

RESEARCH

Open Access



Load forecasting for energy communities: a novel LSTM-XGBoost hybrid model based on smart meter data

Leo Semmelmann*, Sarah Henni and Christof Weinhardt

From The 11th DACH+ Conference on Energy Informatics 2022 Freiburg, Germany. 15-16 September 2022

*Correspondence:
leo.semmelmann@kit.edu

Karlsruhe Institute of Technology,
Kaiserstraße 89-93,
76133 Karlsruhe, Germany

Abstract

Accurate day-ahead load forecasting is an important task in smart energy communities, as it enables improved energy management and operation of flexibilities. Smart meter data from individual households within the communities can be used to improve such forecasts. In this study, we introduce a novel hybrid bi-directional LSTM-XGBoost model for energy community load forecasting that separately forecasts the general load pattern and peak loads, which are later combined to a holistic forecasting model. The hybrid model outperforms traditional energy community load forecasting based on standard load profiles as well as LSTM-based forecasts. Furthermore, we show that the accuracy of energy community day-ahead forecasts can be significantly improved by using smart meter data as additional input features.

Keywords: Smart grids, Day-ahead forecasting, Energy communities, Long short-term memory neural networks, XGBoost

Introduction

Day-ahead load forecasting is an essential task for grid operators and utilities in modern smart power systems to optimize balancing groups and to match upcoming demand and supply. Currently, standard load profiles, which are provided by the German Federal Association of the Energy and Water Industry in every year, are widely used by grid operators and modelers to approximate energy consumption (Peters et al. 2020). However, sector-coupled smart grids require improved forecasting methods, since new large consumers, such as heat pumps and electric vehicles, add significant loads to residential households. In addition, intermittent renewable generation, especially from photovoltaic, changes traditional load patterns. In smart grids, more accurate forecasts could enable an improved management of emerging flexibility potentials, e.g., from battery storage, electric vehicles and heat pumps. The emergence of smart meters creates further possibilities in the field of day-ahead forecasting through the availability

of high-resolution load data on household level. The authors of Zufferey et al. (2016) show with smart meter data from over 10,000 households in Basel, Switzerland, that a higher number of smart meter load profiles increases the general prediction accuracy significantly.

Improving day-ahead load forecasts also plays a vital role for (smart) energy communities. Energy communities are an emerging concept in research and practice, where local communities are collectively managing and optimizing their electricity production and consumption, e.g., through peer-to-peer trading or the joint utilization of storage systems (Shrestha et al. 2019; Henni et al. 2021). The importance of energy communities has been recognized by the European Union who plans to promote and strengthen decentral structures and has introduced the concept of “Citizen Energy Communities” in the 2019 *Directive on common rules for the internal market for electricity* (Golla et al. 2020; European Parliament and Council of the European Union 2019). A central task in these energy communities will be the planning and management of flexibility potentials and electricity production. By improving community load forecasts, energy management can be improved, costs can be lowered and CO₂ emissions reduced (Wen et al. 2019; Grundmeier et al. 2014). While (day-ahead) load forecasting plays an important role on all levels of future smart grids, we specifically focus on energy communities in this work. A special feature of energy communities is their level of aggregation within a smart grid. In literature, energy communities typically consist of usually in between 2 to 500 households: in Coignard et al. (2021), communities between 2-95 households are analyzed, in Reijnders et al. (2020; Abadi et al. 2016) Dutch households are regarded, while Schlund et al. (2018) focuses on 500 distributed households within a network section. This makes (day-ahead) load forecasting of energy communities based on smart meter data a different task than in individual households or larger grid sections. In individual households, smart meter data is either available or not, and load profiles may differ significantly from one household to another. In energy communities, there is already some level of aggregation which means that standard load profiles could be applied here as a (naive) forecast. However, the level of aggregation is much lower than in the case of grid-level forecasts which can contain 10,000s of households (Zufferey et al. 2016). (Day-ahead) load forecasting in energy communities therefore deserves special attention, since the question arises whether smart meter data can be utilized strategically (e.g., by only installing smart meters in selected households) to improve load forecasts. This work thus aims at investigating the potential to improve day-ahead load forecasting of smart energy communities.

Recent research works like (Wang et al. 2019) have identified bi-directional Bi-directional long Short-Term Memory recurrent neural networks (Bi-LSTM) as suitable method to achieve high load forecasting accuracy. Although Bi-LSTM-based forecasts often enable high prediction accuracy in general, the forecasting of peak load hours and peak load quantities remains an important issue, as shown in Sarduy et al. (2016); Liu and Brown (2019). Previous works in the field consider the forecasting of peak loads and peak load hours as part of the overall forecasting process, instead of separating the forecasting of the general load pattern (e.g., through an LSTM) from the explicit forecasting of peak loads. Forecasting peak loads is especially important for grid operators that have to prevent possible congestion situations in the grid or at transformer stations (Kucevic

et al. 2021). Only a fraction of existing works in the load forecasting field incorporates smart meter data into the (LSTM-based) forecasting process. Furthermore, selection criteria for smart metered-households are rarely discussed (Haben et al. 2021; Kong et al. 2017; Ghiani et al. 2019).

In this work, we therefore contribute to the field of community load forecasting through two extensions of previous works. First, we demonstrate the improvements that can be achieved by incorporating smart meter data into day-ahead community load forecasts. We use the concept of feature permutation importance to identify the most important features for the training of a LSTM. This information could potentially be used to install smart meter infrastructure selectively by targeting the most relevant households for the community forecast. Second, we tackle the shortcomings regarding the incorporation of accurate peak load forecasting in previous works by proposing a hybrid bi-directional LSTM-XGBoost forecasting model. In the hybrid model, we deploy a LSTM which is suitable to accurately predict the general trend of aggregated community load. We then separately forecast peak load time and quantity with an XGBoost model using on smart meter data. Lastly, we combine the peak load forecast with the LSTM-based general forecast to obtain a holistic community load forecast. Also, cyclical type-of-day features, such as the *sin* and *cos* transformation of the hour, are engineered to further improve the forecast quality without requiring additional data as demonstrated in Haben and Giasemidis (2016).

We therefore aim to investigate (i) if smart meter data can improve existing LSTM load forecasting models of energy communities and (ii) whether the problem of insufficient peak forecasts can be tackled with a novel hybrid model. The contributions of this work are thus threefold:

- 1 A bi-directional LSTM-based model for the forecast of the aggregated load of an energy community using individual and aggregated smart meter data as input.
- 2 The identification of the most important forecast input features in terms of type-of-day data as well as smart meter data of individual households using feature permutation.
- 3 A novel hybrid LSTM-XGBoost approach is proposed to incorporate accurate peak load forecasting and to improve overall accuracy of existing day-ahead aggregated load forecasting methods.

The remainder of this study is structured as follows. The first section covers the theoretical background of LSTM-based day-ahead load forecasting and XGBoost. The second section describes the methodology of this study and additional feature engineering steps that were undertaken. The third section describes the underlying dataset and the setup of the case study in which we demonstrate the developed methodology. The fourth section gives an overview of the results, whereas the fifth section discusses the findings of the case study. The final section summarizes the results and gives an outlook on further research directions in the field.

Theoretical background

Day-ahead load forecasting has been a relevant topic in research for years. A traditional approach is the Autoregressive moving average (ARIMA) method, mostly combined with other methods like the lifting scheme (Lee and Ko 2011), generalised autoregressive conditional heteroscedasticity (Hor et al. 2006) or artificial neural networks (Dube et al. 2017). More recent works have shown the good applicability and performance of LSTMs for day-ahead forecasting problems (Kong et al. 2017). LSTMs, which were first introduced by Hochreiter and Schmidhuber (1997), are based on Recurrent neural networks (RNNs). RNNs are sequence-based networks that can establish temporal correlations between previous and current information. This makes RNNs suitable for load forecasting problems, since upcoming loads often depend on daily patterns and routines as well as past load data. In Bouktif et al. (2018), France's metropolitan electricity loads are forecasted with a combined model of LSTMs and genetic algorithms for feature selection and hyperparameter tuning. The forecasting error, compared with an ExtraTree model, can be reduced by over 20%. In Jiao et al. (2018), LSTMs are used to forecast the electricity consumption of 48 non-residential consumers. By using LSTMs, a Mean absolute percentage error (MAPE) in the amount of 22.45% is reached. In comparison, with the traditional ARIMA method only a MAPE of 35.87% is achieved. As stated in Bouktif et al. (2020), it is important to find the right combination of LSTM hyperparameters in order to achieve accurate load forecasting results.

