Journal of Scientific & Industrial Research
Vol. 80, October 2021, pp. 894-904

Stacking Machine Learning Models to Forecast Hourly and Daily Electricity Consumption of Household Using Internet of Things

Alisha Banga*, Ravinder Ahuja and S C Sharma

Electronics and Computer Discipline, IIT Roorkee, Saharanpur Campus, Saharanpur, UP, India 247 001

Received 26 October 2020; revised 01 April 2021; accepted 17 August 2021

The objective of this paper is to design an efficient electricity consumption forecasting model using stacking ensemble technique and Internet of Things (IoT). Two stage process is applied in this paper. In the first stage, fifteen forecasting models (Auto-ARIMA, Holt-Winter (Additive), Exponential, Facebook Prophet, Light Gradient Boosting, AdaBoost, Support Vector Regression, Decision Tree, Extra Tree, Random Forest, Elastic net, K-Nearest Neighbour's, XGBoost, Linear Regression, Long Short Term Memory) are applied to forecast electricity consumption at an hourly and daily level. In the next stage, the best four models are selected and stacked. We have considered the dataset of energy consumption by electrical appliances per minute in a house over seven days. The models are evaluated using root mean square error (RMSE), mean absolute error (MAE), R-square, and mean absolute percentage error (MAPE). The results show that the extra tree performed better among all the algorithms, and stacking further improves performance. Elastic net and decision tree algorithms have taken less time as compared to other models applied in this study.

Keywords: Electricity Demand, Ensemble Learning, Regression, Smart Meter, Time Series Forecasting

Introduction

Electricity is the lifeline of today's world. Due to the increase in economic growth, population, industrialization, living standard the electricity consumption has increased.[1] The study[2] on 160 countries over 30 years reveals a correlation between electricity consumption and the nation's economic growth. As per the World Energy Outlook[3], the global energy demand will increase by 1.0% compound annual growth rate during 2016-2040. Approximately 27% of universal energy is consumed in residential buildings.[4] The economic growth of a country mainly depends upon manufacturing industries. These industries consume a large amount of electricity. A shortage of power can cause a decrease in production, which results in financial loss. However, surplus energy can also create problems. The electricity is generated from non-renewable resources like oil and coal. Electricity production in excess can cause wastage of natural resources (coal and oil). So, proper planning of electricity consumption and supply can maintain the country's economic stability/growth. Electricity load forecasting plays a vital role in maintaining the balance between electricity consumption and requirement. Load forecasting would optimize energy consumption and plan household energy needs accordingly, saving solar energy and utilizing it optimally. By using IoT smart meter data, load forecasting at hospitals, commercial buildings, universities, and substation is extensively studied in the literature.[5,6] The electricity consumption can be monitored at house-level/room level/specific appliances with IoT devices advancement. The electricity companies can also use smart meter data in residential areas/commercial areas/cities to understand the electricity requirement. The companies can utilize electricity forecasting values to address the gap of demand and supply, for deciding tariffs[7], to know the electricity consumption trends at a specific time. Another application of a good forecasting model is to save money and energy. The amount of money that can be saved by reducing a 1% error in forecasting for the U.K. power system is £10 million.[8] Designing an electricity forecasting model is not an easy job because it depends upon other factors like weather, season, and occupancy in the house. Thus it is required to design a forecasting model which can forecast electricity demand at household level efficiently so that utility companies can utilize power generation plants maximally, to save energy and money, to avoid wastage of natural resources (coal

*Author for Correspondence
E-mail: alishabanga47@gmail.com

and oil). We have forecasted a single house's electricity consumption using the ensemble of forecasting algorithms. In this paper we have applied fifteen forecasting models from three categories (classical, machine learning, and deep learning) to find the best four models. Further, stacking ensemble is applied on the best four performing models in place of weak models, which improves the performance. We have used MAE, MAPE, and RMSE to evaluate the applied algorithms.

The work done in this paper is as follows:

(i) We have applied various data cleaning processes (missing values imputation, re-sampling, replacing categorical attributes with numeric values, time in seconds to time date format, removing irrelevant features, and normalization) to make the data ready for forecasting models.

(ii) We have applied three different types of models (classical, machine learning, and deep learning)-total of fifteen. Further, Stacking ensemble is applied on the best four performing models which improves the performance of forecasting.

(iii) The experimental results shows that stacking ensemble performed better among all the models and state-of-the-art results in terms of MAE, RMSE, and MAPE performance parameters.

Existing Electricity Forecasting Approaches

Three approaches used to forecast electricity usage are (i) Statistical method, (ii) Machine Learning, and (iii) Deep learning. Indikawati *et al.*[9] have applied the XGBoost gradient boosting algorithm to predict households' electricity consumption data like a furnace, fridge, kitchen, etc. They have computed the root mean square error of the algorithm. Bansal *et al.*[10] have taken data from the DECC website and EMA Singapore to predict smart meter electricity consumption. They have applied a boosted decision tree algorithm, and hyper-parameters are optimized using a random search algorithm. They evaluated the model based on R-square and root mean square error. Rodrigues *et al.*[11] have applied Artificial Neural Networks (ANN) to predict electricity consumption at hourly and daily level. They have considered the data of 93 houses in Portugal over six months duration. Yildiz *et al.*[12] have applied k-means clustering and SOM to cluster the data into user profiles. The data has a large dimension, and for reducing the dimension, the PCA algorithm is applied. After these steps, the CART classification algorithm was applied. The performance was evaluated on the normalized RMSE and MAE. Shirke *et al.*[13] have applied five regression algorithms namely, logistic regression, gradient boosting, random forest, extreme gradient Boosting, and support vector machine to forecast the energy usages of households. The models were evaluated based on RMSE, MAE, R-Square, and MAPE. Khan *et al.*[14] have applied the ensemble of CNN and LSTM-AE to forecast energy requirements in residential and commercial buildings. They have taken data from the UCI machine learning repository containing 2,075,269 records (measured over a minute) from 2006 to 2010. The data were resampled for an hour and daily basis. The models were evaluated based on MSE, RMSE, MAE, and MAPE. Rajabi & Estebsari[15] have applied CNN (Convolution Neural Network 1-D and 2-D), artificial neural network, and support vector machine to forecast the load of the residential home. They have found that CNN-2D has performed better among all the algorithms considered.

