World Academy of Science, Engineering and Technology
International Journal of Energy and Power Engineering
Vol:16, No:8, 2022

Forecasting 24-Hour Ahead Electricity Load Using Time Series Models

Ramin Vafadary, Maryam Khanbaghi

Abstract—Forecasting electricity load is important for various purposes like planning, operation and control. Forecasts can save operating and maintenance costs, increase the reliability of power supply and delivery systems, and correct decisions for future development. This paper compares various time series methods to forecast 24 hours ahead of electricity load. The methods considered are the Holt-Winters smoothing, SARIMA Modeling, LSTM Network, Fbprophet and Tensorflow probability. The performance of each method is evaluated by using the forecasting accuracy criteria namely, the Mean Absolute Error and Root Mean Square Error. The National Renewable Energy Laboratory (NREL) residential energy consumption data are used to train the models. The results of this study show that SARIMA model is superior to the others for 24 hours ahead forecasts. Furthermore, a Bagging technique is used to make the predictions more robust. The obtained results show that by Bagging multiple time-series forecasts we can improve the robustness of the models for 24 hour ahead electricity load forecasting.

Keywords—Bagging, Fbprophet, Holt-Winters, LSTM, Load Forecast, SARIMA, tensorflow probability, time Series.

I. Introduction

FORECASTING is predicting future values based on past and current time series data. By forecasting electricity load we could optimize the planning, operation and control of power systems. Load forecasting can be either short-term or long-term forecasting. Short-term forecasting can be used for scheduling the generation and transmission of electricity [1] and control of power systems [2], whereas long-term load forecasting helps with the power supply development and delivery system [3]. A range of forecasting models have been used for load forecasting in the literature, from parametric models like ARIMA [4], linear regression [5] and Holt-Winters [6] to non-parametric models based on machine learning techniques like Artificial Neural Network (ANN) [7], [8], Support Vector Machines (SVM) [9], K-nearest neighbors (KNN) [10] and Gradient Boosting (GB) [11]. Studies have shown that, although in the long run, the load is strongly influenced by meteorological conditions and special events, an univariate model is sufficient in shorter lead times [12].

In [13] the authors have used four models including ARIMA, Holt-Winters, Nonlinear Auto-Regressive with Exogenous and Box-Cox transforms for load forecasting. They examined some characteristics of the load signals including seasonal patterns, weather effects, calendar effects and long-term trends and considered their effects on forecasting. They showed that yearly pattern and temperature information are only useful for high aggregation level load forecasting. They claimed that double seasonal Holt-Winters performed better compared to other models. In [14] the authors compared SARIMAX, random forest (RF) and gradient boosting regression trees (GBRT), finding that GBRT is superior to others. They used temperature as an exogenous predictor and their result showed that it did not seem to improve the predictions significantly.

In recent years deep learning models specifically, Recurrent Neural Networks (RNNs) are used in time series prediction [15]. Long Short Term Memory (LSTM) is a type of RNN that can learn the order dependence between items in a sequence. LSTMs are very efficient at remembering long term dependencies and are not vulnerable to the vanishing gradient problem which exists in regular RNNs [16]. LSTM is widely used in time series prediction [17]. Fbprophet is a time-series forecasting model published by Facebook company that enables us to test or perform forecasting in python at scale. It gives best results for time series which have several season(s) of historical data and strong seasonal effects [18].TensorFlow Probability (TFP) is a library for probabilistic reasoning and statistical analysis in TensorFlow [19]. Structural Time Series (STS) models in TFP are used for time series forecasting. STS fits the resulting time series models with Variational Inference and Hamiltonian Monte Carlo.

In this work we compare five different models, namely Seasonal ARIMA, Holt-Winters, LSTM, Fprophet and Tensorflow probability for 24 hour ahead load forecast for the NREL residential data. Also a bagging technique is used to optimize the train/test split for training the time series and forecasting. Bagging technique was introduced by Leo Breiman in 1994 [20]. Bagging is one of the most used techniques for combining several predictors in order to produce a highly accurate model.