Load forecasting in energy communities is a special form of day-ahead load forecasting due to the level of load aggregation. For instance, the authors of Coignard et al. (2021) evaluate energy community load forecasts from 2 to 95 households. Furthermore, in Coignard et al. (2021) the importance of peak-load hour forecasts is emphasized in energy communities, since through accurate forecasts the scheduling of battery storage systems and flexible loads can be optimized for high self-sufficiency rates.

Another recent development in machine learning is so called Extreme gradient boosting (XGBoost), which was introduced by Chen and Guestrin (2016). XGBoost is an efficient implementation of gradient boosting that is based on parallel tree learning and efficient proposal calculation and caching for tree learning. The XGBoost algorithm has found a wide variety of use cases, also in the context of energy systems research. In Zheng and Wu (2019), the framework is used for short-term wind power forecasting. In Wang et al. (2017), next month electricity consumption is forecasted through a hybrid wavelet transform and XGBoost model. First works have also combined XGBoost with day-ahead load forecasting models. For instance, in Wang et al. (2021), an adaptive decomposition method is used together with an XGBoost-based regression model to forecast loads of industrial customers in China and Ireland. The authors of Li et al. (2019) separately forecast day-ahead loads through an LSTM neural network and XGBoost. Subsequently, an error-reciprocal method is used to combine the forecasts. However, both methods are used for a general load forecast, instead of focusing the XGBoost forecast on peak loads. Previous works like Shwartz-Ziv and Armon (2022) have shown that XGBoost outperforms neural networks for regression and classification tasks on tabular data.

Several studies have shown that LSTM models are accurately capturing temporal dependencies but often underestimate peak values (Karimian et al. 2019; Feng et al.

2020). Hence, this study combines the LSTM day-ahead forecast, which generally depicts the temporal structure of the load, with a XGBoost forecast of peak load times and quantities. To our knowledge, no studies have pursued this approach so far.

In machine learning, feature importance measures help to better understand relevant inputs. A commonly used method for feature importance analysis is the permutation importance measure, which was introduced by Altmann et al. (2010). In this method, the decrease of prediction accuracy is measured after permuting input features. Thereby, a permutation importance score can be calculated for every feature to assess its importance for the model.

Building on these previous findings, we first develop a LSTM-based day-ahead forecast model and identify the most important input features in terms of easy-to-observe and smart-meter data using permutation importance. We then expand previous models by introducing an XGBoost model for forecasting both peak load time and quantity and combine the two approaches into one holistic hybrid model to improve overall accuracy of day-ahead aggregated load forecasts of energy communities.

Methodology

In this section, we describe our methodology for smart meter data-based LSTM forecasting of day-ahead aggregated community loads. An overview of the research framework of this study is depicted in Fig. 1. In the following, we describe each component of the framework in detail.

Input data and type-of-day features

In a first step, the underlying smart meter data is preprocessed to create additional input features and to create the aggregate load of all smart meters P_{agg} , which serves as target variable. The aggregate load P_{agg} at time t can be calculated by summing up every load $P_{n,t}$ of all N smart metered households: $P_{agg,t} = \sum_{n=1}^N P_{n,t}$.

As shown in Kanda and Veguillas (2019), adding additional type-of-day features to the underlying dataset can improve the general forecasting accuracy. Type-of-day features in this work include variables for the weekday, hour and month. To achieve periodicity for type-of-day variables, sinusoidal transformation is used as described in Haben and Giasemidis (2016). Also, a binary variable for weekends is added.

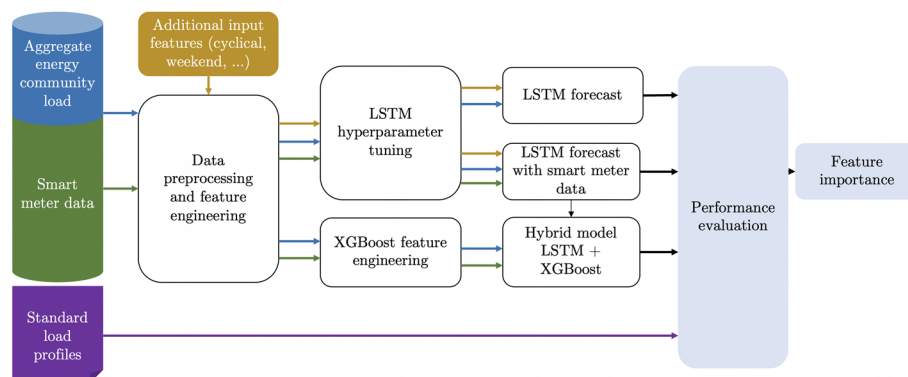


Fig. 1 Proposed research methodology framework. Arrows in different colors depict different input features for subsequent steps

Data preprocessing

For the use of LSTM neural networks, the input data has to be preprocessed first. Every input feature I can be seen as a sequence of data points for the past K timesteps, as stated in Eq. 1:

$$I = \{i_{t-K}, \dots, i_{t-2}, i_{t-1}\} \tag{1}$$

In our case, K represents the amount of timesteps per day in the underlying dataset. Due to the sensitivity of LSTMs to the data scale, all input vectors are normalized to the range of (0,1) by min-max-normalization. The input matrix X_d for the forecast of any day d in the dataset consists of all input features I :

$$X_d = \{I_1, I_2, \dots\} \tag{2}$$

LSTM model

LSTMs are a special form of Recurrent neural networks (RNN), which solves the problem of exploding and vanishing gradients by adding a memory cell and gate (Wang et al. 2019). Thereby, long-distance relationships between elements in sequence data can be processed. To create these temporal relationships, the LSTM defines and maintains a memory cell state over its life cycle. Three different types of timing modules exist in LSTMs: an input gate, a forget gate and an output gate. In turn, every timing module maintains its own memory cell and has its own task. The input gate is used to process incoming information, the forget gate decides about information retention of the historical cell state and the output gate processes outgoing information. The decision about information affecting the cell's state can be done selectively by using sigmoid activation functions. The output of the gates lies between 0 and 1. Thereby, a decision is made about the amount of information that is passed through the respective structure. A recent advance of LSTMs are Bidirectional long Short-Term Memory recurrent neural networks (Bi-LSTM), which can process both past and future information, whereas traditional LSTMs can only work with one-way transmission of information. Several works have shown that Bi-LSTM neural networks outperform traditional LSTMs in load forecasting problems (Wang et al. 2019; Atef and Eltawil 2020), hence they are preferred over traditional LSTMs in this work. The unfolded structure of a Bi-LSTM is depicted in Fig. 2.

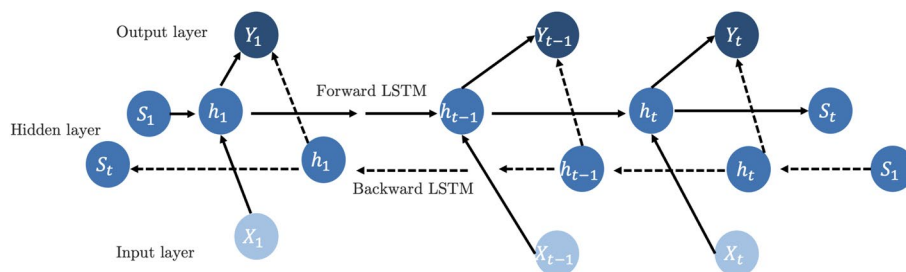


Fig. 2 Structure of Bi-LSTM network that both processes past and future information

The bi-directional LSTM layer in this study is followed by a dense layer, another bi-directional LSTM layer, two dense layers and a dropout layer to prevent overfitting (Tang et al. 2019).