Wu *et al.*[16] have applied Gaussian process, support vector regression, linear regression, LSTM, and BMKR (Boosting Multiple Kernel Regression) models on the two datasets to forecast electricity consumption. They proposed two versions of BMKR; one is homogeneous in which feature space of domain and target are the same. Another one is heterogeneous in which domain and target are different. Kim & Cho[17] have applied GRU, Bi-LSTM, LSTM, LSTM with Attention mechanism, and a combination of CNN-LSTM architecture to forecast electricity consumption in a residential house. Along with these deep learning algorithms, they have applied four machine learning algorithms also. They have considered dataset over four year's duration. Bogomolov *et al.*[18] have applied the random forest algorithm to predict electricity consumption at the day and week levels. They have reduced the dimension of the dataset to 32 without any performance loss. Chen *et al.*[19] have applied SVM model to forecast electricity load in office buildings. Fong *et al.*[20] have proposed method to remove noise using misclassified recall. After pre-processing (removal of noise) decision tree is applied to predict high energy usage by appliance in a building. They have found that their model performed best among all applied in terms of Kappa, accuracy, and running time. They have not explored the capability of ensemble techniques. Dey *et al.*[21] have applied 10 algorithms for forecasting electricity consumption of household appliances and compared their performances on the basis of RMSE. They found that GMDH-MIA-Linear algorithm is performing best among all on the basis of RMSE. They have not applied ensemble techniques which may improve the performance.

Meher[22] have applied VEC, ARIMA, and VAR models to know future demand of the electricity of residential from Odisha (India). He found that VAR is the best performing model because it gives less error as compared to others. He has not explored deep learning, and machine learning models which may predict more accurately. Estebsari & Rajabi[23] have applied image encoding technique on electricity consumption dataset and further CNN algorithm is applied to forecast electricity consumption. They have applied various techniques of image encoding and found recurrence plot as the best performing model. Mukherjee *et al.*[24] have applied KNN, SVR, Logistic Regression, and random forest models to find suitable algorithm for forecasting electricity. They have found that KNN algorithm performed better among all the models applied. Andriopoulos *et al.*[25] have proposed a model based on CNN and statistical properties of the time series dataset. The statistical properties of the time series dataset are explored to tune hyper parameters of the CNN model. The optimized model is used to forecast electricity consumption at hourly level for the household. They have considered only one model namely CNN, others may be explored to find the best one. Gerossier *et al.*[26] have proposed a probabilistic model to forecast electricity demand for a household at hourly level. The RMSE value of their model is 0.289 and MAPE value is 0.38. They have applied only one model to forecast, others may be tested to find the best one.

In the literature survey, the forecasting of electricity usages in commercial and/or residential buildings is done through deep learning and machine learning algorithms. The gap is that not much forecasting models and their ensemble are explored in the past. This paper presents an exploratory analysis of fifteen forecasting models (Auto-ARIMA, Simple Exponential, Prophet, Holt-Winter, Random Forest, Extra Tree, Support Vector Machine, Decision Tree, Adaboost, XGBoost, LightGBM, Linear Regression, K-Nearest Neighbour, Elastic net, and LSTM). Stacking ensemble of best four models is applied to improve the performance.

Methods and Materials

In this section, we will discuss about the dataset considered, data cleaning process, and algorithms applied.

Dataset Description

We have considered a dataset consisting of electricity consumption by residential households from Jan 01st, 2016 to Jan 07th, 2016 duration. The data was captured over a granularity of one minute. The dataset is publicly available at the Kaggle website.[27] It consists of 32 attributes (19 related to electricity and 13 related to weather). The total number of records present is 503910. The electricity consumption is observed over one-minute granularity. The dataset description is given in Table 1. The gen attribute has a correlation value of 1 with solar as shown in Fig. 1. Similarly, the House overall attribute is having a correlation value of 1 with use attribute. That's why gen and house overall attributes are removed from the dataset.

Data Cleaning Process

To prepare the data for forecasting, various preprocessing techniques are applied, which are as follows: (i) the time was given in the form of seconds, we have converted it into Date Time format (ii) attributes like gen and solar have similar values, so we find out such attributes in the dataset and eliminated. (iii) In the dataset, various attributes are representing similar appliances like furnance1, furnance2. We have combined such attributes into a single attribute. (iv) The missing values were present in the dataset, which have been filled by the

Table 1 — Dataset Description

Description	Attributes	Total Records	No. of Attributes
Jan 01st, 2016 to Jan 07th, 2016 Per-minute observation	**Electricity related:** time, use (kW), kitchen 14, House overall (kW), Living Room Dishwasher (kW), Furnance1, Fridge, Garage door, Furnance2 kitchen 12, kitchen 38, Barn, Well, Microwave, Solar, gen (kW), Home office, Wine cellar. **Weather-related:** dewPoint, temperature, cloudCover, pressure, apparentTemperature, humidity, visibility, icon, summary, precipIntensity, windBearing, precipProbability.	503910	Total: 32 18 Attributes related to electricity 13 attributes related to weather 1-time attribute