The paper is organized as follows. In Section II, the time series used is being presented and explored. In Section III, we briefly explain about the five models being used for the electricity load forecasting. In Section IV, we evaluate the performance of all five models, analyze the results and explore a special case. In Section V we use a Bagging technique to improve the robustness of the models. Finally in Section VI we have the conclusion and future work.

II. Time Series Data exploring

Five different models mentioned in Section I are used to forecast the 24 hour ahead electricity load for the residential loads. Our dataset is composed of the hourly load of a

Ramin Vafadary and Maryam Khanbaghi are with the Department of Electrical and Computer Engineering at Santa Clara University, Santa Clara, CA 95053 USA (e-mail: rvafadary@scu.edu; mkhanbaghi@scu.edu).

residential site provided by NREL for a 1-year period. Fig. 1 depicts the plot of load for the whole year. It shows that the load will increase during hot months and then decrease during cold months. To take a deeper look at the data we plotted one week of two different seasons, cold season Fig. 2 and hot season Fig. 3 where various seasonalities are visible.

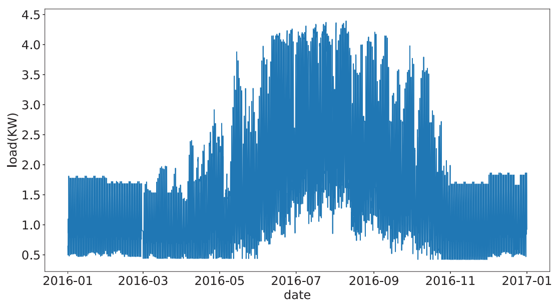

Fig. 1 Load data for the period of one year

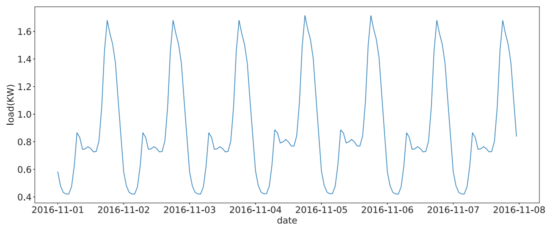

Fig. 2 Load data for cold season

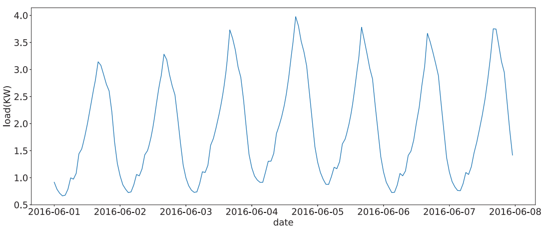

Fig. 3 Load data for hot season

Based on the data used, the hourly electricity load is highest in the hot season months when demand peaks in the afternoon as households are using air conditioning on hot days whereas during the cold season months, hourly electricity load peaks in the morning and the evening due to heater usage. By observing these two specific patterns we decided to forecast for two different seasons one for cold season and the other for hot season. We picked November for the cold season forecast and for the hot season we picked June.

Preprocessing the time-series would let us know whether the series is having linear or exponential trend, additive or multiplicative seasonality which aids us in using appropriate models for considering these effects. We decomposed our time-series data using the Auto Regressive decompose from Statsmodel library that provides three systematic components including level, trend, seasonality, and one non-systematic component called residual. They are shown in Figs. 4 and 5.