LSTM hyperparameter tuning

To achieve a good combination of computational effort and accuracy, a randomized grid search is conducted for hyperparameter tuning, based on Wang et al. (2019). The parameters listed in Table 1 represent the parameter search space, 100 runs are conducted with new random combinations of hyperparameters. The parameters for the search space itself are defined based on existing studies that use LSTM neural networks for load forecasting (Kong et al. 2017; Muzaffar and Afshari 2019; Zheng et al. 2017; Bouktif et al. 2018; Jiao et al. 2018; Bouktif et al. 2020; Jahangir et al. 2020).

Feature importance

Since this paper also aims to improve the general understanding of LSTM neural networks for energy community forecasting, the importance of the respective input features is investigated. Therefore, the measure importance Permutation importance (PIMP) is used, which was introduced by in Altmann et al. (2010). The permutation feature importance metric is deployed in many load forecasting studies and is model-agnostic (Huang et al. 2016; Lahouar and Slama 2015). To evaluate the importance of a certain feature I through permutation importance, its values are randomly shuffled to create a permuted input vector I_{ξ} . Now, the decrease in prediction accuracy in terms of $MAPE_{I_{\xi}}$ is compared to the MAPE of the unpermuted baseline model, as stated in Eq. 3:

$$PIMP_I = MAPE_{I_{\xi}} - MAPE \quad (3)$$

A higher $PIMP_I$ means the model gets worse through a randomization of feature I , which indicates a higher feature importance.

XGB feature engineering

Previous studies on LSTM-based aggregated day-ahead load forecasting have shown improvements over alternative methods. However they are less well suited to predict varying peak load times and (extreme) peak quantities, as a time series forecast will always try to predict an expected value rather than extreme events. To improve the accuracy of peak load prediction within our day-ahead aggregated load forecast, we therefore

Table 1 In each run, a random combination of hyperparameters is tested

Hyperparameter	Value
Past input timesteps as multiple of K	1, 2, 3
Batch size	32, 64
Epochs	50, 75, 100
Steps per epoch	75, 100, 125, 150, 200
Learning rate	0.1, 0.01, 0.005, 0.001
Number of units in LSTM-layer	20, 70, 130, 200, 260
Number of units in dense layers	20, 70, 130, 200, 260

rely on a classification approach that specifically predicts peaks. We divide the task of peak load forecasting into two sub-tasks: predicting the time and quantity of the next day’s peak load. Therefore, two XGBoost models are separately trained to forecast peak load quantities and times. For the model input, the whole data set of smart meter loads is reduced to daily load indicators. Each day d is depicted as vector of K consecutive timesteps t , thus $d = [t_1, \dots, t_K]$.

The two target variables $t_{P_{max,d,agg}}$ and $P_{max,d,agg}$ are calculated for every day d . In Eq. 4, the peak load $P_{max,d,agg}$ is obtained by getting the highest load $P_{t,d,agg}$ on day d :

$$P_{max,d,agg} = \text{Max}(P_{1,d,agg}, \dots, P_{K,d,agg}) \tag{4}$$

In Eq. 5, the peak time $t_{P_{max,d,agg}}$ is obtained by getting the time step of the previously determined $P_{max,d,agg}$:

$$t_{P_{max,d,agg}} = t(P_{max,d,agg}) \tag{5}$$

Then, for every day d a range of statistical measures is calculated, as noted in Table 2, based on the previous day $d - 1$ or up to 21 previous days $d - 1, \dots, d - 21$. In detail, maximum loads, minimum loads, mean loads, median loads and load standard deviations are regarded. The subscript n denotes input features that are derived for each individual household in the respective community, whereas the subscript agg denotes that the input features are derived based on the aggregated energy community load. For the peak time $t_{P_{max}}$ forecasting model, also the peak times of the 20 smart metered households with the largest annual energy consumption, N_{large} , are regarded for the past 21 days. Only the 20 largest households are regarded due to computational limitations.

XGBoost model

XGBoost was introduced by in Chen and Guestrin (2016). The approach builds upon gradient tree boosting algorithms, which are extended by a second-order Taylor expansion for a faster optimization process and to avoid overfitting. Previous works have

Table 2 Features for XGBoost datasets for peak load P_{max} and peak time $t_{P_{max}}$ forecasting for each day d , based on values from previous day ($d - 1$) or previous days. Features are either based on the aggregated energy community load (agg) or smart meter data of households N

Feature	$t_{P_{max}}$ input data	P_{max} input data
$P_{max,d-1,n} \quad \forall n \in N$	Yes	Yes
$P_{min,d-1,n} \quad \forall n \in N$	Yes	Yes
$P_{mean,d-1,n} \quad \forall n \in N$	Yes	Yes
$P_{median,d-1,n} \quad \forall n \in N$	Yes	Yes
$P_{\sigma,d-1,n} \quad \forall n \in N$	Yes	Yes
$t_{P_{max,d-1,agg}} \dots t_{P_{max,d-21,agg}}$	Yes	No
$t_{P_{max,d-1,n}} \dots t_{P_{max,d-21,n}} \quad \forall n \in N_{large}$	Yes	No
$P_{max,d-1,agg} \dots P_{max,d-21,agg}$	No	Yes
$P_{max,d-1,agg}$	No	Yes
$t_{P_{max,d-1,agg}}$	Yes	No

Input features include day-before maximum loads, minimum loads, mean loads, median loads and load standard deviation ($P_{max,d-1,n}, P_{min,d-1,n}, P_{mean,d-1,n}, P_{median,d-1,n}, P_{\sigma,d-1,n}$). For the peak time forecast, especially day-before peak load times $t_{P_{max,d-1}}$ are relevant

shown that the XGBoost algorithm can be well applied for load forecasting tasks. For instance, the authors of Wang et al. (2021) apply XGBoost to the load forecasting of industrial customers in Ireland and China.

XGBoost is based on an ensemble of Classification and Regression Tree (CART), which are used as weak learners. Weak learners are usually performing slightly better than random guesses in classification and prediction tasks and are modified over the iterations of the optimization process to form a well-performing ensemble model. The prediction \hat{y}_i for sample i is defined by Eq. 6,

$$\hat{y}_i = \sum_{m=1}^M f_m(i) \tag{6}$$

where M is the number of CART, and $f_m(i)$ is the forecasted value for the sample i in tree m . The underlying objective function *Obj* is introduced in Eq. 7:

$$\text{Obj} = \sum_{i \in I_j} l(y_i, \hat{y}_i) + \sum_{m=1}^M \Omega(f_m) \tag{7}$$

where I_j is the set of all samples in leaf j and l is the second-order loss function that measures the difference between predicted value \hat{y}_i and actual value y_i . The regularization term $\Omega(f_m)$, as defined in Eq. 8, consists of the number of leaf nodes T . The score of leaf j is measured by w_j . γ and β are parameters of the tree:

$$\Omega(f_m) = \gamma T + \frac{1}{2} \beta \sum_{j=1}^T w_j^2 \tag{8}$$

The structure of the CART and exact split points are determined by the quadratic objective function, which is simplified through the aforementioned second-order Taylor expansion, as noted in Eq. 9:

$$\text{Obj} = \sum_{j=1}^T \left[\left(\sum_{i \in I_j} g_i \right) w_j + \frac{1}{2} \left(\sum_{i \in I_j} h_i + \beta \right) w_j^2 \right] + \gamma T \tag{9}$$

where g_i is the first derivative of the loss function and h_i is the second derivative. The quadratic Eq. 9 is solved to obtain the leaf node score w_j^* :

$$w_j^* = - \frac{\sum_{i \in I_j} g_i}{\sum_{i \in I_j} h_i + \beta} \tag{10}$$

As a scoring function $\tilde{\mathcal{L}}^{(t)}(q)$, Eq. 11 is introduced to evaluate the quality of the tree structure q :

$$\tilde{\mathcal{L}}^{(t)}(q) = - \frac{1}{2} \sum_{j=1}^T \frac{\left(\sum_{i \in I_j} g_i \right)^2}{\sum_{i \in I_j} h_i + \beta} + \gamma T \tag{11}$$

Finally, to determine the tree structure and splitting decisions \mathcal{L}_{split} , a greedy algorithm is used that starts with one leaf and then iteratively adds branches, as noted in Eq. 12:

$$\mathcal{L}_{split} = \frac{1}{2} \left[\frac{(\sum_{i \in I_L} g_i)^2}{\sum_{i \in I_L} h_i + \beta} + \frac{(\sum_{i \in I_R} g_i)^2}{\sum_{i \in I_R} h_i + \beta} - \frac{(\sum_{i \in I} g_i)^2}{\sum_{i \in I} h_i + \beta} \right] - \gamma \tag{12}$$

where I_L are sample sets of left nodes and I_R are sample sets of right nodes. Given that $I = I_L \cup I_R$, the loss reduction after a split is denoted by \mathcal{L}_{split} . Through Eq. 12, possible split candidates are evaluated. For a more detailed explanation of the XGBoost algorithm, we refer to Chen and Guestrin (2016).