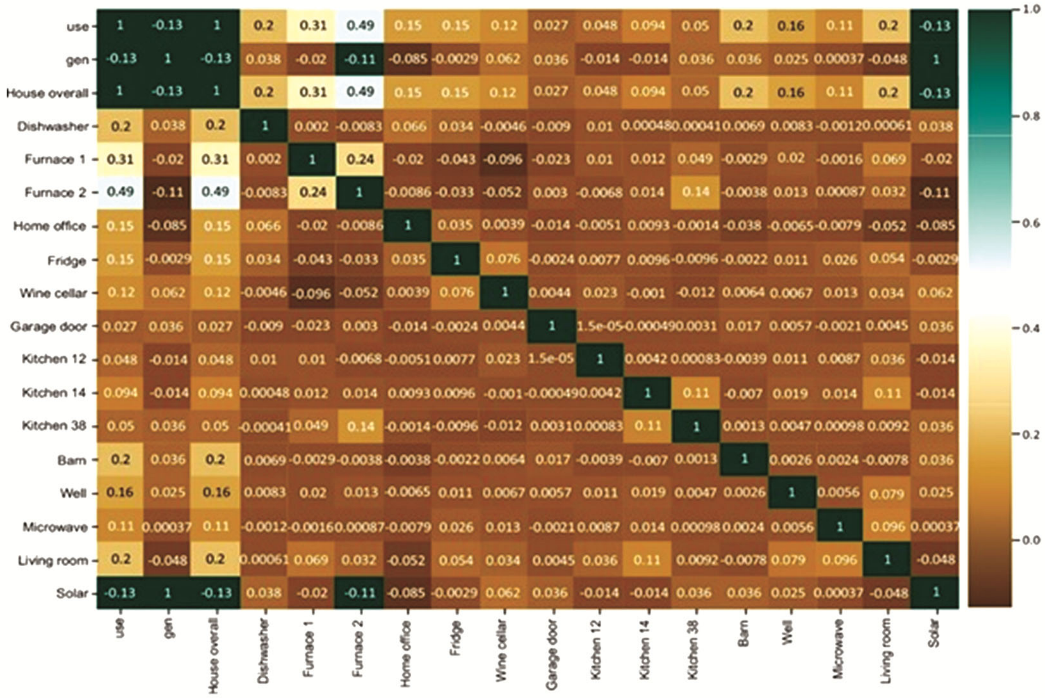

Fig. 1 — Correlation map of dataset attributes

backfilling method (v). The categorical attributes are converted into numeric values (vi) The original data was observed at a minute level. We resampled the data at the hour level and the day level.

Methods Applied

Firstly, we have preprocessed the data by converting the time attribute to time date format consisting of date, day, year, and month. The attributes with missing values are eliminated from the dataset. There are two attributes- furnace1 and furnance2; which have been combined to form a single attribute. The dataset contains missing values that have been replaced with the backfilling method. The data were resampled to day level and hour level. The categorical variables are converted into numerical variables. Once the data is ready for forecasting, there are two stages of operations. In stage one, classical, machine learning, and deep learning forecasting algorithms — total fifteen (Auto-ARIMA, Exponential, Facebook Prophet, Holt-Winter, linear regression, random forest, decision tree, extra tree, light gradient boosting, support vector machine, elastic net regression, Elastic net, XGBoost, Adaboost, and LSTM) are applied to predict the energy consumption of a household. The algorithms are compared on the basis of RMSE, MAE, R-square, and MAPE. In the second stage, the best four algorithms are picked, and further stacking of these algorithms is done for predicting electricity consumption. We found that stacking improved performance. The methodology used is shown in Fig. 2. The various algorithms applied are as follows:

Auto-ARIMA (Auto-Regressive Integrated Moving Average)

It is a stochastic model used to forecast future values based on previous values.[28] ARIMA is represented by three parameters (p, d, and q). Where q is moving average p is autoregressive, and d is the degree of differencing for handling seasonality. In Auto-ARIMA, a range of attributes are given, and it finds the optimal value from the given range.

Facebook Prophet

Prophet is an open-source tool developed by Facebook for time series forecasting.[29] Forecasting is done based on the additive model where non-linear trends fit weekly, daily, yearly, and holiday effects. This model can handle data with outliers, noise, and unpredictable changes. This model allows the domain expert to adjust the parameters for better results. Parameters used in our study are: mcmc_samples = 300, holidays_prior_scale = 0.25, seasonality_mode =

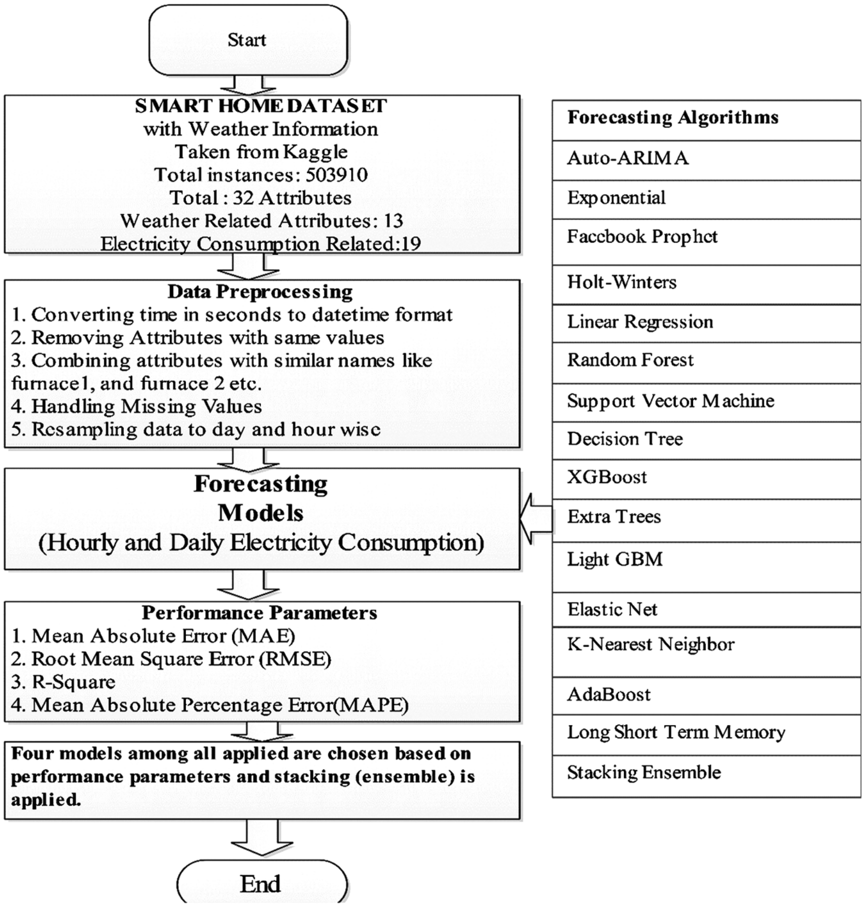
Start
SMART HOME DATASET
with Weather Information
Taken from Kaggle
Total instances: 503910
Total : 32 Attributes
Weather Related Attributes: 13
Electricity Consumption Related:19
Data Preprocessing
1. Converting time in seconds to datetime format
2. Removing Attributes with same values
3. Combining attributes with similar names like furnace1, and furnace 2 etc.
4. Handling Missing Values
5. Resampling data to day and hour wise
Forecasting Models
(Hourly and Daily Electricity Consumption)
Performance Parameters
1. Mean Absolute Error (MAE)
2. Root Mean Square Error (RMSE)
3. R-Square
4. Mean Absolute Percentage Error(MAPE)
Four models among all applied are chosen based on performance parameters and stacking (ensemble) is applied.
End
Forecasting Algorithms
Auto-ARIMA
Exponential
Facebook Prophet
Holt-Winters
Linear Regression
Random Forest
Support Vector Machine
Decision Tree
XGBoost
Extra Trees
Light GBM
Elastic Net
K-Nearest Neighbor
AdaBoost
Long Short Term Memory
Stacking Ensemble