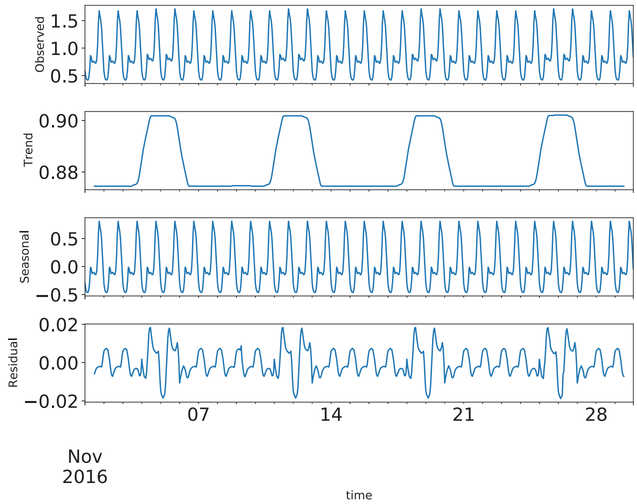

Fig. 4 Seasonal decomposition of the load Time-series for cold season

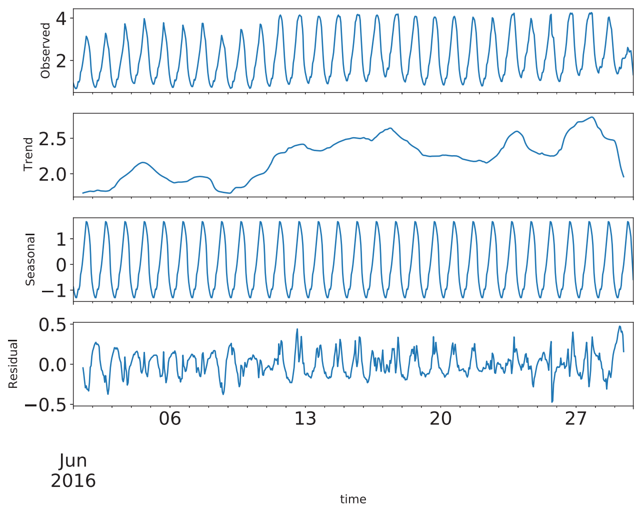

Fig. 5 Seasonal decomposition of the load Time-series for hot season

By looking at decomposition plots we see that there is a daily seasonality in the time series for both cold and hot seasons that their effects in the models need to be considered. In the next section we briefly explain the five models that are used for forecasting.

III. Time Series Analysis Models

A. Holt-Winters

Holt-Winters assigns exponentially decreasing weights and values against historical data to decrease the value of the weight for the older data. It is based on three smoothing equations and a forecast equation. The basic equations with additive seasonality are:

$$\begin{aligned} Level: L_t &= \alpha\frac{Y_t}{S_{t-s}} + (1-\alpha)(L_{t-1}+m_{t-1}) \\ Trend: m_t &= \beta(L_t - L_{t-1}) + (1-\beta)m_{t-1} \\ Seasonal: S_t &= \gamma\frac{Y_t}{L_t} + (1-\gamma)S_{t-s} \\ Forecast: F_{t+q} &= (L_t + m_t q) + (S_{t-s+q}) \end{aligned} \tag{1}$$

where Y_t is the current value, S is the length of seasonality that we set to 24 because of the daily seasonality , L_t is the level (average) of the series, m_t is the trend, S_t is the seasonal component, and F_{t+q} is the forecast for q periods ahead. To fit the model and estimate model parameters α, β, γ we used the inbuilt libraries of Statsmodels which uses the log-likelihood maximization.

B. SARIMA

Seasonal ARIMA model (SARIMA) is formed by adding seasonal terms in the ARIMA model. ARIMA uses a number of lagged observations of time series to forecast the future. SARIMA models are written as:

$$ARIMA(p,d,q)(P,D,Q)_m \tag{2}$$

where (p,d,q) and $(P,D,Q)_m$ are the non-seasonal and seasonal part of the model, respectively. p is auto regressive, d represents the degree of differencing and q is the moving average part. The parameter m is the number of periods per season. To fit a SARIMA model we took the steps below:

1) Check the stationarity using Augmented Dickey-Fuller test (ADF) [21].
2) Select a range for (p,d,q,P,D,Q) by looking at Auto Correlation Function (ACF) and Partial Auto Correlation Function (PACF) graphs shown in Figs. 6 and 7 for each season.
3) Use inbuilt libraries of Statsmodels called Auto ARIMA that automatically select the best parameters from the range defined by minimizing Akaike Information Criterion (AIC).