Based on the previously introduced approach, two separate XGBoost models are trained to forecast $t_{P_{max,d,agg}}$ and $P_{max,d,agg}$. Since forecasting $t_{P_{max,d,agg}}$ is a classification problem, the Receiver Operating Characteristic Curve (ROC AUC) is used as optimization metric. For the forecasting model of $P_{max,d,agg}$, the Mean squared error (MSE) is used as optimization metric, since this is a regression task.

The parameters of the XGBoost model are also determined through a hyperparameter search, based on parameters from Zheng et al. (2017); Wang et al. (2021); Li et al. (2019). The parameter search space is described in Table 3. Parameters are separately determined for the peak time and peak load model. In total, 1000 runs are conducted per model.

Hybrid LSTM-XGB model

After forecasting $t_{P_{max,d,agg}}$ and $P_{max,d,agg}$ with the XGBoost model, the results have to be incorporated into the LSTM forecast, which is a vector of K forecasted loads \hat{P}_t : $\{\hat{P}_1, \dots, \hat{P}_k, \dots, \hat{P}_K\}$. For readability, we simplify the outputs of the XGBoost prediction as $t_{XGB} = t_{P_{max,d}}$ and $P_{XGB} = P_{max,d}$.

The most straightforward approach would be to simply replace the value of the original LSTM load forecast, \hat{P}_k at time step $k = t_{XGB}$ with the predicted peak load quantity \hat{P}_{XGB} . However, this bears the risk that in case the peak load time has not been predicted correctly, the prediction will extremely overestimate the true load. We therefore scale down the predicted peak load by a parameter $\lambda \in [0,1]$. In our case, we set $\lambda = \frac{1}{2}$ and calculate the new peak value $\hat{P}_{t_{XGB}}$ according to Eq. 13.

Table 3 In each run, a random combination of hyperparameters is tested

Hyperparameter	Space	Distribution
N estimators	[40,1000]	Randint
Max depth	[1, 100]	Randint
Learning rate	[0.01, 0.59]	Uniform
Subsample	[0.3, 0.6]	Uniform
Colsample bytree	[0.5, 0.4]	Uniform
Min child weight	[0.05, 0.1, 0.02, 1, 2, 3, 4]	None
Gamma	[0,0.5,2,10]	None

The number of *nestimators* and the *maxdepth* is determined by drawing random integers. *Minchildweight* and *gamma* are determined by drawing from a set of pre-determined values. The remaining hyperparameters are drawn from uniform distributions

$$\hat{P}_{t_{XGB}} = \hat{P}_t + \frac{1}{2}(\hat{P}_{XGB} - \hat{P}_t) \quad (13)$$

Since load peaks are usually patterns of subsequent, elevated loads, in Eq. 14 also the previous load $\hat{P}_{t_{XGB}-1}$ and subsequent load $\hat{P}_{t_{XGB}+1}$ are adapted by a quarter of the difference between the XGB and LSTM-based peak load forecast:

$$\hat{P}_{t_{XGB}-1} = \hat{P}_{t-1} + \frac{1}{4}(\hat{P}_{XGB} - \hat{P}_t) \quad \hat{P}_{t_{XGB}+1} = \hat{P}_{t+1} + \frac{1}{4}(\hat{P}_{XGB} - \hat{P}_t) \quad (14)$$

Thereafter, the adjusted values are inserted into the forecasting vector:

$$\{\hat{P}_1, \dots, \hat{P}_{t_{XGB}-1}, \hat{P}_{t_{XGB}}, \hat{P}_{t_{XGB}+1}, \dots, \hat{P}_k, \dots, \hat{P}_K\} \quad (15)$$

Performance evaluation

Finally, the forecasting performance is evaluated by the most commonly used metric in day-ahead forecasting, the Mean absolute percentage error (MAPE). The MAPE divides the sum of percentual deviations from the forecasted loads P_{ft} by the actual loads P_{rt} with the number of time steps K , as described in Eq. 16:

$$\text{MAPE} = \frac{1}{K} \sum_{t=1}^K \left| \frac{P_{ft} - P_{rt}}{P_{rt}} \right| \times 100 \quad (16)$$

As a second metric, the Root-mean-squared error is used, which is the root of the mean squared error from P_{ft} and P_{rt} , as denoted in Eq. 17:

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n |P_{ft} - P_{rt}|^2} \quad (17)$$

In this work, the MAPE is calculated for all forecasted day-ahead loads as well as only for the highest forecasted load, averaged over all days in the test data set. For the general load forecast, also the RMSE metric is regarded. Through this, we can assess both overall load forecast quality and the peak load forecasting capabilities of our model.

In order to achieve more stable and unbiased results, the dataset is further split with a twelvefold-cross-validation, where every split represents 30 days (Burman 1989). To achieve comparable results within splits and even-sized train-validation-test sets, the dataset is shifted for 30 days in every iteration.

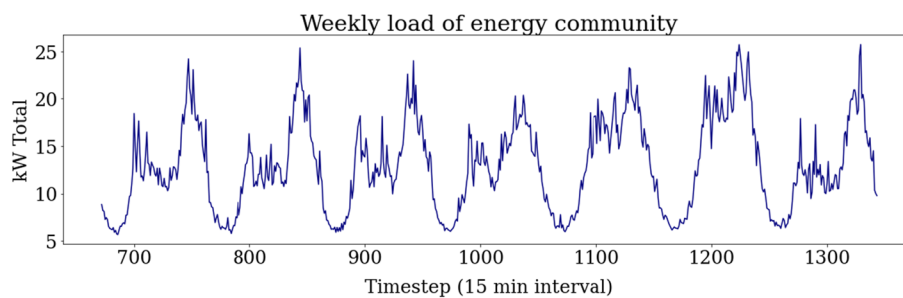
In the following, we apply the developed methodology to a case study in order to demonstrate the achievable improvements in energy community load forecasting through our developed model.

Case study

In this section, the setup of our study is described. In particular, the underlying dataset is described, the results of the LSTM and XGBoost hyperparameter tuning are presented and the four forecast scenarios are introduced.

Table 4 General information and descriptive statistics about the underlying dataset

Parameter	Value
Time covered	01.01.2019 - 31.12.2019
Time resolution	15 min
Smart-metered households (N)	130
Minimum annual household consumption	116.65kWh
Maximum annual household consumption	2011.03kWh
Mean annual household consumption	779.91kWh
Aggregated annual load of energy community	101.4 MWh
Mean load of energy community	11.57 kW
Minimum load of energy community	4.92 kW
Maximum load of energy community	31.57 kW

**Fig. 3** Aggregated energy community load from 07.01.2019 to 14.01.2019

Dataset

The introduced method is evaluated based on a dataset of German smart meter household data from 2019 published by Beyertt et al. (2020). The dataset includes 200 households that agreed on the publication of their loads, thereof 70 households participated in a behavioral experiment. The data of the remaining 130 households is used in this study. The households from the study are distributed all over Germany, which prevents us from adding geographically dependent weather features to the data set. The calculated aggregated load of all 130 households represents the load of an hypothetical energy community. In Table 4, the dataset is described. In Fig. 3, an exemplary load of the energy community is depicted. We can observe an repeating pattern of load peaks in the morning and evening and load valleys in the night. The households in the dataset are relatively small with a mean annual household consumption of 779kWh.