Fig. 2 — The approach used for forecast electricity consumption on hourly and daily basis

'additive', seasonality_prior_scale = 0.4, weekly_seasonality = True, changepoint_prior_scale = 0.01.

Exponential Smoothing

This technique is applied to the time-series data to convert it into smoothed data. It assigns exponentially decreasing weights to the data which gets older.[30] The more recent data is given higher weightage, and older data is given a lower weightage. The way of forecasting using exponential smoothing is given in Eq. 1.

$$y_f = \alpha . y_{t-1} + (1 - \alpha) . yat_{t-1} \qquad \text{... (1)}$$

where, y_f is the forecasted value at time t, α is the smoothing factor; its value varies between 0 and 1. y_{t-1} is the actual data of the previous period, the yhat_{t-1} forecast for the period before the current period. α close to 1 means the large influence of previous data, and close to 0 means less influence of the past data.

Holt-Winter (Additive)

Holt's equation was generalized by winters by adding seasonality in the Holt's equation. The seasonality can be of two types: additive and multiplicative.[31] In this model, the recent values are given higher weightage as compared to older values. The additive model is given in Eq. 2.

$$y_t = b + l * t + st + re \qquad \text{... (2)}$$

where, y_t is the forecasted value at time t, b is the base value, l is the linear trend, st is the seasonal additive factor, and re is random error.

Linear Regression

Linear regression is used to find a direct relationship between dependent and independent variables.[32] The association is given in Eq. 3.

$$Y = mX + b \qquad \text{... (3)}$$

where, Y is the target variable, X is the dependent variable, and b is constant. There can be a positive and negative relationship between the variables.

Support Vector Regression

It is a technique that belongs to the supervised machine learning category[33] and can be used for both regression analysis and classification. The objective is to find a function f(x) such as a minimum deviation from the outcome variable. It can be written in the form of an optimization function, as given in Eq. 4. Parameters used in our study are kernel='rbf' and degree=1

$$min\frac{1}{2}\|w\|^2 \qquad \text{... (4)}$$

Constraints are $y_i - wx_i$-$b\leq$ · and wx_i+b-$y_i \leq$ ·

Decision Tree Regression

This is a supervised algorithm that can be used for regression as well as classification. The tree-like structure is formed in this model.[34] It divides the dataset into smaller parts and, at the same time, builds a tree structure incrementally. Internal nodes in the tree represent the test on the dataset's features and leave nodes to represent the output. We have used max_features = 12.

Random Forest

This model[35] is used for regression and classification both. This model is based on multiple decision trees. Every decision tree learns from random samples of training data with bootstrapping. Only some of the features are shown to each tree for splitting a node. The average output of each decision tree is calculated for the final output. This algorithm is robust and easy to use. We have used 300 number of estimators and 1234 as random state hyper parameters in this study.

XGBoost Algorithm

This is an ensemble of weak learning algorithms prone to overfitting, and combining weak algorithms will give better performance.[36] This algorithm can be used for classification as well as regression. Various fields found the application of XGBoost like ecology, web search, and won many competitions on Kaggle. We have used n_estimators=300, learning rate =0.1, random_state=1234, and max_depth=3 in this study.

Light GBM

This is a gradient-based algorithm that uses tree structure learning.[37]It grows the tree vertically compared to other algorithms that grow tree horizontally (level by level). It selects the leaf with maximum delta loss. This algorithm's advantage is that it can handle large data with less memory and produces faster results. We have used learning_ rate = 0.1, n_estimators = 1300, and boosting = Gradient Boosting Decision Tree parameters in this study.

Extra Tree Regression

This algorithm is an extended version of the random forest proposed by Geurts *et al.*[38] which is computationally better than random forest. In a random forest subset of the training, the set is used to train the decision tree, but in extra tree regression, each decision tree is trained on the complete training set. Another difference is that it selects the features along with its values to split the tree. These two differences give better performance and avoid over fitting. We have used n_estimators = 200, random_state = 1234.

Elastic Net Regression

This model was introduced to reduce the limitation of Lasso regression. This model is a mixed model of ridge and lasso regression.[39] The quadratic part is added to the L1 penalty. In this tuning, parameter α is present, which can be tuned between 0 and 1. If α is zero, it corresponds to ridge regression, and one corresponds to Lasso regression. We have used alpha = 1.0, l1_ratio = 0.5.