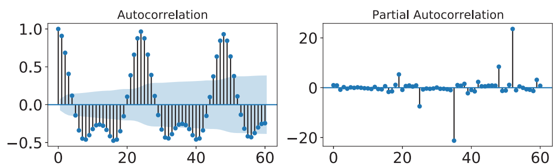

Fig. 6 Autocorrelation (ACF) and Partial Autocorrelation (PACF) for cold season

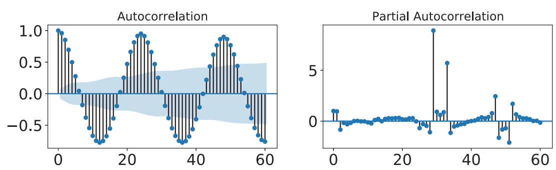

Fig. 7 Autocorrelation (ACF) and Partial Autocorrelation (PACF) for hot season

C. LSTM

The Long Short-Term Memory network, or LSTM network, is a recurrent neural network that is trained using Backpropagation through time. We used Bidirectional LSTM [22] for forecasting since Bidirectional LSTM allows the model to learn the input sequence both forward and backwards and concatenate both interpretations. To forecast the 24 hour ahead of the load, we defined the neural network structure and trained it with historical load data. The structure of the LSTM is shown in Fig. 8.

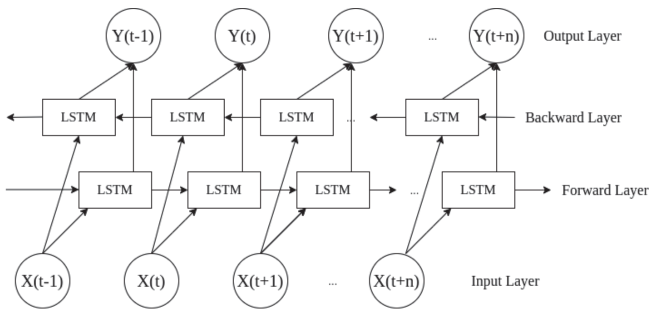

Fig. 8 LSTM Architecture

D. Fbprophet

Fbprophet is an open source forecasting framework developed by Facebook. It works best with time series that have strong seasonal effects. Fbprophet is based on regressive models that represent the time series with their components like seasonality, trend and residuals:

$$y(t) = g(t) + s(t) + h(t) + \varepsilon t \tag{3}$$

with $g(t)$ representing the trend, $s(t)$ representing the seasonality, $h(t)$ representing the effects of holidays, and εt representing any unusual change which is not accommodated by the model. Some highlights of Fbprophet are being very fast since it is built in Stan (programming language), robust to missing data and outliers and easy procedure to tweak and adjust forecasts while adding domain knowledge or business insights. To forecast 24 hours ahead load with Fbprophet we used the inbuilt python library of Fbprophet provided by Facebook.

E. TensorFlow Probability

An STS model express an observed time series as the sum of simpler components:

$$f(t) = \sum_{1}^{N} f_k(t) + \varepsilon \; where \; \varepsilon \sim N(0, \sigma^2) \tag{4}$$

Each assumed $f(t)$ has a particular structure, e.g. specific seasonality, trend, autoregressive terms, etc. the parameters of the model are fit with Variational Inference [23] and Hamiltonian Monte Carlo [24].

In the next section we are going to use these five models to forecast 24-ahead load and compare their performances.

