Electrical Engineering
https://doi.org/10.1007/s00202-020-01126-z

ORIGINAL PAPER

Prediction of electrical energy consumption based on machine learning technique

Rita Banik[1] · Priyanath Das[1] · Srimanta Ray[2] · Ankur Biswas[3]

Received: 8 January 2020 / Accepted: 9 October 2020

Abstract

The forecast of electricity demand in recent years is becoming increasingly relevant because of market deregulation and the introduction of renewable resources. To meet the emerging challenges, advanced intelligent models are built to ensure precise power forecasts for multi-time horizons. The use of intelligent forecasting algorithms is a key feature of smart grids and an effective tool of resolving uncertainty for better cost and energy efficiency decisions like scheduling the generations, reliability and power optimization of the system, and economic smart grid operations. However, prediction accuracy in forecasting algorithms is highly demanded since many important activities of power operators like load dispatch depend upon the short-term forecast. This paper proposes a model for the estimation of the consumption of electricity in Agartala, Tripura in India, which can accurately predict the next 24 h of load with and estimation of load for 1 week to 1 month. A number of specific characteristics in the city have been analysed in order to extract variables that could affect the pattern of electricity consumption directly. In addition, the present paper shows the way to significantly improve the accuracy of the prediction through ensemble machine learning process. We demonstrated the performance of individual Random forest and XGBoost along with their ensemble. The RF and XGBoost ensemble obtained an accuracy with an improvement of 15–29%. The analyses or findings also provide interesting results in connection with energy consumption.

Keywords Energy prediction · Random forest · Xgboost · Ensemble machine learning · Short-term load forecasting

1 Introduction

The demand for electrical energy is increasing with rising economic growth [1, 2]. The efficiency of energy plants decreases in the event of an immediate need for additional energy, and they generate toxic exhausts. An appropriate energy production plan can reduce the detrimental effect of the plants [3]. Moreover, with the emergence of renewable energy resources and smart grids, the prediction of electrical energy consumption or load forecast is getting more essential. Load forecasting, which aims to predict future load demand means projecting how the electrical load of an individual apartment, a grid, an area, and even a whole country will evolve in the future. This forecast is carried out in one or more phases over a time period called the prediction horizon. An hour or several-hours ahead predictions called Short-term load forecasting (STLF) is generally needed for programming and energy transfer scheduling, unit allocation, and decisions about load imbalance, with lead times of half an hour or one day. Some increase in STLF's accuracy can thus result in an improvement in power management efficiency and a reduction in power system costs. Regarding maintenance and power management strategies, forecasting for a longer horizon is also useful. Increasing the accuracy by 1% leads to a massive decrease in operating costs [4], so it is interesting to make the smallest change in prediction error. The error can be a power overestimation or underestimation, which both lead to problems balancing demand and supply. The idea of the smart grid involves the use of advanced and intelligent computing technologies, including power prediction to control supply so as to meet demand in real-time. The management of supply is strongly associated with the spinning reserve. Prediction of load then includes estimating the

✉ Rita Banik
rita.nit@gmail.com

[1] Department of Electrical Engineering, NIT Agartala, Agartala, Tripura, India

[2] Department of Chemical Engineering, NIT Agartala, Agartala, Tripura, India

[3] Department of Computer Science and Engineering, TIT Narsingarh, Agartala, Tripura, India

Published online: 30 October 2020

Spinning Reserve = Cadangan kapasitas pembangkit yang siap untuk diaktifkan dengan cepat untuk menyeimbangkan pasokan dan permintaan listrik ketika permintaan meningkat secara tak terduga, atau ketika generator mengalami gangguan atau kegagalan

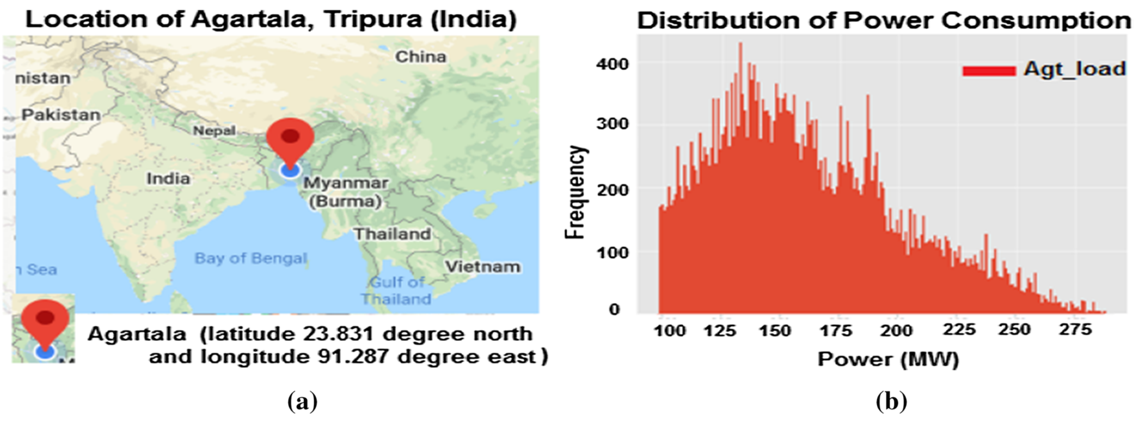


Fig. 1 **a** Site location of Agartala, **b** electrical energy consumption distribution

spinning reserve as well, which is crucial when demand raises unexpectedly, generators break down or fails. The spinning reserve is able to compensate for any shortcomings quickly when the forecast is right. For longer horizons, the load profile forecast dictates the ability to add to the entire network, to avoid an emergency.

For recent intelligent energy management systems, load electricity demand forecasting is important [5]. It plays a key role in the allocation of short-term load and long-term planning for a new generation and transmission infrastructure. An accurate forecast also allows for better cost and energy efficiency decisions. More and more load forecasting applications are being made each day. In this regard, many works have devoted their content to the domain of electricity demand prediction [6, 7]. Some of the literature predicts that weather variables like temperature, humidity, rainfall, or season may help to influence energy consumption [8], while other studies estimated on the basis of socio-economic and population variables [9]. While the prediction of electricity consumption based on general variables is significant, a better result can be achieved if the model of prediction is altered. To validate this assumption, a forecasting model is developed for a particular city's power consumption i.e. Agartala, Tripura located in India, based on the original data obtained from State load Despatch centre. The model is developed using machine learning ensemble of Random Forest (RF) and extreme gradient boosting (XGBoost) [10]. The location of the site [11] and the distribution of electrical energy consumption is shown in Fig. 1.

Hence, the contribution of this paper refers in multiple directions to the prediction of electricity consumption:

1. First, an RF and XGBoost ensemble model for Agartala based on real data is presented in this paper.
2. It demonstrates the potential of the RF-XGBoost ensemble technique.
3. It analyses the output of the model with different models of a similar standard for multiple time horizons.
4. Finally, the research discusses the different parameters that affect the forecast and identifies the most important parameters.

The rest of the paper is organized as follows. Section 2 presents a literature review of related work, Sect. 3 delivers the hypothesis of machine learning with boosting and ensemble. Then, Sect. 4 presents the proposed model, and Sect. 5 demonstrates the analysis of the results after the modelling process. Finally, Sect. 6 concludes this paper.