AdaBoost Algorithm

Also called adaptive boosting. It combines various weak learners on the weighted training dataset.[40] Initially, equal weight is given to all the data points. If there is a wrong prediction in the first learner, then the weight of that data point is increased. Iteratively multiple algorithms are added until it reaches the limit in terms of models or performance. This algorithm can be used for both classification and regression. We have used n_estimators = 200, base_estimator = DecisionTreeRegresson, and learning_rate = 0.1.

K-Nearest Neighbors

The algorithm KNN[41] can be used for both classification and regression. The distance from each test and training sample point is calculated, and the k nearest neighbor is selected. The majority of classes of these neighbors are taken into account for obtaining the test input class. It does not use training data points to make any generalization and is based on feature similarity. It stores the entire dataset that it uses for representation. We have used

Euclidean distance, and the number of neighbors taken is 3.

Long Short-Term Memory

This is a special type of recurrent neural network which can handle long term dependencies. A recurrent neural network suffers from the problem of vanishing gradient. Hochreiter & Schmidhuber[42] developed a model called LSTM. LSTM is composed of three gates: (i) input gate — controls how much information from current input will be passed to cell state (ii) output gate — decides which information need to be sent to hidden state (iii) forget gate — controls how much information from the previous cell, and current input will flow into cell state. There are two other components in LSTM called cell state (c_t), representing the internal memory and hidden state (h_t) containing information of previous states that are used for prediction. Parameters used in this study are: (Optimizer = 'adam'), activation = 'relu', number of epochs = 50, LSTM = 100.

Stacking of Algorithms

Stacking is an ensemble method in which various machine learning algorithms predict and act as input to the next level predictor.[43] We found the Extra tree, random forest, XGBoost, and light GBM algorithms performed better among all applied. In hourly data, it was found the best combination of extra trees, AdaBoost, and light GBM in the first layer, and the output of these algorithms is given as an input to the random forest algorithm. In daily data, it was found the best combination of random forest, XGBoost, and light GBM in the first layer, and the output of these algorithms is given as an input to the extra tree algorithm. We found that stacking improves performance.

Experimental Results

We have performed all the experiments on the Kaggle notebook using Python language. We have used various libraries like pandas, scikit learn, NumPy, stats, Keras, matplotlib, and seaborn. Data cleaning is done using panda's library. Classical models, and machine learning models are implemented sci-kit-learn library. Graphs are plotted using seaborn library. Electricity consumption data of households in a smart home from Jan 1st, 2016 to Jan 7th, 2016, is taken from Kaggle. The portion of dataset used for testing is 20% and remaining 80% for training.

Evaluation Measures

The following parameters are used for the evaluation of the models applied:

Mean Absolute Error

This is the average of the absolute difference between the predicted value and actual value. The calculation of MAE is given in Eq. 5.

$$MAE = \frac{1}{n}\sum_{k=1}^{n} abs(yk - ypred) \qquad \ldots (5)$$

Root Mean Square Error

This is computed as taking the square root of the average squared difference between the predicted value and actual value, as given in Eq. 6.

$$RMSE = \sqrt{\frac{1}{N}\left(\sum_{k=1}^{n}(yk - ypred)^2\right)} \qquad \ldots (6)$$

where, yk is the actual value, and $ypred$ is the predicted value.

R-Square

This is a statistical way of measuring how close the data is concerning the regression line. Its value ranges from 0 to 1. This defines the variation in the target variable that is explained by dependent variables. Its calculation is given in Eq. 7.

$$R^2 = 1 - \frac{SS_{regression}}{SS_{total}} \qquad \ldots (7)$$

where, $SS_{regression}$ is the square sum of regression errors, and SS_{total} is the square sum of total error.

Mean Absolute Percentage Error (MAPE)

It is the average of percentage errors in forecasting. Error is a difference between actual and forecasted value. It is easier to understand than another matrics because it is presented in the form of a percentage. The MAPE calculation is given in Eq. 8.

$$MAPE = \frac{100\%}{n}\sum_{t-1}^{n} Abs(\frac{(A_t - F_t)}{A_t} \qquad \ldots (8)$$

where, A_t is actual value, F_t is forecasted value, n number of times the summation iteration happens. After the data was cleaned and resampled at the hourly and day level. Three types: classical, machine learning, and deep learning — a total of fifteen models are applied to forecast the electricity consumption of a household and best four models are ensemble using stacking. The results of fifteen forecasting models and one ensemble method (Stacking) on the daily forecasting data are given in Table 2. Out of all the fifteen models, the extra tree performed better among all, giving root mean square error value 0.12499, mean absolute error value 0.08576, R-square value of 0.86480, and mean absolute percentage error 10.35057. The next to the extra tree is a random forest model, giving root mean square error value 0.14499, mean absolute error value 0.09645, R-square value 0.81808, and mean absolute percentage error 11.41637. The worst performing model is k-nearest neighbour giving root mean square error 0.40480, mean absolute error 0.29017, R-square –0.41791, and mean absolute percentage error 30.47510. Out of these fifteen models, the best four performing models are the extra tree, random forest, light gradient boosting, and AdaBoost, and then stacking ensemble was applied. The stacking ensemble combines various models (also called base models) via meta-model to improve the performance. As shown in Fig. 3, data is fed to level 0 models whose output is given as input to the level 1 model. At level 1, the extra tree algorithm is used. Stacking improved the results, giving root mean square error of 0.10540, mean absolute error 0.08475, R-square value 0.92343, and mean absolute percentage error 8.59440. We have tried various combinations for level 0 and level 1 and found that AdaBoost, random forest, and LGBM at level 0 and extra tree at level 1 are giving better performance.