IV. Results

A. Comparing Models' Performances

In this section we use the models explained above and make predictions for the next 24-hour ahead, based on the historical data that we have from NREL. As stated before we forecast for two seasons (hot and cold), We picked November for cold season and June for hot season. The whole month's data are split into train data and test data. 29 days of the month are being used to train the models and the 30th day of the month is used to test and evaluate the models. The train/test splits are shown in Fig. 9 for November (cold season) and Fig. 11, for June (hot season). The prediction results for all models are shown in Fig. 10 for November (cold season) and Fig. 12 for June (hot season).

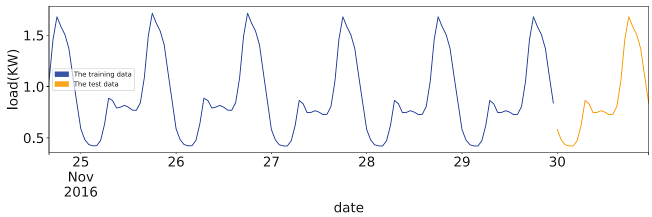

Fig. 9 Train-Test Split for November (cold season)

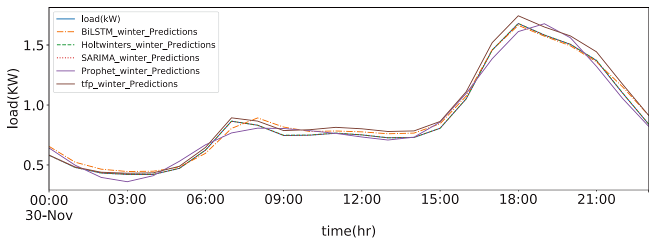

Fig. 10 Prediction results from all models for November (cold season)

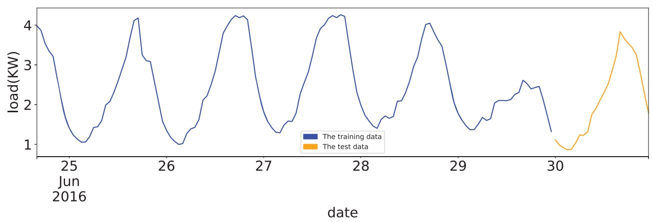

Fig. 11 Train-Test Split for June (hot season)

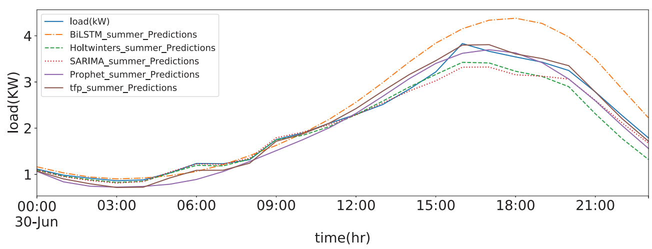

Fig. 12 Prediction results from all models for June (hot season)

B. Evaluation Metrics

The performance of these different models is evaluated by using the forecasting accuracy criteria namely, the Mean Absolute Error (MAE) and Root Mean Square Error (RMSE). The advantage of these two measures is that the average forecast error of a model is expressed in the same units of the variable to be predicted. The two measures can assume values greater than or equal to 0, and lower values are considered better. MAE expresses the absolute error, thus it is easy to understand. RMSE assigns high penalties to large errors, since the prediction errors are squared. It follows that the RMSE can be useful when we want to abstain from large forecasting errors.

$$
\begin{aligned}
MAE &= (\frac{1}{n})\sum_{i=1}^{n}|y_i - \hat{y}_i| \\
RMSE &= \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}
\end{aligned}
\tag{5}
$$

where $\hat{y}_i$ and y_i are the predicted and the real values respectively. The performance results are shown in Table I for November (cold season) and Table II for June (hot season).