2 Related work

Prediction is a statistical analysis applied to sequences of time series, which implies that several factors, including future and historical observations, have to be tested. The load signal is a time series and a forecasting tool must predict its future development from historical observations and some predictor variables that influence the future load. In the first steps, the forecast issue was discussed by statistical methods like regression, multiple regression, smoothing, weighted least squares, etc. up to fuzzy logic and machine learning. The first load prediction studies showed a regression [12] whose authors employed linear regression for load prediction, while an approach based on nonlinear load regression has been developed by Hyde et al. [13] and Broadwater et al. [14]. Several autoregressive modelling approaches have been used: El-Keib et al. [15] worked on short-term prediction models using exponentially smoothing, and Hurang [16] proposed autoregressive model for short-term load prediction. Many forecasting model designs were developed for very short-term horizon; for less than an hour, for short-term; for an hour or more upto a week, for medium-term; for one or more months and for a longer time; for one to several years. The techniques of modelling can be classified into statistical, artificial intelligence, and hybrid.

In statistical models, the output is related directly with inputs via mathematical equations. The implementation of these techniques is easy and is well suited short-term predictions, but fails to accommodate the nonlinearity of the load series which is why an intelligent technique is demanded. Statistical methods comprise linear regression [17], multiple regression [18], autoregressive moving average (ARMA) [19], and autoregressive integrated moving average (ARIMA) [20], etc.

Artificial intelligence (AI) techniques are black-box approaches of unknown internal dynamics. Three major methods are included in this category, including fuzzy inference system (FIS) [21] artificial neural networks (ANN) [22, 23] and support vector machines (SVM) [24]. In FIS, a collection of linguistic rules for fuzzy structures will decide the association between input and output. While in SVM and ANN, this association is determined by training. The problem of the ANN model to find solutions for local optimization problems and the tendency of both under-fitted and over-fitted is addressed by SVM through empirical risk minimization principle [25, 26]. Another popular approach is random forest (RF) [27, 28], which rely on training. The improvement in RF is its lower sensitiveness towards parameter values and suitable for nonlinear estimation [29]. All the AI-based techniques require an optimum architectural design and tuning of parameters, which can be effectively handled through hybridization. Recently, Khayatian et al. [30] applied ANN to predict energy performance certificates of domestic buildings in Italy. Ascione et al. [31] investigated the association of energy usage and occupant thermal comfort in the prediction of energy performance.

A hybrid or combined model of SVM and multi-resolution wavelet decomposition was developed by Chen and Tan [32] for the prediction of power utilization in different buildings. The precision of linear and regression GP was contrasted by Rastogi et al. [33] which revealed that the reliably of GP beats linear regression by four times while simulating a building efficiency. Tree-based predictive models are also evaluated by Papadopoulos et al. [34] for the prediction of energy efficiency in buildings. A recent short-term forecast of power using RF was studied by Wang et al. [35] for office building parameters of envelope, climate, and time. In that study, RF was shown to be dominant over regression trees and SVM while forecasting hourly electricity load in buildings. Several implementations of Deep Neural networks are also available. Ahmad et al. [36] predicted energy demand utilizing climate, date, and building usage rate. Lee et al. [37] computed the country-wise environmental consumption level with a big data analytics tool. Li et al. [38] utilized Autoencoder to extract the building energy demand and forecast future energy consumptions. Kim et al. [39] utilized State Explainable Autoencoder to predict household electricity consumption with 5 years of data.

3 Machine learning model and variables

In order to increase prediction accuracy over conventional energy consumption methods, Machine learning (ML) techniques have been developed [40, 41]. The prevalent approach of forecasting electricity consumption is Regression analysis which is based on the concept that there is a relation between the amount of energy usage and other meteorological parameters such as temperature, precipitation, air density, humidity, and wind speeds. In this section, we review the predominantly used algorithms of machine learning in energy consumption forecasting domain such as RF and XGB along with their ensembles.

3.1 Random forest model

In the ML context, the RF models can be seen as bagging strategies with minor changes. In RF, the bagging estimator algorithm is enhanced where decision trees form the base estimators. Random samples from the training set are drawn in this technique. In contrast to bagging, however, where each tree has a full range of features, RF draws a few features to train each tree for the best split. This allows the trees more independent, making the prediction more effective than bagging. As each tree is trained by a subset of features, it is also faster. On the other hand, Bagged decision trees choose to split variables in a greedy way that reduces errors. As such, even Bagging can keep several structural similarities, and their predictions are indeed strongly correlated. A mixture of several models' predictions in the ensemble thus works best if the predictions of the submodels are not correlated or correlated very poorly. RF modifies the algorithm to learn the sub-trees and minimize the correlation between all subtrees predictions. While choosing a split point, the learning algorithm is allowed to select the best split point in every variable. This function is modified by the RF algorithm so that only a random sample of features can be checked. The steps of RF are as follows:

1. Generate random subset (bootstrapping) in the sample.
2. Select a random feature set for the optimal split for each node in the decision tree.
3. Generate a model decision tree for each subset.
4. Aggregate forecasts from all decision trees and the average for final forecasting.

3.2 Ensemble learning

Ensemble is the strategy of combining various learners (single models) to improve prediction and model stability collectively. All the forecasts are integrated into ensemble learning and more than one learner is equipped to accomplish the same classification or regression goal [42]. Unlike a ML model, which attempts to form a single conclusion in the training system, ensemble models propose and integrate multiple conclusions. The main purpose of the model is to reduce generalization error by reducing the variance or bias [43]. Each learner of ensembles is a base learner and the prediction of an ensemble of base learners is enhanced compared to individual base learners. Error in ensemble model is given by [44],

$$\begin{aligned} \text{Error} &= \left(E\left[y'\right] - y\right)^2 + E\left(y' - E\left[y'\right]\right)^2 + \sigma_e^2 \\ \text{Error} &= \text{Bias}^2 + \text{Variance} + \text{Irreducible error} \end{aligned} \tag{1}$$

Bias error tends to measure the mean difference between the expected and real prediction values (i.e. how much on an average are the predicted values different from the actual values). A high bias error implies the model is inaccurate, and important trends tend to be lacking. On the other hand, variance quantifies how the results of the same observation are different (i.e. the prediction made on the same observation different from each other). A model of high variance overpasses and underperforms outside of the training instances. Irreducible error is the error ensuing from noise present in the dataset like missing values.

3.3 Boosting

In ensemble learning, boosting is a popular technique used to generate an accurate classification or regression resulting from a number of poor classification systems. It can be done by building models from training data through assigning a weight to the instance. It is an evolving process that generates a model for correcting sample errors from the previous model in the next iteration and model add-ons are continued until the training data prediction is accurate or a threshold is reached. At each iteration, the weight of an instance is modified and for an incorrectly classified instance, the weight is increased for the next iteration. Boosting decreases the bias error and therefore creates a better prediction model.