The results of fifteen forecasting models and one ensemble method (Stacking) on the hourly forecasting data are present in Table 3. Out of all the fifteen models, the extra tree is performing better among all, giving root mean square error value 0.20776, mean absolute error value 0.10219, R-square value of 0.93646, and mean absolute percentage error 12.02758. The next to the extra tree is the light gradient boosting model, which gives a root mean square error value of 0.21530, mean absolute error value of 0.10832, R-square value of 0.92805, and mean absolute percentage error of 12.96347. Out of these fifteen models, the best four performing models are an extra tree, random forest, light gradient boosting, and XGBoost is selected, and the stacking ensemble is applied. The stacking ensemble combines various models (also called base models) via meta-model to improve the performance. As shown in Fig. 4, data is fed to level 0 models whose output was

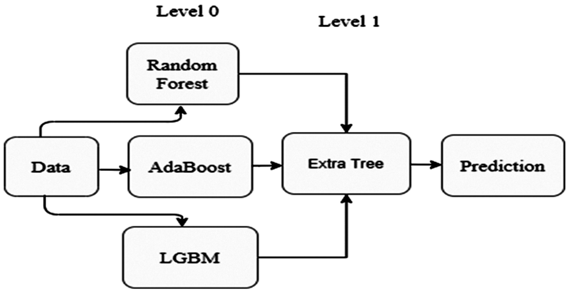

Fig. 3 — Stacking of models on daily data

Table 2 — Performance comparison of fifteen regression models on daily forecasting data

Models	RMSE	MAE	R-Square	MAPE	Time Taken (s)
Auto-ARIMA	0.14865	0.11719	0.43647	19.31362	50.84000
Exponential	0.22463	0.17438	−0.28681	20.90232	0.310700
Prophet	0.17977	0.13929	0.17577	21.91200	13.43820
Holt-Winter	0.15765	0.12382	0.36617	19.32949	0.32170
LR	0.28409	0.20258	0.30164	24.20845	**0.01200**
RF	0.14499	0.09645	0.81808	11.41637	1.56089
SVM	0.33990	0.25806	0.00030	29.56340	0.01724
DT	0.23350	0.15435	0.52822	18.73307	**0.01232**
XGBoost	0.16652	0.09317	0.76006	11.18346	1.09400
Extra Tree	0.12499	0.08576	0.86480	10.35057	0.61340
LGBM	0.18584	0.11547	0.70117	12.97763	0.90744
Elastic Net	0.34827	0.26562	−0.04953	30.30201	**0.00956**
KNN	0.40480	0.29017	−0.41791	30.47510	0.21690
AdaBoost	0.14940	0.09236	0.80685	11.52736	1.07660
LSTM	0.23311	0.17205	−0.38579	22.06227	10.63260
Stacking	0.10540	0.08475	0.92343	8.59440	2.58840

Table 3 — Performance comparison of fifteen regression models on hourly forecasting data

Models	RMSE	MAE	R-Square	MAPE	Time Taken (s)
Auto-ARIMA	0.53839	0.38481	−0.48214	60.07794	1161.429
Exponential	0.62865	0.54823	−1.02071	40.07622	0.149947
Prophet	0.50571	0.35445	−0.30768	49.47551	734.0000
Holt-Winter	0.79474	0.72184	−2.22953	45.58178	0.13450
LR	0.51791	0.30932	0.58364	54.52883	**0.01706**
RF	0.26209	0.13071	0.89338	14.62941	43.5613
SVM	0.72592	0.40098	0.18204	48.52642	5.71709
DT	0.42048	0.20384	0.72556	50.3057	0.24192
XGBoost	0.22530	0.11956	0.92121	17.33116	2.92039
Extra Tree	0.20766	0.10219	0.93646	12.02758	9.63167
LGBM	0.21530	0.10832	0.92805	12.96347	6.46062
Elastic Net	0.80279	0.48023	−0.00037	56.10739	**0.01386**
KNN	0.70743	0.39681	0.22317	49.55058	1.90454
AdaBoost	0.22477	0.10940	0.92158	13.48812	39.6465
LSTM	0.40112	0.19768	0.80356	0.359852	220.513
Stacking	0.18066	0.10196	0.94112	10.70761	24.9661

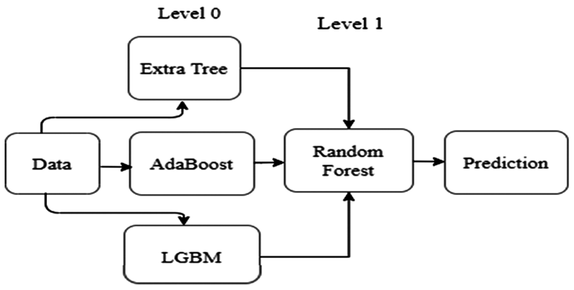

Fig. 4 — Stacking of models on hourly data

given as input to the level 1 model. At level 0, extra tree, AdaBoost, and light GBM models are used. At level 1 random forest algorithm is used. Stacking improved the results and giving root mean square error 0.18066, mean absolute error 0.10196, R-square value 0.94221, and mean absolute percentage error 10.70761.LSTM couldn't produce better predictions than Statistical models and machine learning models because LSTM need more data to tune their parameters. And, in our case, we have only one-week data. We have also computed the time taken by each algorithm on both the datasets. It is observed that elastic net, decision tree, and linear regression has taken less time as compared to other algorithms applied but their performance is poor on both the datasets (hourly level and day level). Stacking ensemble is taking 2.58840 seconds on daily level and 24.9661 seconds at hourly level.

The comparison of this work with the approaches reported in the literature is presented in Table 4. Our models show a 2% improvement from the state of art results in terms of RMSE and MAPE. The performance shown in Table 4 on existing works is achieved by researchers on their own datasets. It is observed that stacking can perform better as compared to existing work on electricity load forecasting.

Table 4 — Comparison with existing Approaches

Paper	RMSE	MAPE	Approach Used
Indikawati & Zamroni[9]	0.19904	—	XGBoost
Dey *et al.*[21]	0.6323	—	GMDH-MIA-linear
Gerossier *et al.*[26]	0.2890	0.38	Probabilistic Model
Andriopoulos, Nikos *et al.*[25]	0.93	—	CNN Model
Khan *et al.*[14]	0.47	0.76	CNN-LSTM-AE
Our Work	0.18066	0.1070	Stacking Ensemble

Discussion

Based on result analysis of 16 algorithms on energy forecasting of households following insights are as follows:

(i) The electricity consumption forecasting at residential level is required because 87% people in the world uses electricity. The availability of data at residential level electricity consumption is less and consumption depends upon occupant behavior and other factors also due to which it is quite challenging to perform forecasting at residential level.