TABLE I
Comparing Models Errors for Cold Season Predictions

Model	RMSE	MAE
$Holt-Winters$	1.097e-3	8.853e-4
$SARIMA$	3.534e-7	3.22e-7
$BiLSTM$	1.729e-2	1.345e-2
$Fprophet$	5.053e-2	4.355e-2
TFP	0.266	0.183

TABLE II
Comparing Models Errors for Hot Season Predictions

Model	RMSE	MAE
$Holt-Winters$	0.229	0.151
$SARIMA$	0.180	0.116
$BiLSTM$	0.266	0.183
$Fprophet$	0.266	0.183
TFP	0.266	0.183

Both SARIMA modeling and Holt-Winters smoothing produced better 24-ahead forecasts. Specifically, the errors obtained from SARIMA are significantly smaller than the values obtained from the other models and this is reasonable since BiLSTM and Tensorflow Probability (TFP) are deep learning models and they require a lot of data, they will produce better results for long-term forecasting. For short-term forecasting SARIMA and Holt-Winters are better while for long-term forecasting they will converge to a mean value. In the next section we will explore a special case where we have a transition from cold season to hot season.

C. Special Case

As stated before we randomly split our data to train and test sets. We used 29 days as the training set to predict the

24-hour ahead of the 30th day and used the 30th day forecast to evaluate the performance of our models. In this section we want to explore a case when we have the transition from cold season to hot season or vice versa. When we look at the October and November data shown in Fig. 13, we can see that the pattern of the load is changing gradually around October 25th. The peak load is reduced by about 20% from Oct to Nov. Now if we want to forecast a day ahead in November for example November 11th, what would be the best train/test split? Let's consider two scenarios, one where we use the 29 days before as the training set shown in Fig. 13 and forecast with the SARIMA model.

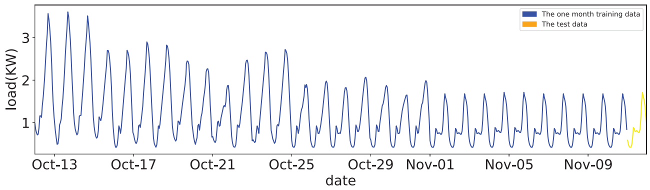

Fig. 13 Train/test split with one month train (Oct 10-Nov 10) for special case

The second scenario where we use one week before as the training set shown in Fig. 14 and forecast with SARIMA for one day ahead. We compare the performance of one week training set versus one month training set shown in Table III We can observe one week training set gives us better forecast with smaller errors.

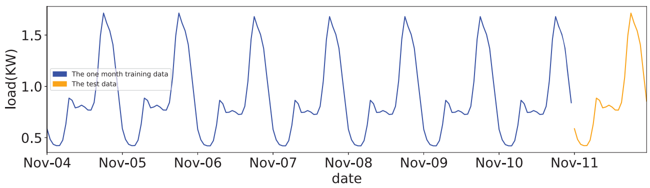

Fig. 14 Train/test split with one week train (Nov 4-Nov 10) for special case

TABLE III
COMPARING TWO TRAINING SETS ERRORS FOR TRANSITION CASE

Training set length	RMSE	MAE
One week training set	0.031	0.026
One month training set	0.053	0.044

These results are justifiable since the one month training set contains different peak load but one week training set only contains one peak load which is much closer to the next day forecast. In order to improve the robustness of these results we are going to use a bagging technique in the next section that makes our predictions more robust against any uncertainties and outliers in our data.

V. BAGGING TECHNIQUE

In this section we are going to use a Bagging technique to make the predictions more robust by training multiple time-series with different train/test splits and aggregate the

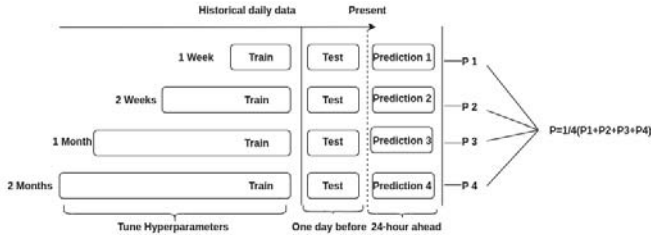

Fig. 15 Bagging for 24-hour ahead load forecasting

forecasts of all the models trained with different training sets. The Bagging technique is shown in Fig. 15.