Step 1: Divide the training data into '*n*' sub-sample.
Step 2: Train '*n*' decision trees.
Build decision tree (based on features) for each subset
Step 3: Generate tuple predictions independently of each tree for the test set.
Step 4: Combine and build a final forecasting model

3.4 Gradient boosting

The most effective predictive model creation approaches are gradient boosting. The principle of boosting was based on a possibility to increase the development of a slow learner. The first design of the boosting that was successful in the project was Adaptive boosting or Adaboost. The first highly effective binary grading boost was Adaboost which is used in small decision trees. The training data use tree performance after the development of the first tree to allocate weight to the next tree. More weight is assigned to the data that are hard to forecast while those can be easily predicted are assigned with less weight. Sequentially, models are developed, adjusting the weights of the training data, each influencing the learning of the next tree in a row. Predictions for new data are made after all trees are completed. Because the algorithm emphasizes error correction, smooth data, i.e. free of outliers, is important. The Gradient Boosting model comprises of the following: loss function, decision trees, and additive models.

3.5 Extreme gradient boosting (XGBoost)

XGBoost, because of its fast parallelism and low predictive error, is the flexible and efficient ML implementation techniques. XGB uses the original Friedman's gradient boosting model theory [45]. The training data *xi* has been trained to predict a target variable 'y_i' and an ensemble of 'K' Classification and Regression Trees (CART) [46] {T1 (x_i, y_i)… TN (x_i, y_i)} where x_i is the given training set of descriptors. The XGB implementation through the gradient boost decision tree algorithm has shown that it is a highly efficient ML method. XGB's prediction results are high and run time is 10 times faster than conventional methods of gradient boosting. The XGB ensemble creates a new model that evaluates the errors of the previous model and integrates with a gradient descent approach to minimize the loss to obtain the overall prediction. The following advantages also make XGBoost's versatility to forecast variable instability:

- Multithreaded parallelism is incorporated into predictions incorporating large time-series data, hence faster than other common ensembles.
- The availability of L1 and L2 regularizations functions.
- There is no need to normalize the data in the tree structure model.
- Ability to handle missing data.

The data $d\{(x_i,y_i): i = 1 \text{ to } n\}$ with ’n’ sample of ’m’ feature, while $y_i^{'}$ is defined as the model prediction value given by,

$$y_i^{'} = \sum_{j=1}^{J} f_j(x_i), \quad f_j \in N \tag{2}$$

where ‘f_j’ is the regression tree and ‘$f_j(x)$’ represents the prediction score given by j-th tree to the data sample. $N = \{f(x) = W_{p(x)}\}$ (p: $\mathbb{R}^m \to T$, $W \epsilon \mathbb{R}^T$), the space of regression tree (CART), where ‘W’ is the leaf weight and ‘p’ is the structure of each tree mapping to its corresponding leaf index. Lastly, ‘T’ represents the number of leaf nodes in the tree. The function ‘f_j’ is learned by minimizing the objective function,

$$\Phi = \sum_{i=1}^{n} l\left(y_i, y_i^{'}\right) + \sum_{j=1}^{J} \Omega(f_k) \tag{3}$$

where ’l’ is the training loss and the regularization term ’Ω’ penalizes model complexity in order to avoid over-fitting. The optimal weight of leaf is given by,

$$\Omega(f_j) = \lambda_1 T + 1/2\lambda_2 ||w_t||^2 \tag{4}$$

where λ_1 and λ_2 are the degree of regularizations. ‘T’ and ‘w_t’ are leaf nodes and score, respectively, Assuming, $y_i^{'}$ (t) prediction at ’t’ iteration, ‘f_t’ is added to minimize the objective,

$$\Phi^{(t)} = \sum_{i=1}^{n} l\left(y, y'^{(t-1)} + f_t(x)\right) + \Omega(f_t) \tag{5}$$

The first- and the second-order gradient on $'l'$ are $\partial_{y'^{(t-1)}} l \left(y, y'^{(t-1)}\right)$ and $\partial^2_{y'^{(t-1)}} l\left(y, y'^{(t-1)}\right)$ denoted by $'g_i'$ and $'h_i'$, respectively. Hence, using second-order Taylor expansion, Eq. (5) can be written as,

$$\Phi^{(t)} = \sum_{i=1}^{n} \left[g_i f_t(x) + \frac{1}{2} h_i f_t(x^2) \right] + \Omega(f_t) \tag{6}$$

where g_i and h_i is the second-order gradient on ’l’. It is defined as $I_k = \{f_i | p(x_i) = k\}$ as the instance set of leaf ‘k’. Hence, we can write Eq. (6) by expanding ‘Ω’ as,

$$\begin{aligned}\Phi^{(t)} &= \sum_{i=1}^{n} [g_i f_t(x) + \frac{1}{2} h_i f_t\left(x\right)^2] + \lambda_1 T + \frac{1}{2}\lambda_2 \sum_{k=1}^{T} w_k^2 \\ &= \sum_{k=1}^{T} [\left(\sum_{i \in I_k} g_i\right) w_k + \frac{1}{2} (\sum_{i \in I_k} h_i + \lambda_1) w_k^2] + \lambda_2 T \end{aligned} \tag{7}$$

The optimal weight, w_k^* of leaf ‘k’ on a fixed structure $q(x)$ is given by,

$$w_k^* = -\frac{G_k}{H_k + \lambda_2} \tag{8}$$

whose value is computed by,

$$\Phi^* = -\frac{1}{2} \sum_{k=1}^{T} \frac{G_k^2}{H_k + \lambda_2} + \lambda_2 T \tag{9}$$

where $G_k = \sum_{i \in I_k} g_i$, $H_k = \sum_{i \in I_k} h_i$ and Φ represents the scoring function for the tree structure where smaller value means better tree structure. Both the gradient and second-order gradient statistics on each leaf are to be added before implementing the scoring algorithm to get the overall reliable score. The optimal split finding algorithm and loss reduction after the split is the key which is discussed later.

4 Proposed model

4.1 Data

For forecasting the overall electricity load in Tripura, original data from 2016 to 2019 have been collected from State Load Despatch Centre (SLDC), Tripura State Electricity Corporation Limited (TSECL), Agartala, which is the only source of electricity consumption data in Tripura. The weather data have been obtained from Modern-Era Retrospective Analysis for Research and Applications (MERRA)-2 meteorological data for latitude 23.831° north and longitude 91.287° east provided by National Aeronautics and Space Administration (NASA)/Goddard Space Flight Center [47]. All the data gathered from two different sources were preprocessed to ensure error-free prediction. The overall electricity load data from 2016 upto September 2019 had some missing values which were filled up by most probable average values manually. The following attributes were finally considered: datetime, temperature, pressure, humidity, air density, wind speed, wind direction, and Agt_load (total energy consumed). The distribution of the important parameters in the dataset is shown in Fig. 2.