(ii) In this study, classical, machine learning, deep learning, and ensemble learning methods have been applied to forecast household's electricity consumption at hourly and daily level. The stacking ensemble method produces better results on both hourly and daily electricity consumption dataset among all the algorithms applied. Stacking algorithm on the hourly dataset gives RMSE value 0.18066, MAE value 0.10196, R^2 value 0.94112, and MAPE value 10.70761. Stacking algorithm on the daily dataset gives RMSE value 0.10540, MAE value 0.08475, R^2 value 0.92343, and MAPE value 8.59440.

(iii) The results show that stacking ensemble learning algorithm and Extra tree algorithms can produce better results on the electricity forecasting problems. The stacking ensemble model performed 2% better than the state of the art results (given in Table 4).

(iv) One deep learning algorithm (LSTM), which does not produce good results on the dataset considered because of the small dataset size.

(v) We can apply other deep learning algorithms like CNN, Auto encoders, Attention Mechanism, Transformers, and their ensemble on the same problem in the future.

(vi) The algorithms are compared on the basis of time taken also, it is observed from the Tables 2 and 3 that elastic net and logistic regression model has taken less time but their performance is poor.

Despite the good results produced, there are some limitations also, which are as follows:

(i) This study aims to forecast electricity through machine learning, and the Internet of Things enabled smart home. Many developed countries uses smart meter to monitor the electricity consumption of households. But in developing countries still, smart meters are not being used. Due to this limitation, our approach cannot be used for the countries not using smart meter in the house.

(ii) The usage of different types of smart meters by customers can lead to inaccurate results of forecasting. This study will be applicable to customers using a particular type of meter.

(iii) In this study only one dataset is taken into consideration over the small duration.

Conclusions

In this study fifteen forecasting models and a stacking ensemble of the best algorithms (Light GBM, Ada Boost, Extra Tree, Random Forest, and XGBoost) on the electricity consumption data of a smart home are applied. The dataset was preprocessed by handling missing values, resampling, removing attributes with the same values. The granularity of data measurement is minute, and we have resampled the data at the day and hour levels. All the algorithms are compared on the basis of RMSE, MAE, MAPE, and R^2performance parameters. We have found that the stacking algorithm performed best among all the algorithms on a daily and hourly dataset. The extra tree best performance values on the daily dataset are MAE 0.085766, RMSE 0.12499, MAPE 10.35057, and R^2 0.864801, and stacking on this dataset gives MAE of 0.084746, RMSE 0.1054038, MAPE 8.59440, and R^2 0.92343. The extra tree best performance values on the hourly dataset are MAE 0.1021993, RMSE 0.2076643, MAPE 12.02758, and R^2 0.936459. The stacking on this dataset gives MAE 0.101956, RMSE 0.180658, MAPE 10.70761, and R^2 0.94112. It is concluded that an extra tree performs better among all the algorithms applied on both the datasets, and stacking further improves the performance. In the future, various meta-heuristic approaches like moth search algorithm, monarch butterfly optimization, elephant herding optimization, and earthworm optimization algorithm can be applied to extract features from the dataset. Further, on the selected features, forecasting algorithms can be applied.

References

1 Sorrell S, Reducing energy demand: A review of issues, challenges, and approaches, *Renew Sustain Energy Rev*, **47** (2015) 74–82.

2 Karanfil F & Li Y, Electricity consumption and economic growth: exploring panel-specific differences, *Energy Policy*, **82** (2015) 264–277.

3 Sieminski A, International energy outlook, Energy Inf Adm (EIA) 2014- available at (https://www.eia.gov/pressroom/presentations/sieminski_09222014_columbia.pdf)-accessed on 10 September 2020.

4 Nejat P, Jomehzadeh F, Taheri M M, Gohari M & Majid M Z A, A global review of energy consumption, CO_2 emissions, and policy in the residential sector, *Renew Sustain Energy Rev*, **43** (2015) 843–862.

5 Quilumba F L, Lee W J, Huang H, Wang D Y & Szabados R L, Using smart meter data to improve the accuracy of intraday load forecasting considering customer behavior similarities, *IEEE Trans Smart Grid*, **6(2)** (2015) 911–918.

6 Kwac J, Flora J & Rajagopal R, Household energy consumption segmentation using hourly data, *IEEE Trans Smart Grid*, **5(1)** (2014) 420–430.

7 Haben S, Singleton C & Grindrod P, Analysis and clustering of residential customers energy behavioral demand using smart meter data, *IEEE Trans Smart Grid*, **7** (2016) 136–144.

8 Bunn D & Farmer E D, *Comparative Models For Electrical Load Forecasting*, (Wiley, New York, NY, USA) 1985.

9 Indikawati F I & Zamroni G M, Household power consumption forecasting using IoT smart home data, *Journal Ilmiah Teknik Komputer*, **5(1)** (2019) 8–15.

10 Bansal A, Rompikuntla S K, Gopinadhan J, Kaur A & Kazi Z A, "Energy consumption forecasting for smart meters." arXiv preprint arXiv:1512.05979 (2015).

11 Rodrigues F, Cardeira C & Calado J M F, The daily and hourly energy consumption and load forecasting using artificial neural network method: a case study using a set of 93 households in Portugal, *Energy Procedia*, **62** (2014) 220–229.

12 Yildiz, Baran J I, Bilbao J, Dore A & Sproul, Household electricity load forecasting using historical smart meter data with clustering and classification techniques, *2018 IEEE Innov Smart Grid Technol Asia* (ISGT Asia) (2018) 873–879.

