ranks instead of loads.

NOMENCLATURE

$\mathbb{P}$	accuracy
τ	feature
$BDPM$	BigDEAL Peaktime Metric
D	amount of days
d	day
dRA	difference to rolling average
G	relative feature change
MAE	Mean Absolute Error
N	amount of predictions
P_d^t	load at timestep t of day d
$P_{\max}$	peak load
r	rank
RA	rolling average
t	time step
T_n	amount of correctly predicted negative labels
T_p	amount of correctly predicted positive labels
$t_{d,P_{\max},pred}$	predicted time of peak load
$t_{d,P_{\max}}$	time of peak load
tf	time frame

AUTHOR CONTRIBUTIONS

Leo Semmelmann: Conceptualisation; methodology; resources; software; supervision; validation; visualisation; writing. **Oliver Resch**: Conceptualisation; methodology; resources; software; validation; visualisation; writing. **Sarah Henni**: Conceptualisation; methodology; resources; supervision; writing. **Christof Weinhardt**: Funding acquisition; supervision; writing.

ACKNOWLEDGEMENTS

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Open Access funding enabled and organized by Projekt DEAL.

CONFLICT OF INTEREST STATEMENT

The authors declare no conflicts of interest

DATA AVAILABILITY STATEMENT

The authors elect not to share their data publicly.

ORCID

Leo Semmelmann https://orcid.org/0000-0001-6735-9518

REFERENCES

1. Bunn, F.: Review of short-term forecasting methods in the electric power Industry. IEEE Trans. Power Syst. (1985)
2. Srinivasan, D., Lee, M.: Survey of hybrid fuzzy neural approaches to electric load. Forecasting 5(5), 4004–4008 (1995)
3. Groß, A., et al.: Comparison of short-term electrical load forecasting methods for different building types. Energy Informatics 4(S3), 1–16 (2021). https://doi.org/10.1186/s42162-021-00172-6
4. Lee, J., Cho, Y.: National-scale electricity peak load forecasting: traditional, machine learning, or hybrid model? Energy 239, 122366 (2022). https://doi.org/10.1016/j.energy.2021.122366
5. Goia, A., May, C., Fusai, G.: Functional clustering and linear regression for peak load forecasting. Int. J. Forecast. 26(4), 700–711 (2010). https://doi.org/10.1016/j.ijforecast.2009.05.015
6. Haida, T., Muto, S.: Regression based peak load forecasting using a transformation technique. IEEE Trans. Power Syst. 9(4), 1788–1794 (1994). https://doi.org/10.1109/59.331433
7. Semmelmann, L., Henni, S., Weinhardt, C.: Load forecasting for energy communities: a novel LSTM-XGBoost hybrid model based on smart meter data. Energy Informatics 5(S1), 1–21 (2022). https://doi.org/10.1186/s42162-022-00212-9
8. Olsthoorn, M., Schleich, J., Klobasa, M.: Barriers to electricity load shift in companies: a survey-based exploration of the end-user perspective. Energy Pol. 76, 32–42 (2015). https://doi.org/10.1016/j.enpol.2014.11.015
9. BigDEAL Challenge 2022 (2022). http://blog.drhongtao.com/2022/10/bigdeal-challenge-2022.html
10. Kong, W., et al.: Short-term residential load forecasting based on LSTM recurrent neural network. IEEE Trans. Smart Grid 10(1), 841–851 (2017). https://doi.org/10.1109/tsg.2017.2753802
11. Nti, I.K., et al.: Electricity load forecasting: a systematic review. J. Electr. Syst. Inf. Technol. 7(1), 1–19 (2020). https://doi.org/10.1186/s43067-020-00021-8
12. Tang, X., et al.: Short-term load forecasting using channel and temporal attention based temporal convolutional network. Elec. Power Syst. Res. 205, 107761 (2022). https://doi.org/10.1016/j.epsr.2021.107761
13. Wang, C., et al.: A transformer-based method of multienergy load forecasting in integrated energy system. IEEE Trans. Smart Grid 13(4), 2703–2714 (2022). https://doi.org/10.1109/tsg.2022.3166600
14. Juberias, G., et al.: A new ARIMA model for hourly load forecasting. IEEE 1, 314–319 (1999)
15. Lee, C.M., Ko, C.N.: Short-term load forecasting using lifting scheme and ARIMA models. Expert Syst. Appl. 38(5), 5902–5911 (2011). https://doi.org/10.1016/j.eswa.2010.11.033
16. Nie, H., et al.: Hybrid of ARIMA and SVMs for short-term load forecasting. Energy Proc. 16, 1455–1460 (2012). https://doi.org/10.1016/j.egypro.2012.01.229
17. Lin, J., et al.: Short-term load forecasting based on LSTM networks considering attention mechanism. Int. J. Electr. Power Energy Syst. 137, 107818 (2022). https://doi.org/10.1016/j.ijepes.2021.107818
18. Kucevic, D., et al.: Reducing grid peak load through the coordinated control of battery energy storage systems located at electric vehicle charging parks. Appl. Energy 295, 116936 (2021). https://doi.org/10.1016/j.apenergy.2021.116936
19. Kucevic, D., et al.: Peak shaving with battery energy storage systems in distribution grids: a novel approach to reduce local and global peak loads. Electricity 2(4), 573–589 (2021). https://doi.org/10.3390/electricity2040033
20. Syed, D., et al.: Investigation on optimizing cost function to penalize underestimation of load demand through deep learning modeling. IEEE, 1–7 (2021)
21. Amjady, N.: Short-term hourly load forecasting using time-series modeling with peak load estimation capability. IEEE Trans. Power Syst. 16(3), 498–505 (2001). https://doi.org/10.1109/59.932287
22. Dai, Y., et al.: Improving the Bi-LSTM model with XGBoost and attention mechanism: a combined approach for short-term power load prediction. Appl. Soft Comput. 130, 109632 (2022). https://doi.org/10.1016/j.asoc.2022.109632
23. Haida, T., Muto, S.: Regression based peak load forecasting using a transformation technique. IEEE Trans. Power Syst. 9(4), 1788–1794 (1994). https://doi.org/10.1109/59.331433
24. Saini, L.M.: Peak load forecasting using Bayesian regularization, resilient and adaptive backpropagation learning based artificial neural networks.