The number of households in the energy community constructed in this paper lies in the range of the community sizes from existing studies. In Coignard et al. (2021), the communities are randomly sampled with 5 to 95 households with 4MWh annual consumption each, resulting in an aggregated load between 20MWh to 380MWh. In a case study from Heeten, Netherlands an energy community of 47 households is depicted, with a calculated energy usage of 164.500kWh per year Reijnders et al. (2020). In Schlund et al. (2018), different configurations of up to 500 distributed households are regarded.

Table 5 Final LSTM parameters

Parameter	Value
Optimizer	<i>Adam</i>
Time steps per day K	96
Past input timesteps as multiple of K	1
Batch size	64
Epochs	75
Steps per epoch	200
Learning rate	0.001
Number of units in LSTM-layer	200
Number of units in dense layers	130

Table 6 Final XGBoost parameters for peak time and peak load model

Parameter	Value peak time model	Value peak load model
N estimators	87	929
Max depth	41	96
Learning rate	0.18	0.43
Subsample	0.88	0.46
Colsample bytree	0.75	0.88
Min child weight	4	0.05
Gamma	10	2

The dataset is split in twelve parts for the twelve-fold cross-validation. The first 252 days (36 weeks) of data serve as training data, the following 83 days (11.86 weeks) for validation and the remaining 30 days (4 weeks) as test data, representing approximately one month each. After every iteration, the dataset is shifted by 30 days. Therefore, our train-validate-test split is 70%, 23% and 7%.

LSTM model

The proposed LSTM is set up based on best practices from existing research (Kong et al. 2017; Muzaffar and Afshari 2019; Zheng et al. 2017; Bouktif et al. 2018; Jiao et al. 2018; Bouktif et al. 2020; Jahangir et al. 2020). Several optimizers are compared (SGD, Adagrad, RMSProp, Adam). Due to slightly better results, the Adam optimizer is used. For improved computational efficiency, training is stopped early when no further improvements in valuation loss can be observed. The final LSTM parameters obtained from the hyperparameter search are listed in Table 5.

The models are trained and evaluated on a Google virtual machine with 8 virtual CPUs and 64 GB RAM. The LSTM neural networks are realized with the help of the tensorflow toolkit (Abadi et al. 2016).

XGBoost

To find the optimal parameters for the XGBoost models for peak time and peak load forecasting, a hyperparameter search has been conducted. The resulting parameters are listed in Table 6.

Scenarios

In this work, four different scenarios are compared. Standard load profiles (SLP) for the year 2019 are used as baseline case, obtained from Standardlastprofil Haushalt (2019). The standard load profiles are scaled proportionally to the aggregated energy community load (Meier 2000). In a second scenario, the LSTM is used to forecast day-ahead energy community loads, with the only input features being day-before aggregated energy community load P_{agg} and type-of-day features as inputs, such as the *sin* and *cos* of the hour, weekday or month. The second scenario is in the following denoted as LSTM. In the third scenario, we add the smart metered loads of the past day of each individual household of the 130 consumers (LSTM SM). Finally, in the fourth scenario we combine the results of the third scenario with the XGB peak load finetuning (LSTM SM XGB) (Table 7). All four scenarios and the respective input datasets are summarized in Table 8.

Results

In this section we describe and compare the results of the four introduced scenarios. We also evaluate the standalone performance of the XGBoost model and present the results of the permutation feature importance analysis.

In Fig. 4, the day-ahead forecast for October 17 2019, a weekday, is displayed for the standard load profiles (SLP), the general LSTM model (LSTM) and the LSTM model with smart meter data (LSTM SM). We can observe that both the LSTM and LSTM SM manage to forecast the general load pattern quite well, whereas the SLP overestimates the actual load profile on this certain day. When we also take the day-ahead forecasts of other days into account, we can see that the SLP follows a rather generic pattern, that only manages to match the daily load irregularly. We also note that the LSTM SM forecasts the day-ahead loads slightly better than the LSTM.

Before its integration into the LSTM model, the peak load and peak time forecasting performances of the XGBoost model are compared to a forecast based on historical values. The XGBoost model is compared with a day-ahead forecast based on the peak load and peak time of the same day in the week before. For the evaluation, a twelvefold cross validation is conducted in the same way as described in the previous chapter. For the peak load forecast, the averaged XGBoost MAPE over the twelvefold cross validation

Table 7 Summary of included data and methods in evaluated scenarios

Scenario	Standard Load Profiles	LSTM	Smart meter data	XGB finetuning
SLP	Yes	No	No	No
LSTM	No	Yes	No	No
LSTM SM	No	Yes	Yes	No
LSTM SM XGB	No	Yes	Yes	Yes

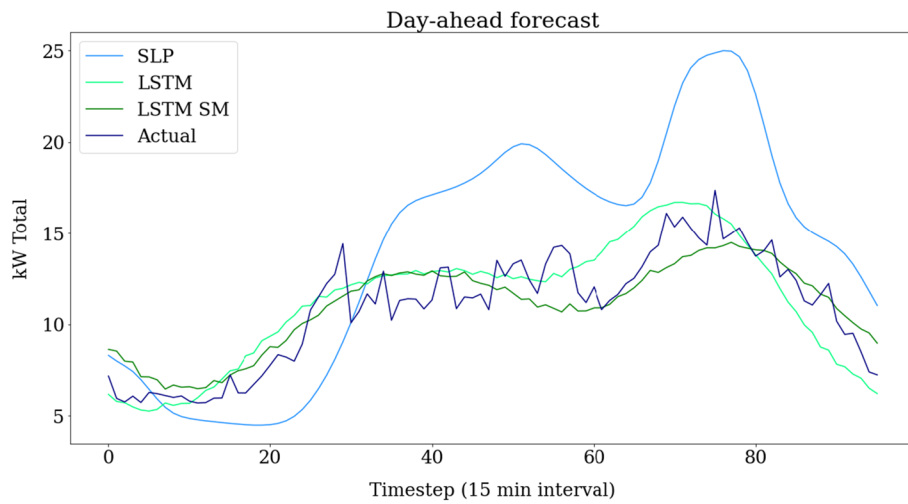


Fig. 4 Day-ahead forecast for 2019-10-16 with standard load profiles (SLP), the general LSTM model (LSTM) and the smart meter-based LSTM model (LSTM SM). We can observe that both LSTM models forecast the actual day-ahead load pattern more accurate

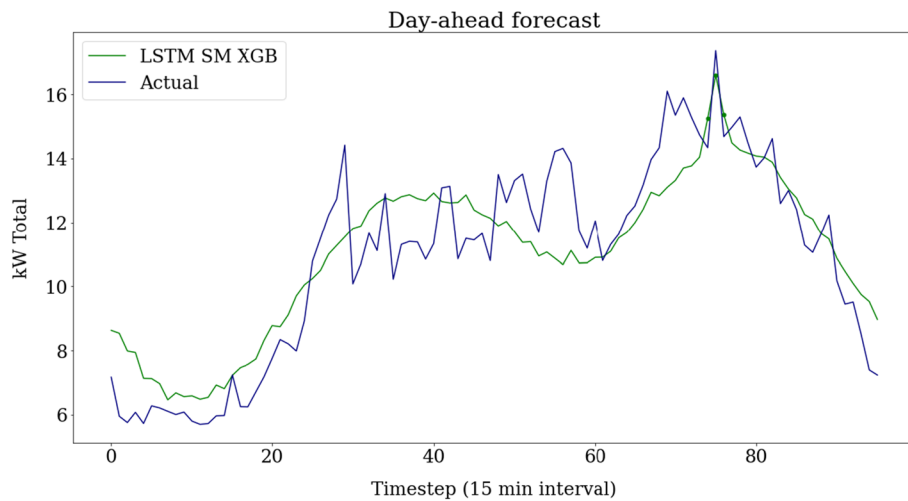


Fig. 5 Day-ahead forecast for 2019-10-16 with hybrid LSTM-XGBoost model. Peak load finetuning is conducted on timestep 75, marked with green dots. The overall forecasting accuracy in terms of MAPE is improved