4.2 Selection of model inputs

The Pearson correlation heatmap is used to detect the most influencing parameters of past data on future energy consumption. The electricity consumption load has, of course, the greatest similarities at the same hour of previous days (in case of 24 h ahead prediction). This plot tends to push for every hour of the day to build a model for 1-day, 1 week, and 1 month predictions. Therefore, the consumption load

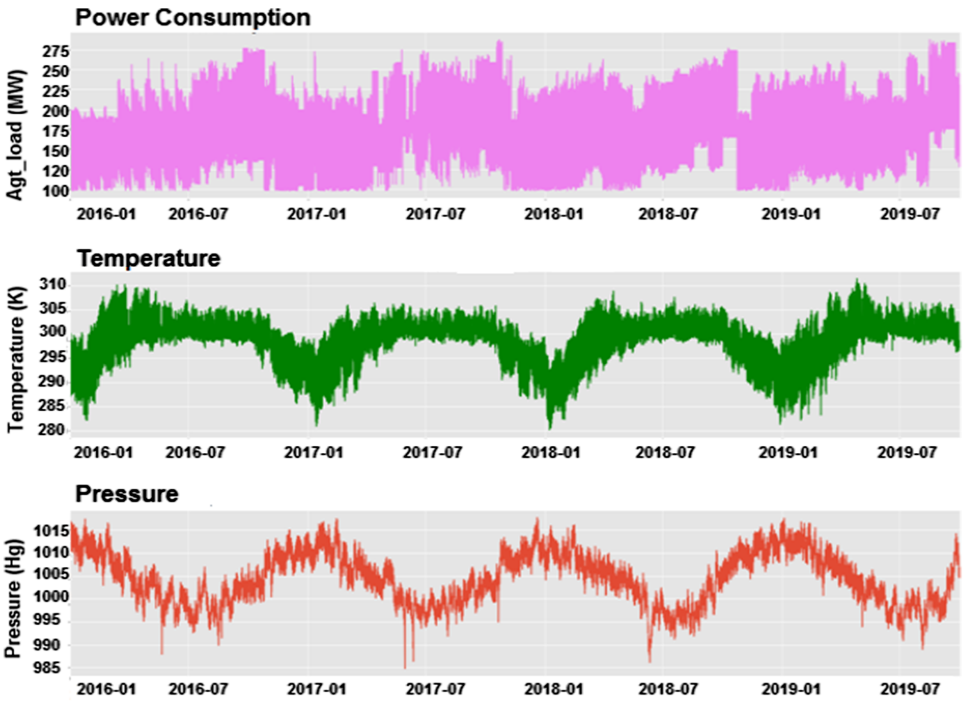


Fig. 2 The distribution of load, temperature, and pressure from 2016 to 2019

of all the days at every particular hour is selected as input of the model. In addition, two peak hours of a day (Hour_x and Hour_y), Day_of_week, Day_of_year, and peak seasons of the year (Day_of_year_x and Day_of_year_y) are chosen as inputs. The other method to ensure the collinearity between variables is the feature importance plot which is computed in terms of F-score. It is found that few parameters are weakly correlated enclosing poor F-score like Month, Day_of_week, etc. can be safely dropped. The parameters with a high score are usually chosen as the input of the model. However, the immunity of the machine to irrelevant inputs is ensured. The Pearson correlation heatmap and Feature importance are plotted and discussed in the subsequent sections.

4.3 Model selection and implementation

The prediction for electricity load in a state is very essential; this helps the operator conserve power and reduce waste. Because the noisy disturbance and predictability are not obvious, the accurate forecast of electricity load does not seem to be easy. In this paper, we propose a model for prediction of the electricity load in Agartala, Tripura based on a machine learning ensemble called RF-XGBoost. We have used the XGBoost regressor for this work because it was the fastest and strongest model for supervised learning to predict the use of electricity in Agartala. We have compared other approaches such as SVM, neural network, lone RF, and Adaboost to perform the prediction process but the results achieved were less accurate than results obtained with XGBoost. Also, the reasons for the selection of an XGBoost regressor include the ability to simultaneously predict future values of more than one variable and to model the nonlinear relationship in the data structure.

Initially, the dataset is split into training and test data in a fivefold CV strategy. Fast algorithms, like decision trees, are frequently seen in ensembles. The RF are decision tree ensemble that enhances the variance of base models by incorporating the principle of bagging with random subspaces (CART), enhancing this model 's efficiency. In the proposed model, multiple RF is built on the training data which is further trained by XGBoost. Generally, XGBoost is used to train gradient-boosted decision trees that perform well only on training data and sometimes cause overfitting because of the flexible model. Hence, in this study, RF is trained by XGBoost since RF randomly select data points when building trees and considers random subsets of features while splitting nodes. The optimal split finding algorithm, as well as the loss reduction after the split, are similar to the ideas referred in Ref. [10]. The best split implementation is carried out in XGBoost that supports the exact greedy algorithm. The following parameters are set for XGBoost training: learning rate (eta) = 1, booster = ‘gbtree’, ’subsample’ and ’colsample_bynode’ = 0.8. The ’max_depth’ = 5, ’num_parallel_tree’ = 100 is preferred. To prevent the model from boosting multiple random forests, num_boost_round is set to 1. The model has been further validated with a tenfold CV strategy, which is an established practice to evaluate model performance. The experiments are executed on Python environment on Windows GPU platform with an Intel Core