Elec. Power Syst. Res. 78(7), 1302–1310 (2008). https://doi.org/10.1016/j.epsr.2007.11.003
25. Haq, M.R., Ni, Z.: A new hybrid model for short-term electricity load forecasting. IEEE Access 7, 125413–125423 (2019). https://doi.org/10.1109/access.2019.2937222
26. Chin, J.X., et al.: Load forecasting of privacy-aware consumers. IEEE, 1–6 (2019)
27. Hou, S., et al.: A new privacy-preserving framework based on edge-fog-cloud continuum for load forecasting. IEEE, 1–8 (2020)
28. Dong, S., Liu, Y.: A Privacy-Preserving Electricity Theft Detection (PETD) Scheme for Smart Grid, pp. 44–61. Springer (2022)
29. Dwork, C.: Differential privacy. International colloquium on automata, languages, and programming, pp. 1–12 (2006)
30. Li, Q., et al.: Privacy-preserving gradient boosting decision trees. Proc. AAAI Conf. Artif. Intell. 34(01), 784–791 (2020). https://doi.org/10.1609/aaai.v34i01.5422
31. Le, N.K., et al.: Fedxgboost: Privacy-Preserving Xgboost for Federated Learning. *arXiv preprint arXiv:2106.10662* (2021)
32. Fernández, J.D., et al.: Privacy-preserving federated learning for residential short-term load forecasting. Appl. Energy 326, 119915 (2022). https://doi.org/10.1016/j.apenergy.2022.119915
33. Venkataramanan, V., Kaza, S., Annaswamy, A.M.: DER forecast using privacy-preserving federated learning. IEEE Internet Things J. 10(3), 2046–2055 (2022). https://doi.org/10.1109/jiot.2022.3157299
34. He, Y., et al.: Privacy-preserving and hierarchically federated framework for short-term residential load forecasting. IEEE Trans. Smart Grid, 1 (2023). https://doi.org/10.1109/tsg.2023.3268633
35. Tran, H.Y., Hu, J., Pota, H.R.: Smart meter data obfuscation with a hybrid privacy-preserving data publishing scheme without a trusted third party. IEEE Internet Things J. 9(17), 16080–16095 (2022). https://doi.org/10.1109/jiot.2022.3153043
36. Eibl, G., et al.: The influence of differential privacy on short term electric load forecasting. Energy Informatics 1(1), 93–113 (2018). https://doi.org/10.1186/s42162-018-0025-3
37. Chapelle, O., Chang, Y.: Yahoo! learning to rank challenge overview. PMLR, 1–24 (2011)
38. Vidović, K., et al.: Ranking network devices for alarm prioritisation: intrusion detection case study. IEEE, 1–5 (2021)
39. Syed, D., et al.: Smart grid big data analytics: survey of technologies, techniques, and applications. IEEE Access 9, 59564–59585 (2020). https://doi.org/10.1109/access.2020.3041178
40. Rajapaksha, D., Bergmeir, C.: LImref: Local interpretable model agnostic rule-based explanations for forecasting, with an application to electricity smart meter data. Proceedings of the AAAI Conference on Artificial Intelligence 36(11), 12098–12107 (2022). https://doi.org/10.1609/aaai.v36i11.21469
41. Bharadiya, J.: Machine learning in cybersecurity: techniques and challenges. Eur. J. Technol. 7(2), 1–14 (2023). https://doi.org/10.47672/ejt.1486
42. Herm, L.V., et al.: I don't get IT, but IT seems valid!. The Connection between Explainability and Comprehensibility in (X) AI Research (2021)
43. Kuhn, M., Johnson, K.: Feature Engineering and Selection. A Practical Approach for Predictive Models. CRC Press (2019)