($M = 7.75$, $SD = 1.35$) compared to the averaged MAPEs of the forecast through week-before peak loads ($M = 9.65$, $SD = 1.45$) demonstrated a significant improvement, $t(20) = 3.2$, $p = .005$. For the peak time forecast evaluation, the amount of correctly forecasted peak times is compared first. Again, comparing the averaged matches of the XGBoost peak time forecast ($M = 2.82$, $SD = 1.54$) with the matches of the forecast with week-before values ($M = 1.55$, $SD = 1.37$) yields a significant improvement, $t(20) = -2.05$, $p = .05$. As final metric, it is counted how often the forecasted peak time is amongst the five highest day-ahead load time steps. Comparing the averaged top five occurrences through the XGBoost model ($M = 19.27$, $SD = 3.26$) and the top five occurrences

Table 8 MAPE and RMSE results by periods, for overall forecasting accuracy

Evaluated period	MAPE (overall) [%] / RMSE (overall)			
	SLP	LSTM	LSTM SM	LSTM SM XGB
01.01–30.01.	24.85 / 3.52	19.63 / 3.70	18.25 / 3.27	<u>17.98 / 3.23</u>
31.01.–01.03.	25.64 / 3.87	<u>19.42 / 3.20</u>	25.35 / 3.87	25.04 / 3.83
02.03.–31.03.	24.05 / 3.94	17.82 / 2.79	11.18 / 2.14	<u>11.04 / 2.11</u>
01.04.–30.04.	24.95 / 3.97	25.63 / 3.00	19.40 / 2.83	<u>19.13 / 2.79</u>
01.05.–30.05.	28.03 / 4.71	24.81 / 2.84	<u>23.57 / 2.79</u>	23.61 / 2.79
31.05.–29.06.	27.53 / 4.78	28.50 / 3.33	10.71 / 1.60	<u>10.70 / 1.59</u>
30.06.–29.07.	27.76 / 4.81	16.69 / 1.96	<u>11.00 / 1.41</u>	11.15 / 1.43
30.07.–28.08.	27.01 / 4.87	23.68 / 2.60	20.41 / 2.35	<u>20.39 / 2.35</u>
29.08.–27.09.	28.64 / 4.90	21.61 / 2.68	<u>17.36 / 2.06</u>	17.40 / 2.06
28.09.–27.10.	27.19 / 4.46	16.89 / 2.19	11.29 / 1.73	<u>11.20 / 1.71</u>
28.10.–26.11.	21.50 / 3.21	17.38 / 2.52	15.39 / 2.74	<u>14.99 / 2.67</u>
27.11.–26.12.	26.20 / 3.71	27.63 / 4.13	19.52 / 3.36	<u>19.16 / 3.30</u>
Average	26.11 / 4.23	21.64 / 2.91	16.95 / 2.51	<u>16.81 / 2.49</u>

Best values in respective rows highlighted in bold font and underlined

Table 9 MAPE results by periods, for accuracy of forecasted peaks

Evaluated Period	MAPE (forecasted peak) [%]			
	SLP	LSTM	LSTM SM	LSTM SM XGB
01.01–30.01.	33.71	22.86	14.64	<u>9.22</u>
31.01.–01.03.	33.91	20.04	10.36	<u>10.2</u>
02.03.–31.03.	38.99	14.75	17.02	<u>12.27</u>
01.04.–30.04.	43.61	13.19	13.62	<u>11.45</u>
01.05.–30.05.	68.92	25.7	26.99	<u>23.24</u>
31.05.–29.06.	69.35	37.44	<u>9.61</u>	10.11
30.06.–29.07.	71.47	19.48	<u>13.80</u>	19.04
30.07.–28.08.	82.33	36.48	31.64	<u>27.42</u>
29.08.–27.09.	72.80	36.32	22.98	<u>18.99</u>
28.09.–27.10.	60.94	18.05	12.61	<u>9.72</u>
28.10.–26.11.	33.02	12.97	20.54	<u>10.09</u>
27.11.–26.12.	30.87	<u>11.44</u>	22.03	11.52
Average	53.33	22.39	17.99	<u>14.44</u>

Best values in respective rows highlighted in bold font and underlined

through the week-before forecast ($M = 16.64$, $SD = 3.32$), a slightly significant improvement can be observed once again, $t(20) = 1.87$, $p = .07$.

After evaluating the standalone performance of the XGBoost model, the forecast of the hybrid LSTM-XGBoost model is depicted in Fig. 5. For this exemplary day it can be seen how the incorporation of the XGBoost-based peak load and peak time forecast can improve the overall forecast quality.

The results of the twelvefold cross validation of the four scenarios are depicted in Table 8 and Table 9. We can observe that, on average, the LSTM SM XGB outperforms all other models in terms of overall MAPE. In comparison with the LSTM SM, an average improvement of 0.14 percentage points is achieved. Within the test period

between the 28.10.-26.11., the LSTM SM XGB model manages to improve the accuracy by 0.4 percentage points compared to the LSTM SM model. Another remarkable observation is that adding individual smart meter data as additional input data significantly improves the model. Compared to the simple LSTM model, LSTM SM reaches a MAPE of 16.95 compared to 21.64 without smart meter data, an improvement of 4.69 percentage points. Only in the second evaluated period, the LSTM model performs better than the LSTM SM. All models consistently outperform the SLP. The outperformance of the LSTM SM XGB model is confirmed by the RMSE metric.

Furthermore, we evaluate the MAPE of forecasted peaks. Again, the LSTM SM XGB outperforms all other models. In comparison with the LSTM SM, an improvement of 3.55 percentage points is reached on average. In 9 out of 12 months, the LSTM SM XGB outperforms the other models in terms of overall MAPE. In 8 out of 12 periods, the LSTM SM XGB forecasted peak MAPE outperforms the other models. Once again, we can observe that adding smart meter data (LSTM SM) improves the forecast accuracy from a MAPE of 22.39 to a MAPE of 17.99, which reflects an improvement of 4.4 percentage points. Most notably is the improvement in peak forecast accuracy compared to the SLP, with an improvement of 38.89 percentage points between SLP and LSTM SM XGB.

As the addition of individual smart meter data significantly improved the overall community forecast performance, we are interested in finding out which features, and especially which households' smart meter data, is important to improve forecast quality. This information could be used to identify characteristics of households in which it is particularly helpful for forecasting tasks to install smart meters.

The Permutation importance (PIMP) for the LSTM SM are depicted in Fig. 6. We can observe that the aggregated energy community load (sum) is by far the most important

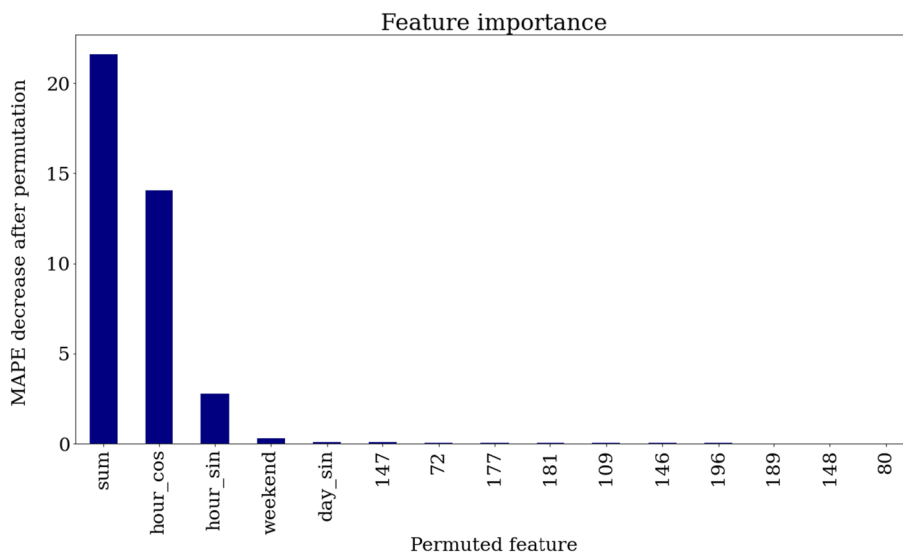


Fig. 6 Average feature importances for smart meter-based LSTM (LSTM SM). We can see that the aggregated energy community load, the sin and cos transform of the hour and day, weekend binary as well as selected households serve as most important input features to forecast day-ahead energy community loads. Interestingly, most important households are also amongst the households with the highest annual electricity consumption

feature. Further important features are the \sin and \cos transformed hour and day, as well as the binary variable for weekends. Also, the loads of selected customers are important input features for the LSTM. While the feature importances of the households seem relatively low in comparison to the sum and the cyclical features, we know from the results in Table 8 and 9 that the addition of smart meter data leads to significant improvements and therefore even though seemingly small, these feature importances should not be neglected. Most of the households with a high feature importance are also households with relatively high annual electricity consumption. For instance, household 147 is the fourth largest household amongst the 130 smart metered households with an annual electricity consumption of 1,700 kWh. Household 177 is the 8th largest household with an annual consumption of 1,448kWh, household 181 is 11th with 1,352kWh annual consumption. However, there are also several households with high feature importances that do not belong to the largest households.