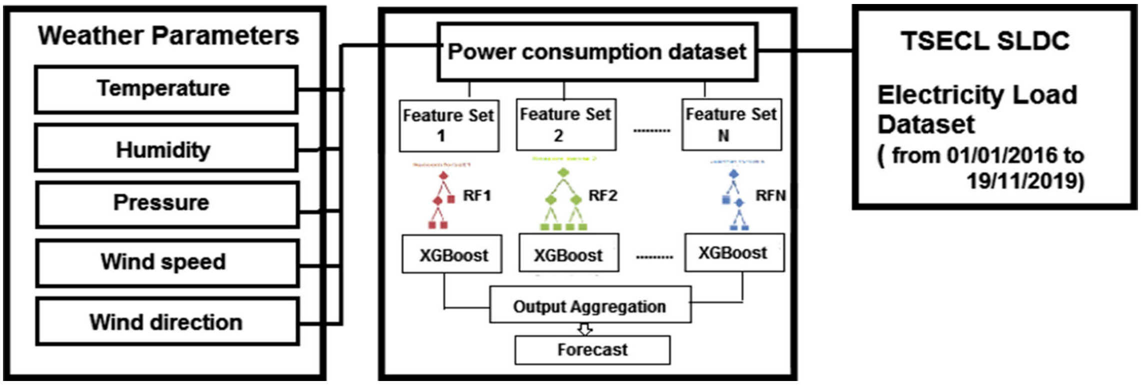


Fig. 3 The overall prediction model

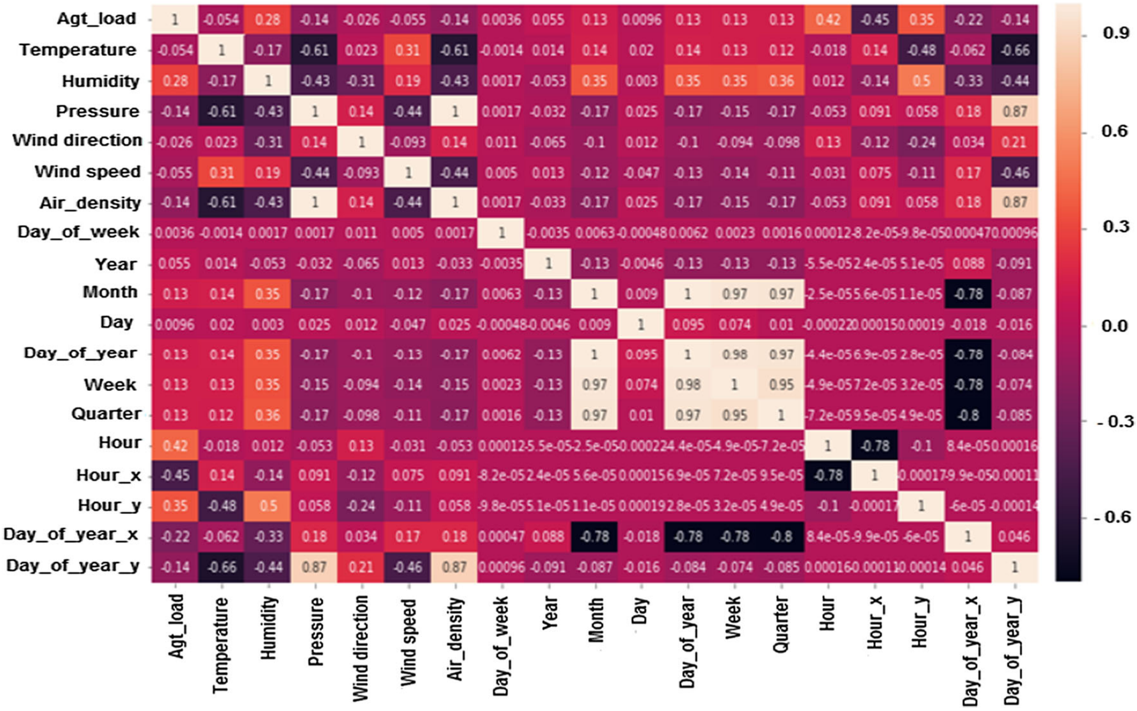


Fig. 4 Pearson correlation heatmap

i7-4790 processor, 3.6 GHz with 16 GB RAM. The overall prediction model is shown in Fig. 3.

5 Experimental results

Pearson correlation heatmap is plotted to obtain the correlations between the parameters of two datasets (i.e. dependencies among electricity load and weather parameters). Few parameters are excluded from the heatmap (shows weak relationships among the variables), while few attributes like year, month, day, week, hour, etc. are additionally plotted to search for correlations and added for future training. The plotted heatmap is shown in Fig. 4.

Feature importance graph corresponds to a technique that allows input parameters to be assigned a score to a prediction model that shows the relative significance of each variable to predict. The feature importance is therefore crucial for the parameters within the dataset to weed out irrelevant data and identify the most significant features in the dataset as shown in Fig. 5.

From the plotted heatmap and feature importance graph, attributes that have a considerably low influence on energy consumption are not included for training. From the correlation heatmap, it is revealed that there exist correlations among various parameters. Hence, a bivariate correlation is plotted in which the Agt_load is chosen as a function of all the

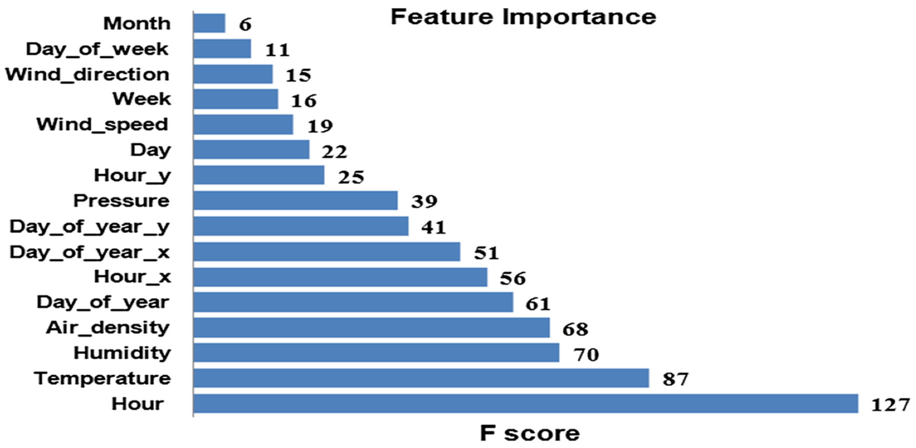


Fig. 5 Feature importance graph of parameters with F score

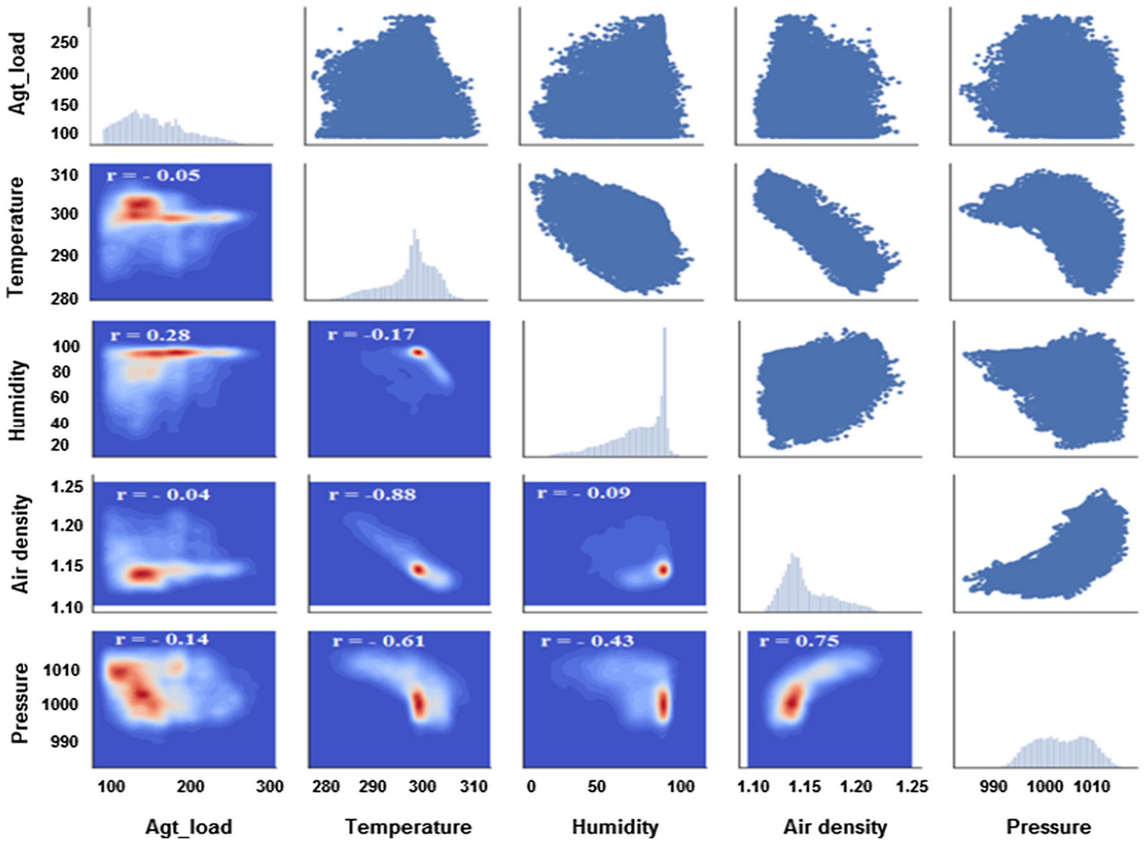


Fig. 6 Bivariate density matrix

attributes. The Bivariate matrix for major attributes of the dataset is shown in Fig. 6.