44. Kanda, I., Veguillas, J.Q.: Data preprocessing and quantile regression for probabilistic load forecasting in the GEFCom2017 final match. Int. J. Forecast. 35(4), 1460–1468 (2019). https://doi.org/10.1016/j.ijforecast.2019.02.005
45. Gürses-Tran, G., Körner, T.A., Monti, A.: Introducing explainability in sequence-to-sequence learning for short-term load forecasting. Elec. Power Syst. Res. 212, 108366 (2022). https://doi.org/10.1016/j.epsr.2022.108366
46. Mayrink, V., Hippert, H.S.: A hybrid method using exponential smoothing and gradient boosting for electrical short-term load forecasting. IEEE, 1–6 (2016)
47. Natras, R., Soja, B., Schmidt, M.: Ensemble machine learning of random forest, AdaBoost and XGBoost for vertical total electron content forecasting. Rem. Sens. 14(15), 3547 (2022). https://doi.org/10.3390/rs14153547
48. Chen, T., Guestrin, C.: Xgboost: A scalable tree boosting system, pp. 785–794 (2016)
49. Liao, X., et al.: Research on short-term load forecasting using XGBoost based on similar days. IEEE, 675–678 (2019)
50. Abbasi, R.A., et al.: Short Term Load Forecasting Using XGBoost, pp. 1120–1131. Springer (2019)
51. Bardenet, R., et al.: Collaborative hyperparameter tuning. PMLR, 199–207 (2013)
52. Snoek, J., Larochelle, H., Adams, R.P.: Practical bayesian optimization of machine learning algorithms. Adv. Neural Inf. Process. Syst. 25 (2012)
53. Wu, J., et al.: Hyperparameter optimization for machine learning models based on Bayesian optimization. J. Electron. Sci. Technol. 17(1), 26–40 (2019)
54. Trierweiler Ribeiro, G., et al.: Bayesian optimized echo state network applied to short-term load forecasting. Energies 13(9), 2390 (2020). https://doi.org/10.3390/en13092390
55. Jin, X.B., et al.: Deep-learning forecasting method for electric power load via attention-based encoder-decoder with bayesian optimization. Energies 14(6), 1596 (2021). https://doi.org/10.3390/en14061596
56. Munem, M., et al.: Electric power load forecasting based on multivariate LSTM neural network using Bayesian optimization. IEEE, 1–6 (2020)
57. Bayesian optimpization PyPi package: bayesian-optimization 1.4.3 (2022)
58. Wang, Y., et al.: Short-term load forecasting of industrial customers based on SVMD and XGBoost. Int. J. Electr. Power Energy Syst. 129, 106830 (2021). https://doi.org/10.1016/j.ijepes.2021.106830
59. Massaoudi, M., et al.: A novel stacked generalization ensemble-based hybrid LGBM-XGB-MLP model for Short-Term Load Forecasting. Energy 214, 118874 (2021). https://doi.org/10.1016/j.energy.2020.118874
60. Liu, J., Brown, L.E.: Prediction of hour of coincident daily peak load, pp. 1–5 (2019)
61. Bracale, A., et al.: Short-term industrial load forecasting: a case study in an Italian factory. IEEE, 1–6 (2017)
62. Ma, J., et al.: Identification of the most influential areas for air pollution control using XGBoost and Grid Importance Rank. J. Clean. Prod. 274, 122835 (2020). https://doi.org/10.1016/j.jclepro.2020.122835

How to cite this article: Semmelmann, L., et al.: Privacy-preserving peak time forecasting with Learning to Rank XGBoost and extensive feature engineering. IET Smart Grid. 1–14 (2023). https://doi.org/10.1049/stg2.12137