Discussion

In this section we discuss the presented results and their implications for day-ahead load forecasting in energy communities. The study has been conducted with load data of a limited number of German households. Hence, it has to be investigated if the results of this study still prove valid in communities with a higher number of smart metered households, as well as data from other countries and differing community configurations. Also, we were not able to include weather data as input feature due to the geographic distribution of the households from the underlying dataset. This leaves opportunities for further research. In the following, we discuss two aspects of our work in particular.

First, we observe that the addition of smart meter data in energy communities can improve the day-ahead forecasting accuracy of energy communities significantly in our case study. This confirms the results of Zufferey et al. (2016), where also a higher accuracy in aggregated load prediction was reached by increasing the number of smart meters. Hence, we suggest to consider the installation and implementation of smart meters in the planning process for energy communities. Our results indicate that selected households contribute more to the improvement of forecasting quality than others. For instance, households with a larger annual consumption seem to have a larger impact on the forecast than smaller households. Still, this does not hold true for all households with a high feature importance. Thus, further research has to focus on identifying characteristics of households that improve the forecasting quality. With this information, grid operators and energy community managers could selectively install smart meters to optimize their day-ahead forecasting model.

Our feature importance analysis showed that the most important factor for forecasting day-ahead loads of energy communities is the past aggregated energy community load P_{agg} itself. It has to be noted that engineered type-of-day features, such as the \sin and \cos transformation of the hour, are by far the second most important input features. Hence, we strongly propose that coming works in the field of load forecasting also include \sin and \cos transformed type-of-day features.

Second, we introduce a novel hybrid LSTM-XGBoost model that enables improved peak load forecasts by separately forecasting the general load pattern and peak loads. To our knowledge, we are the first ones to propose peak load time and quantity forecasting through

a dedicated XGBoost model and to combine an LSTM and XGB forecast into a holistic model. By using the hybrid LSTM-XGBoost model, we can improve the overall model performance and peak forecasting performance in our study. In addition, we propose that further research also evaluates the performance of a hybrid peak load forecasting XGBoost model in combination with other recent proposed algorithms like temporal attention based convolutional networks (Tang et al. 2022) or federated learning (Fekri et al. 2022).

Conclusion

In this paper, we propose a framework for smart meter-based day-ahead forecasting in energy communities with bi-directional LSTM neural networks and a combined LSTM-XGBoost model. Furthermore, we contribute to the general understanding of important input features in smart meter-based energy community load forecasting. We can draw three main conclusions.

First, our results confirm that the LSTM-based models achieve a significantly higher accuracy than forecasting based on standard load profiles. In addition, using smart meter data as additional input data further improves the forecasting accuracy in our case study.

Second, the novel hybrid LSTM-XGBoost manages to further increase the forecasting accuracy of smart meter-based models, especially in terms of peak load forecasting.

Third, the most important features for the forecast of the aggregated energy community load are, in our case study, the past aggregated load itself, transformed hour and day data, a binary weekend variable as well as past loads of selected households. We see a tendency that the past loads of households with a higher annual consumption may be more important features, but this needs to be confirmed and further investigated in future research.

This paper gives scope for further research in the field of energy community load forecasting. Future work should further confirm and deepen the assessment of the hybrid LSTM-XGBoost model and its viability in cases without smart meter data or in combination with alternative forecasting algorithms. Furthermore, adding weather data to the forecasting process could be an interesting addition to this study.

Abbreviations

l	Input feature for LSTM layer as vector of the past K timesteps
K	Time steps per day
N	Set of all smart metered households
Obj	Objective function
$P_{max,d,agg}$	Highest energy community load during day d
$\Omega(f_m)$	Regularization term used in XGBoost
X_d	LSTM input matrix for forecast of day d
$\hat{P}_{t_{XGB}}$	Adjusted energy community load at forecasted peak load time t_{XGB}
\mathcal{L}_{split}	Tree splitting decision
P_{agg}	Aggregated load of all households in energy community
$\hat{P}_{t_{XGB}+1}$	Adjusted energy community load after forecasted peak load time t_{XGB}
$\hat{P}_{t_{XGB}-1}$	Adjusted energy community load before forecasted peak load time t_{XGB}
\hat{P}_t	Forecasted energy community load at point of time t
$\mathcal{L}^{(t)}(q)$	Scoring function for forecasting quality of tree q
\hat{y}_i	Predicted value for sample i
$t_{P_{max,d,agg}}$	Time of highest load during day d
w_j^*	Leaf node score

Author contributions

LS, SH and CW conceived and developed the general idea of the paper. LS and SH developed and implemented the forecasting model. All authors read and approved the final manuscript.

About this supplement

This article has been published as part of Energy Informatics Volume 5 Supplement 1, 2022: Proceedings of the 11th DACH+ Conference on Energy Informatics. The full contents of the supplement are available online at <https://energyinformatics.springeropen.com/articles/supplements/volume-5-supplement-1>.

Availability of data and materials

The underlying smart meter dataset is publicly available at Beyertt *et al.* (2020).

Declarations**Competing interests**

The authors declare that they have no competing interests.