From the above correlation plot, it is expected to show strong linear correlation of Energy consumption with temperature humidity, air density, and pressure. Recalling the most correlated features, we plot Agt_load as a function of Temperature, Humidity, Air density, and Pressure on the various timescales (mean hourly and weekly) as shown in Fig. 7.

After obtaining the correlation of variables, the model is trained on selected parameters using the proposed RF-XGBoost ensemble with parameter settings as discussed earlier. Figure 8 shows the comparison of actual and predicted values of the model while attempting 24 h ahead forecast for 1 day (01/07/2019).

To predict energy consumption, a simple practice is followed. We separated the target variable i.e. ‘Agt_load’ from the features matrix containing Temperature, Humidity, Air density, Pressure, Hour etc. for representing strong correlations in the graph. Then, a fivefold CV procedure was executed on the model for predicting future energy consump-

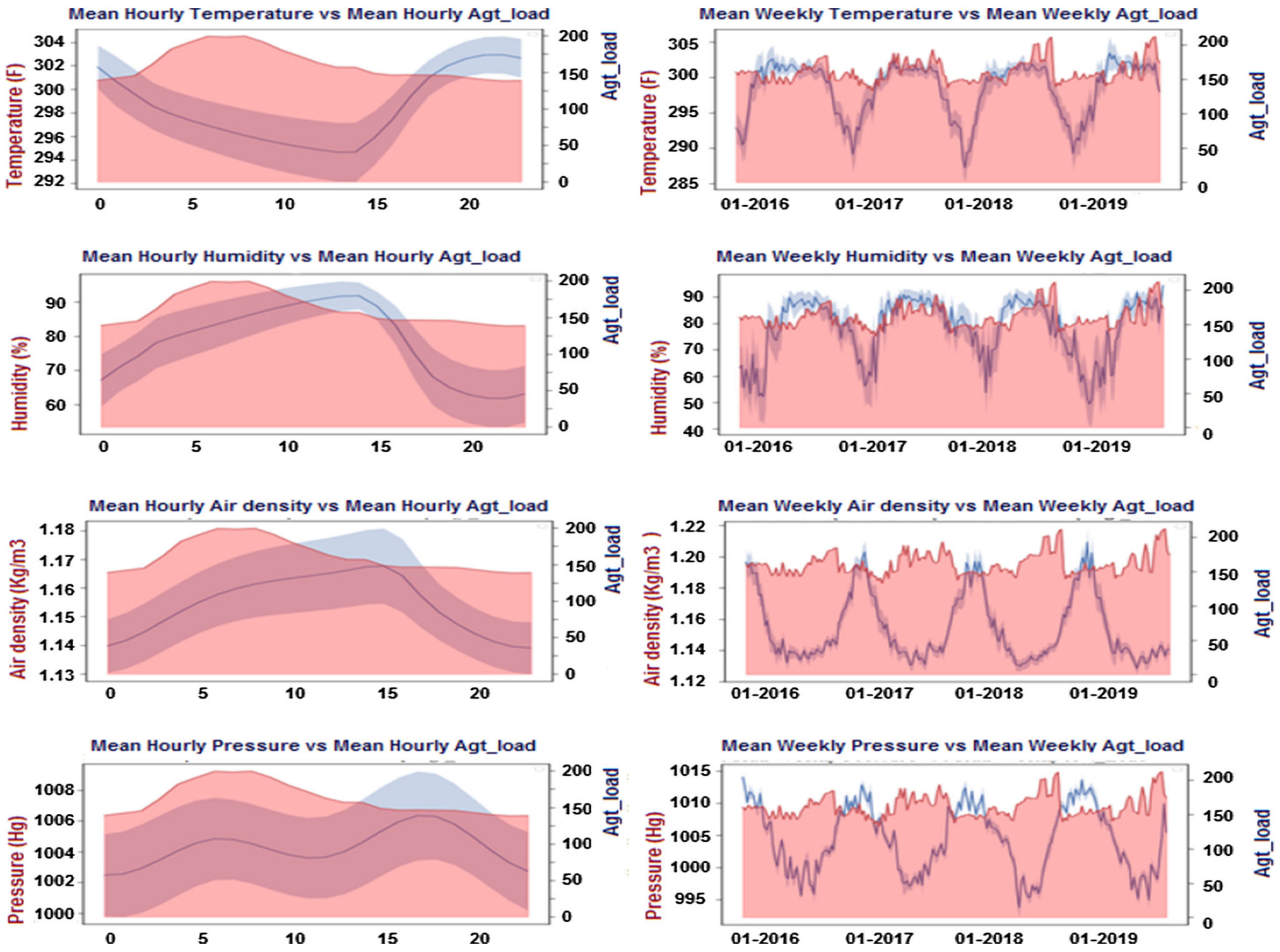
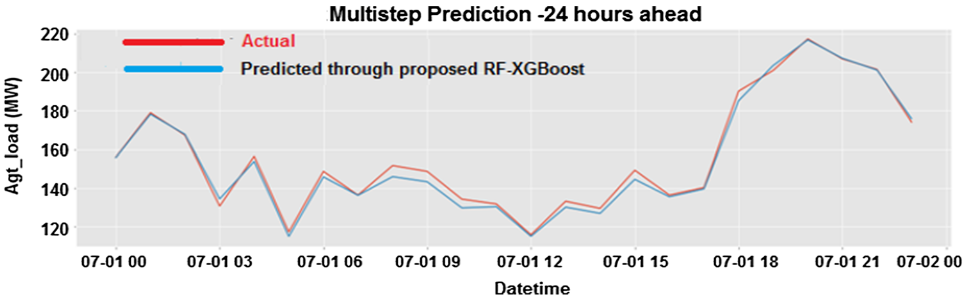


Fig. 7 Graph of Agt_load with Temperature, Humidity, Air density and Pressure on the various timescales

Fig. 8 One day (24 h) ahead forecast for 1 day (01/07/2019)



tion (i.e. Agt_load) for multiple time horizons separately. The graphical comparison of actual and predicted values of Agt_load for multiple time horizons (i.e. 1 day, 1 week, and 1 month) is shown in Fig. 9.