13 Shirke A & Chandane M M, IoT data based predictive modeling for energy usage of appliances in smart home, *Int Eng Technol*, **7(3)** (2018) 931–934.

14 Khan Z, Ahmad T, Hussain A, Ullah S & Rho, Towards efficient electricity forecasting in residential and commercial buildings: A novel hybrid CNN with a LSTM-AE based framework, *Sensors*, **20(5)** (2020) 1399–1399.

15 Rajabi R & Estebsari A, Deep learning based forecasting of individual residential loads using recurrence plots, in *Proc 2019 IEEE Milan Power Tech*, 2019, 1–5.

16 Wu D, Wang B, Precup D & Boulet B, Multiple kernel learning-based transfer regression for electric load forecasting, *IEEE Trans Smart Grid*, **11(2)** (2019) 1183–1192.

17 Kim T Y & Cho S B, Predicting residential energy consumption using CNN-LSTM neural networks, Energy, **182** (2019) 72–81.

18 Bogomolov A, Lepri B, Larcher R, Antonelli F, Pianesi F & Pentland A, Energy consumption prediction using people dynamics derived from cellular network data, *EPJ Data Sci*, **5** (2016) 13–13.

19 Chen Y, Xu P, Chu Y, Li W, Wu Y, Ni L, Bao Y & Wang K, Short-term electrical load fore- casting using the Support Vector Regression (SVR) model to calculate the demand response baseline for office buildings, *Appl Energy*, **195** (2017):659–670.

20 Fong S, Li J, Song W, Tian Y, Wong R K &Dey N, Predicting unusual energy consumption events from smart home sensor network by data stream mining with misclassified recall, *J Ambient Intell Humaniz Comput*, **9(4)** (2018) 1197–1221.

21 Dey N, Fong S, Song W & Cho K. Fore- casting energy consumption from smart home sensor network by deep learning, in *Int Conf Smart Trends Inf Technol Comput Commun*, 2017, 255–265.

22 Meher S, Estimating and forecasting residential electricity demand in Odisha, *J Public Aff*, **20(3)** (2020) 2065–2065.

23 Estebsari A & Rajabi R, Single residential load forecasting using deep learning and image encoding techniques, Electronics, **9** (2020) 68–68.

24 Mukherjee A, Mukherjee P, Dey N, De D & Panigrahi B K, Lightweight sustainable intelligent load forecasting platform for smart grid applications, Sustain Comput Inform Syst, **25** (2020) 100356–100356.

25 Andriopoulos N, Magklaras A, Birbas A, Pa- palexopoulos A, Valouxis C, Daskalaki S, Bir- bas M, Housos E & Papaioannou G P, Short term electric load forecasting based on data transformation and statistical machine learning, *Appl Sci*. **11(1)** (2021) 158–158.

26 Gerossier A, Girard R, Kariniotakis G & Mi-chiorri A, Probabilistic day-ahead forecasting of household electricity demand, *CIRED-Open Access Proc J*, **1** (2017) 2500–2504.

27 https://www.kaggle.com/taranvee/smart-home- dataset-with-weather-information, (Accessed on May 10th 2020)

28 Melard G & Pasteels J M, Automatic ARIMA modeling including interventions, using time series expert software, Int J Forecast **16(4)** (2000) 497–508.

29 Alejandro G M, "A comparison between LSTM and Facebook Prophet models: a financial forecasting case study." Available at http://openaccess.uoc.edu/ webapps/ o2/bitstream/10609/107367/8/agonzalez_mataTFG0120presentation.pdf, accessed on 15September 2020.

30 Ostertagova E & Ostertag O, Forecasting using simple exponential smoothing method, *Acta Electrotech Inform*, **12(3)** (2012) 62–66.

31 Hansun S, Charles V & Indrati C R, Revisiting the Holt-Winters' Additive Method for Better Forecasting, *Int J Enterp Inf Syst* (IJEIS), **15(2)** (2019) 43–57.

32 Montgomery D C, Peck E A & Vining G G, *Introduction to Linear Regression Analysis,* Vol 821, John Wiley & Sons, 2012.

33 Drucker H, Christopher J C, Burges L, Kaufman A J, Smola A & Vapnik V, Support vector regression machines, *NIPS'96: Proc 9th Int Conf Neural Info Proces Syst*, 1996, 155–161.

34 Breiman L, Stacked regressions, Machine learning, **24**(1) (1996) 49–64.

35 Ho T K, Random decision forests, in Proc Int Conf Doc Anal Recog (1995) 278–282.

36 Chen T, Guestrin C & Xgboost, *Proc 22nd ACM SIGKDD Int Conf Knowl Discov Data Mining-KDD '16*, 2016, 785–794.

37 Ke G, Meng Q, Finley T, Wang T, Chen W, Ma W, Ye Q & Liu T Y, Lightgbm: A highly efficient gradient boosting decision tree, *Adv Neural Inf Proc Syst*, (2017) 3146–3154.

38 Maleki H, Sorooshian A, Goudarzi G, Baboli Z, Birgani Y T & Rahmati M, Air pollution prediction by using an artificial neural network model, Clean Technol Envir, **21** (2019) 1341–1352.

39 Ogutu J O, Schulz-Streeck T & Piepho H P, Genomic selection using regularized linear regression models: ridge regression, lasso, elastic net and their extensions, BMC proc, **6** (2012) 10–10.

40 Solomatine D P & Shrestha D L, T: a boosting algorithm for regression problems, *2004 IEEE Int Joint Conf Neural Networks (IEEE Cat No 04CH37541)*, **2** (2004) 1163–1168.

41 Yu B, Song X, Guan F, Yang Z & Yao B, k-Nearest neighbor model for multiple-time-step prediction of short-term traffic condition, *J Transport Eng*, **142(6)** (2016) 4016018–4016018.

42 Hochreiter S & Schmidhuber J, Long-short-term memory, *Neural Comput*, **9(8)** (1997) 1735–1780.

43 Breiman L, Stacked regressions, Mach Learn, **24(1)** (1996) 9–64.