Published: 7 September 2022

References

- Abadi M, Barham P, Chen J, Chen Z, Davis A, Dean J, Devin M, Ghemawat S, Irving G, Isard M, *et al.* (2016) TensorFlow: A system for Large-Scale machine learning. In: 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16), pp 265–283
- Altmann A, Tološi L, Sander O, Lengauer T (2010) Permutation importance: a corrected feature importance measure. *Bioinformatics* 26(10):1340–1347
- Atef S, Eltawil AB (2020) Assessment of stacked unidirectional and bi-directional long short-term memory networks for electricity load forecasting. *Electric Power Syst Res* 187:106489
- Beyertt A, Verwiebe P, Seim S, Milojkovic F, Müller-Kirchenbauer J (2020) Felduntersuchung zu Behavioral Energy Efficiency Potentialen von privaten Haushalten. An example dataset that accompanies the working paper
- Bouktif S, Fiaz A, Ouni A, Serhani MA (2018) Optimal deep learning LSTM model for electric load forecasting using feature selection and genetic algorithm: comparison with machine learning approaches. *Energies* 11(7):1636
- Bouktif S, Fiaz A, Ouni A, Serhani MA (2020) Multi-sequence LSTM-RNN deep learning and metaheuristics for electric load forecasting. *Energies* 13(2):391
- Burman P (1989) A comparative study of ordinary cross-validation, v-fold cross-validation and the repeated learning-testing methods. *Biometrika* 76(3):503–514
- Chen T, Guestrin C (2016) Xgboost: a scalable tree boosting system. In: Proceedings of the 22nd Acm Sigkdd International Conference on Knowledge Discovery and Data Mining, pp. 785–794
- Coignard J, Janvier M, Debusschere V, Moreau G, Chollet S, Caire R (2021) Evaluating forecasting methods in the context of local energy communities. *Int J Electr Power Energy Syst* 131:106956
- Dube M, Awodele K, Olayiwola O, Akpeji K (2017) Short term load forecasting using arima ann and hybrid ann-dwt. In: Southern African Universities Power Engineering Conference, p. 6
- European Parliament and Council of the European Union: Directive 2019/944 on common rules for the internal market for electricity
- Fekri MN, Grolinger K, Mir S (2022) Distributed load forecasting using smart meter data: federated learning with recurrent neural networks. *Int J Electr Power Energy Syst* 137:107669
- Feng D, Fang K, Shen C (2020) Enhancing streamflow forecast and extracting insights using long-short term memory networks with data integration at continental scales. *Water Resour Res* 56(9):2019–026793
- Ghiani E, Giordano A, Nieddu A, Rosetti L, Pilo F (2019) Planning of a smart local energy community: the case of Berchidda municipality (Italy). *Energies* 12(24):4629
- Golla A, Henni S, Staudt P, Weinhardt C (2020) Scaling the concept of citizen energy communities through a platform-based decision support system. European Conference on Information Systems 2020, Marakesh
- Grundmeier N, Hahn A, Ihle N, Runge S, Meyer-Barlag C (2014) A simulation based approach to forecast a demand load curve for a container terminal using battery powered vehicles. In: 2014 International Joint Conference on Neural Networks (IJCNN), pp. 1711–1718. IEEE
- Haben S, Giasemidis G (2016) A hybrid model of kernel density estimation and quantile regression for gefcom2014 probabilistic load forecasting. *Int J Forecast* 32(3):1017–1022
- Haben S, Arora S, Giasemidis G, Voss M, Greetham DV (2021) Review of low voltage load forecasting: methods, applications, and recommendations. *Appl Energy* 304:117798
- Henni S, Staudt P, Weinhardt C (2021) A sharing economy for residential communities with PV-coupled battery storage: benefits, pricing and participant matching. *Appl Energy*. <https://doi.org/10.1016/j.apenergy.2021.117351>
- Hochreiter S, Schmidhuber J (1997) Long short-term memory. *Neural Comput* 9(8):1735–1780
- Hor C-L, Watson SJ, Majithia S (2006) Daily load forecasting and maximum demand estimation using arima and garch. In: 2006 International Conference on Probabilistic Methods Applied to Power Systems, pp. 1–6. IEEE
- Huang N, Lu G, Xu D (2016) A permutation importance-based feature selection method for short-term electricity load forecasting using random forest. *Energies* 9(10):767

- Jahangir H, Tayarani H, Gougheri SS, Golkar MA, Ahmadian A, Elkamel A (2020) Deep learning-based forecasting approach in smart grids with microclustering and bi-directional LSTM network. *IEEE Trans Industr Electron* 68(9):8298–8309
- Jiao R, Zhang T, Jiang Y, He H (2018) Short-term non-residential load forecasting based on multiple sequences LSTM recurrent neural network. *IEEE Access* 6:59438–59448
- Kanda I, Veguillas JQ (2019) Data preprocessing and quantile regression for probabilistic load forecasting in the gef-com2017 final match. *Int J Forecast* 35(4):1460–1468
- Karimian H, Li Q, Wu C, Qi Y, Mo Y, Chen G, Zhang X, Sachdeva S et al (2019) Evaluation of different machine learning approaches to forecasting pm_{2.5} mass concentrations. *Aerosol Air Qual Res* 19(6):1400–1410
- Kong W, Dong ZY, Jia Y, Hill DJ, Xu Y, Zhang Y (2017) Short-term residential load forecasting based on LSTM recurrent neural network. *IEEE Trans Smart Grid* 10(1):841–851
- Kucevic D, Semmelmann L, Collath N, Jossen A, Hesse H (2021) 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
- Lahouar A, Slama JBH (2015) Day-ahead load forecast using random forest and expert input selection. *Energy Convers Manage* 103:1040–1051
- Lee C-M, Ko C-N (2011) Short-term load forecasting using lifting scheme and arima models. *Expert Syst Appl* 38(5):5902–5911
- Li C, Chen Z, Liu J, Li D, Gao X, Di F, Li L, Ji X (2019) Power load forecasting based on the combined model of lstm and xgboost. In: *Proceedings of the 2019 the International Conference on Pattern Recognition and Artificial Intelligence*, pp. 46–51
- Liu J, Brown LE (2019) Prediction of hour of coincident daily peak load. In: *2019 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT)*, pp. 1–5. IEEE
- Meier H (2000) Practical application of the vdev standard load profiles. analytical vs. synthetic load profiles; anwendung der vdev-lastprofile. analytisches versus synthetisches verfahren. *ET, Energiewirtschaftliche Tagesfragen* **50**
- Muzaffar S, Afshari A (2019) Short-term load forecasts using LSTM networks. *Energy Procedia* 158:2922–2927
- Peters D, Völker R, Schuldt F, von Maydell K (2020) Are standard load profiles suitable for modern electricity grid models? In: *2020 17th International Conference on the European Energy Market (EEM)*, pp. 1–6. IEEE
- Reijnders VM, van der Laan MD, Dijkstra R (2020) Energy communities: a Dutch case study. In: *Behind and beyond the meter*, pp. 137–155. Elsevier, Amsterdam
- Sarduy JRG, Di Santo KG, Saidel MA (2016) Linear and non-linear methods for prediction of peak load at university of São Paulo. *Measurement* 78:187–201
- Schlund J, Pflugradt N, Steber D, Muntwyler U, German R (2018) Benefits of virtual community energy storages compared to individual batteries based on behaviour based synthetic load profiles. In: *2018 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT-Europe)*, pp. 1–6. IEEE
- Shrestha A, Bishwokarma R, Chapagain A, Banjara S, Aryal S, Mali B, Thapa R, Bista D, Hayes BP, Papadakis A et al (2019) Peer-to-peer energy trading in micro/mini-grids for local energy communities: a review and case study of Nepal. *IEEE Access* 7:131911–131928
- Shwartz-Ziv R, Armon A (2022) Tabular data: deep learning is not all you need. *Inf Fusion* 81:84–90
- Standardlastprofil Haushalt 2019 (Berlin). *Stromnetz Berlin GmbH* (2019). <https://daten.berlin.de/datensaetze/standardlastprofil-haushalt-2019-berlin>
- Tang X, Dai Y, Wang T, Chen Y (2019) Short-term power load forecasting based on multi-layer bi-directional recurrent neural network. *IET Gener Transm Distrib* 13(17):3847–3854
- Tang X, Chen H, Xiang W, Yang J, Zou M (2022) Short-term load forecasting using channel and temporal attention based temporal convolutional network. *Electric Power Syst Res* 205:107761
- Wang S, Wang X, Wang S, Wang D (2019) Bi-directional long short-term memory method based on attention mechanism and rolling update for short-term load forecasting. *Int J Electr Power Energy Syst* 109:470–479
- Wang Y, Sun S, Chen X, Zeng X, Kong Y, Chen J, Guo Y, Wang T (2021) Short-term load forecasting of industrial customers based on SVM and xgboost. *Int J Electr Power Energy Syst* 129:106830
- Wang X, Fang F, Zhang X, Liu Y, Wei L, Shi Y (2019) Lstm-based short-term load forecasting for building electricity consumption. In: *2019 IEEE 28th International Symposium on Industrial Electronics (ISIE)*, pp. 1418–1423. IEEE
- Wang W, Shi Y, Lyu G, Deng W (2017) Electricity consumption prediction using xgboost based on discrete wavelet transform. *DEStech Trans. Comput. Sci, Eng*
- Wen L, Zhou K, Yang S, Lu X (2019) Optimal load dispatch of community microgrid with deep learning based solar power and load forecasting. *Energy* 171:1053–1065
- Zheng H, Wu Y (2019) A xgboost model with weather similarity analysis and feature engineering for short-term wind power forecasting. *Appl Sci* 9(15):3019
- Zheng H, Yuan J, Chen L (2017) Short-term load forecasting using EMD-LSTM neural networks with a xgboost algorithm for feature importance evaluation. *Energies* 10(8):1168
- Zufferey T, Ulbig A, Koch S, Hug G (2016) Forecasting of smart meter time series based on neural networks. In: *International workshop on data analytics for renewable energy integration*, pp. 10–21. Springer

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.