5.1 Evaluation of model performance

To assess the execution of the proposed model, mainly two different measures that best suits the model (RF_XGBoost) are used, the coefficient of determination (R^2) and root mean square error (RMSE) [48–50]. R^2 signifies the percentage variation of prediction values whose range is between 0 and 1. It is given by,

$$R^2 = 1 - \frac{\sum(\text{Actual} - \text{Predicted})^2}{\sum(\text{Actual} - \text{Mean\_actual})^2} \qquad (10)$$

RMSE is the variability test of the predicted values and the actual values of the model. It is defined as the square root of the mean squared error given by,

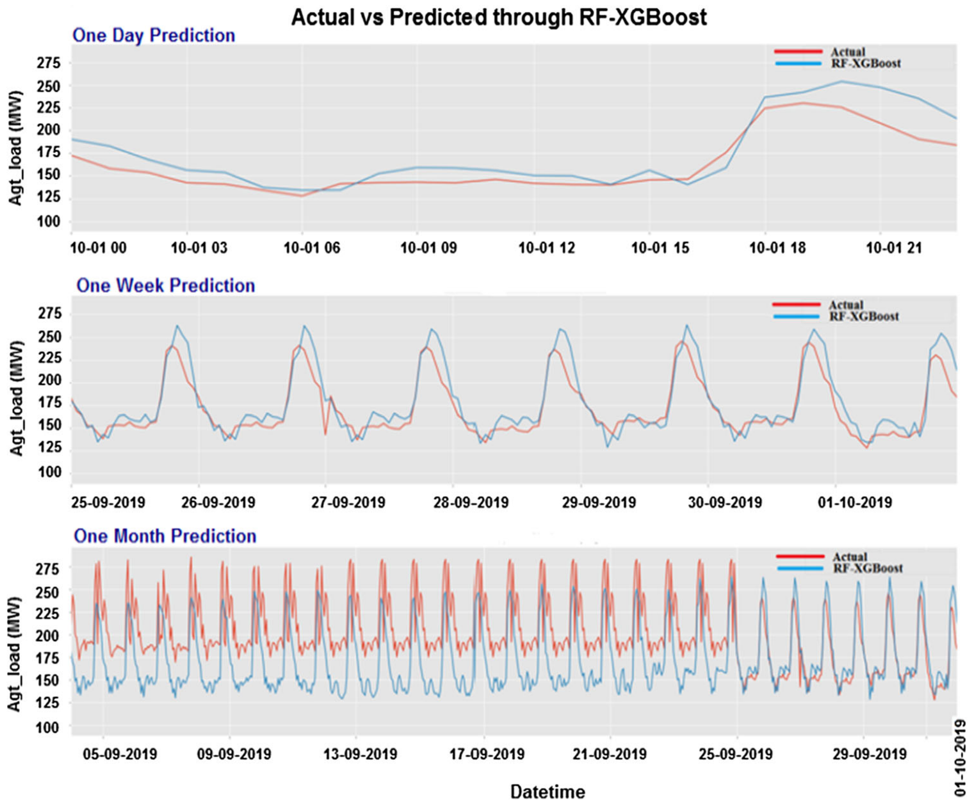


Fig. 9 Comparison of actual and predicted values for 1 day, 1 week, and 1 month

Table 1 Performance evaluation of RF-XGBoost model with other standard models for 24h, 1 week, and 1 month electrical energy consumption prediction

Model	1 month prediction		1 week prediction		24-h prediction	
	R^2-score	RMSE	R^2-score	RMSE	R^2-score	RMSE
Support vector regression	0.277	4.913	0.276	4.915	0.286	4.856
Neural network	0.367	3.461	0.371	3.312	0.379	3.241
Random forest	0.576	3.219	0.580	3.111	0.588	3.172
Tuned random forest	0.580	3.221	0.581	3.213	0.624	2.817
AdaBoost regressor	0.659	3.992	0.667	2.917	0.725	2.686
XGBoost	0.711	2.914	0.713	2.886	0.762	2.463
RF-XGBoost (proposed)	0.814	1.997	0.879	1.132	0.914	0.955

$$\text{RMSE} = \sqrt{\sum \frac{(\text{Predicted} - \text{Actual})^2}{N}}. \tag{11}$$

A comparison of the different models in terms of R^2 and RMSE for 24h, 1 week, and 1 month prediction is shown in Table 1.

The comparative analysis resulted in a higher rating of the proposed RF-XGBoost ensembles than other models for the short-term (1 day–1 week) and the long-term (1 month) electricity consumption forecast. Therefore, for both short- and long-term electrical energy consumption predictions, the proposed RF-XGBoost ensemble seems to be a superior option.

6 Conclusions and future work

This paper proposes a new ensemble model for the prediction of overall electrical energy consumption in Tripura. The ensemble of RF and XGBoost, which are among machine learning techniques, are used to create the model. The model has examined multiple parameters such as Temperature, Humidity, Air density, and Pressure which could affect the electricity load directly and shown to be a useful and promising criterion in the current prediction of overall electrical energy consumption. Also, it showed how data analyses differentiated by time of the day could significantly improve the accuracy as reflected in the test results. The study attempted to advance the new idea of using RF instead of decision trees with XGBoost. The R^2 score of 0.914 reflects the accuracy and revealed that this approach was successful to provide valuable statistics in connection with the Tripura power consumption prediction. The proposed model has been compared with other renowned machine learning algorithms, including SVR, NN, and Adaboost on the same data using various statistical measures. The obtained result suggests several conclusions of RF-XGBoost ensemble: (1) The efficiency in comparison with single structure or other analogous methods is much higher. (2) The model is suitable for multiple time horizon (short-term, medium-term, and long-term) predictions.

We are presently investigating various options of integrating renewable sources of power and accumulating data to extend this work to improve its accuracy. We are also exploring other possible ensembles of machine learning towards the above-mentioned areas.

Acknowledgment We thank Tripura State Electricity Corporation Limited (TSECL), State Load Despatch Centre (SLDC), Agartala, for providing data for hourly electricity load in the state.

Funding The author(s) received no financial support for the research, authorship, and/or publication of this article.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest

References

1. Su Y-W (2019) Residential electricity demand in Taiwan: consumption behavior and rebound effect. Energy Policy 124:36–45
2. Shahbaz M, Sarwar S, Chen W, Malik MN (2017) Dynamics of electricity consumption, oil price and economic growth: global perspective. Energy Policy 108:256–270
3. Chen H-Y, Lee C-H (2019) Electricity consumption prediction for buildings using multiple adaptive network-based fuzzy inference system models and gray relational analysis. Energy Rep 5:1509–1524
4. Kulkarni S, Simon SP, Sundareswaran K (2013) A spiking neural network (SNN) forecast engine for short-term electrical load forecasting. Appl Soft Comput 13(8):3628–3635
5. Tiana Y, Yu J, Zhao A (2020) Predictive model of energy consumption for office building by using improved GWO-BP. Energy Rep 6:620–627
6. Yoo SG, Hernandez-Alvarez M (2018) Predicting residential electricity consumption using neural networks: a case study. IOP Conf Ser J Phys Conf Ser 1072(18):012005. https://doi.org/10.1088/1742-6596/1072/1/012005
7. Cai H, Shen S, Lin Q, Li X, Xiao H (2019) Predicting the energy consumption of residential buildings for regional electricity supply-side and demand-side management. IEEE Access 7:30386–30397. https://doi.org/10.1109/ACCESS.2019.2901257
8. Tso G, Yau K (2007) Predicting electricity energy consumption: a comparison of regression analysis, decision tree and neural networks. Energy 20:1761–1768
9. Kankal M et al (2011) Modeling and forecasting of Turkey's energy consumption using socioeconomic and demographic variables. Appl Energy 88:1927–1939
10. Chen T, Guestrin C (2016) XGBoost: a scalable tree boosting system. In: Proceedings of the 22nd ACM SIGKDD International Conference on knowledge discovery data mining, San Francisco, CA, USA, pp 785–794
11. Google Maps (2020) Location of Agartala. https://www.google.com/maps/place/Agartala,+Tripura/@25.2810646,78.5062462,4z/. Accessed 8 June 2020
12. Gross G, Galiana FD (1987) Short-term load forecasting. Proc IEEE 75:1558–1573
13. Hyde O, Hodnett PF (1997) An adaptable automated procedure for short-term electricity load forecasting. IEEE Trans Power Syst 12:84–94
14. Broadwater RR, Sargent A, Yarali A et al (1997) Estimating substation peaks from load research data. IEEE Trans Power Deliv 12:451–456
15. El-Keib AA, Ma X, Ma H (1995) Advancement of statistical based modeling techniques for short-term load forecasting. Electr Power Syst Res 35:51–58
16. Huang SR (1997) Short-term load forecasting using threshold autoregressive models . IEEE Proc Gener Transm Distrib 144:477–481
17. Goia A, May C, Fusai G (2010) Functional clustering and linear regression for peak load forecasting. Int J Forecast 26(4):700–711
18. Amral N, Ozveren C, King D (2007) Short term load forecasting using multiple linear regression. In: UPEC 2007. 42nd international universities power engineering conference, 2007, pp 1192–1198
19. Pappas S, Ekonomou L, Karamousantas D, Chatzarakis G, Katsikas S, Liatsis P (2008) Electricity demand loads modeling using autoregressive moving average (ARMA) models. Energy 33(9):1353–1360
20. Lee C-M, Ko C-N (2011) Short-term load forecasting using lifting scheme and ARIMA models. Expert Syst Appl 38(5):5902–5911
21. Mastorocostas P, Theocharis J, Bakirtzis A (1999) Fuzzy modeling for short term load forecasting using the orthogonal least squares method. IEEE Trans Power Syst 14(1):29–36
22. Mandal P, Senjyu T, Funabashi T (2006) Neural networks approach to forecast several hour ahead electricity prices and loads in deregulated market. Energy Convers Manag 47(15–16):2128–2142
23. Senjyu T, Takara H, Uezato K, Funabashi T (2002) One-hour-ahead load forecasting using neural network. IEEE Trans Power Syst 17(1):113–118
24. Lin C-T, Chou L-D (2013) A novel economy reflecting short-term load forecasting approach. Energy Convers Manage 65:331–342
25. Wu C-H, Tzeng G-H, Lin R-H (2009) A novel hybrid genetic algorithm for kernel function and parameter optimization in support vector regression. Expert Syst Appl 36(3):4725–4735
26. Wang J, Li L, Niu D, Tan Z (2012) An annual load forecasting model based on support vector regression with differential evolution algorithm. Appl Energy 94:65–70
27. Nagi J, Yap KS, Nagi F, Tiong SK, Ahmed SK (2011) A computational intelligence scheme for the prediction of the daily peak load. Appl Soft Comput 11(8):4773–4788
28. Cheng Y-Y, Chan P, Qiu Z-W (2012) Random forest based ensemble system for short term load forecasting. In: 2012 international conference on machine learning and cybernetics (ICMLC), vol. 1. pp 52–56
29. Krawczak M, Popchev I, Rutkowski L et al (2015) Intelligent systems'2014. Adv Intell Syst Comput 323:821–828
30. Khayatian F, Sarto L, Dall'O' G (2016) Application of neural networks for evaluating energy performance certificates of residential buildings. Energy Buildings 125:45–54
31. Ascione F, Bianco N, De Stasio C, Mauro GM, Vanoli GP (2017) Artificial neural networks to predict energy performance and

retrofit scenarios for any member of a building category: a novel approach. Energy 118:999–1017

32. Chen Y, Tan H (2017) Short-term prediction of electric demand in building sector via hybrid support vector regression. Appl Energy 204:1363–1374
33. Rastogi P, Polytechnique E, Lausanne FD (2017). Gaussian-process-based emulators for building performance simulation. In: Building simulation 2017: the 15th international conference of IBPSA. IBPSA, San Francisco
34. Papadopoulos S, Azar E, Woon W-L, Kontokosta CE (2017) Evaluation of tree-based ensemble learning algorithms for building energy performance estimation. J Build Perform Simul 1493:1–11
35. Wang Z, Wang Y, Zeng R, Srinivasan RS, Ahrentzen S (2018) Random forest based hourly building energy prediction. Energy Build 171:11–25
36. Ahmad MW, Mourshed M, Rezgui Y (2017) Trees vs neurons: comparison between random forest and ANN for high-resolution prediction of building energy consumption. Energy Build 147:77–89
37. Lee D, Kang S, Shin J (2017) Using deep learning techniques to forecast environmental consumption level. Sustainability 9:1894
38. Li C, Ding Z, Zhao D, Yi J, Zhang G (2017) Building energy consumption prediction: an extreme deep learning approach. Energies 10:1525
39. Kim J-Y, Cho S-B (2019) Electric energy consumption prediction by deep learning with state explainable autoencoder. Energies 12(4):739. https://doi.org/10.3390/en12040739
40. Garcia-Martin E, Rodrigues CF, Riley G, Grahn H (2019) Estimation of energy consumption in machine learning. J Parallel Distrib Comput 134:75–88
41. Mosavi A, Bahmani A (2019) Energy consumption prediction using machine learning; a review. https://doi.org/10.20944/preprints201903.0131.v1. Preprints 2019030131
42. Zhang C, Ma Y (2012) Ensemble machine learning: methods and applications. Springer, Berlin
43. Valentini G, Dietterich TG (2004) Bias-variance analysis of support vector machines for the development of SVM- based ensemble methods. J Mach Learn Res 5:725–775
44. Leggio T (2019) Bias-variance trade-off 101. https://medium.com/opex-analytics/bias-variance-trade-off-101-7d3aae4485a8. Accessed 20 June 2020
45. Friedman JH (2001) Greedy function approximation: a gradient boosting machine. Ann Stat 29(5):1189–1232
46. Breiman L, Freidman JH, Olshen RA, Stone CJ (1984) Classification and Regression Trees. CRC Press, London
47. Global Modeling and Assimilation Office (GMAO) (2015) MERRA-2 tavg1_2d_slv_Nx: 2d,1-hourly,time-averaged, single-level, assimilation, single-level diagnostics V5.12.4, Greenbelt, MD, USA, Goddard Earth Sciences Data and Information Services Center (GES DISC). https://doi.org/10.5067/VJAFPLI1CSIV. Accessed 06 Dec 2019
48. Chai T, Draxler RR (2014) Root mean square error (RMSE) or mean absolute error (MAE)? Arguments against avoiding RMSE in the literature. Geosci Model Dev 7:1247–1250
49. Luca Massidda L, Marrocu M (2018) Quantile regression post-processing of weather forecast for short-term solar power probabilistic forecasting. Energies 11(7):1–20
50. Salami AA, Ajavon ASA, Dotche KA, Bedja K-S (2018) Electrical load forecasting using artificial neural network: the case study of the grid inter-connected network of Benin electricity community (CEB). Am J Eng Appl Sci 11(2):471–481

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