We used one week, two weeks, one month and two months as training sets and predict the transition case, then we aggregate the results by averaging the predictions for all the models. The performance results for predictions using the Bagging technique are shown in Table IV for the transition case. We can observe that the errors of Bagging technique is less than the one with one month's training set. Based on this result we claim that Bagging technique can make our forecasts more robust.

TABLE IV
COMPARING ERRORS FOR TRANSITION CASE FOR DIFFERENT TRAINING SETS

Model	RMSE	MAE
One week training set	0.031	0.026
One month training set	0.053	0.044
Aggregating results	0.044	0.037

VI. CONCLUSION AND FUTURE WORK

This work presents a comparison of five different time series models that may be used to forecast a 24 hour ahead electricity load. Various classes of time series models, namely Holt-Winters, SARIMA, LSTM, Fbprophet and Tensorflow Probability have been considered. Results indicated that the SARIMA model performed better compared to the other models. We proposed a Bagging technique that could aggregate the results of multiple time-series trained with different train/test splits. This technique could improve the forecasts when we have uncertainties in the data. As future work, we suggest the inclusion of additional exogenous variables in our models, such as temperature and humidity in order to improve the forecast accuracy. Finally, in our next study we will explore additional models like XGBOOST and Random Forest.

REFERENCES

[1] K. Chen, K. Chen, Q. Wang, Z. He, J. Hu and J. He, "Short-Term Load Forecasting With Deep Residual Networks," IEEE Transactions on Smart Grid, vol. 10, no. 4, pp. 3943-3952, July 2019, doi: 10.1109/TSG.2018.2844307.
[2] Cortes A and Garg A, "Impact of forecast on control methods for customer sited battery storage", CIRED 2019, Madrid, Spain
[3] A. A. Mamun, M. Sohel, N. Mohammad, M. S. Haque Sunny, D. R. Dipta and E. Hossain, "A Comprehensive Review of the Load Forecasting Techniques Using Single and Hybrid Predictive Models," IEEE Access, vol. 8, pp. 134911-134939, 2020, doi: 10.1109/ACCESS.2020.3010702.

[4] Norizan, M., Ahmad, H.M. , Suhartono and Ismail, Z. (2011).“ Improving Short Term Load Forecasting Using Double Seasonal Arima Model”, World Applied Sciences Journal, Vol. 15, pp. 223-231.
[5] J. Zhao, L. Li, Z. Xu, X. Wang, H. Wang and X. Shao, “Full-Scale Distribution System Topology Identification Using Markov Random Field,” IEEE Transactions on Smart Grid, vol. 11, no. 6, pp. 4714-4726, Nov. 2020, doi: 10.1109/TSG.2020.2995164.
[6] J. W. Taylor and P. E. McSharry, “Short-Term Load Forecasting Methods: An Evaluation Based on European Data,” IEEE Transactions on Power Systems, vol. 22, no. 4, pp. 2213-2219, Nov. 2007, doi: 10.1109/TPWRS.2007.907583.
[7] J. Wang, X. Chen, F. Zhang, F. Chen and Y. Xin, “Building Load Forecasting Using Deep Neural Network with Efficient Feature Fusion,” Journal of Modern Power Systems and Clean Energy, vol. 9, no. 1, pp. 160-169, January 2021, doi: 10.35833/MPCE.2020.000321.
[8] Huaizhi Wang, Yangyang Liu, Bin Zhou, Canbing Li, Guangzhong Cao, Nikolai Voropai, Evgeny Barakhtenko, “Taxonomy research of artificial intelligence for deterministic solar power forecasting,” Energy Conversion and Management, Volume 214, 2020, 112909.
[9] Y. Fu, Z. Li, H. Zhang, P. Xu, “Using support vector machine to pre-dict next day electricity load of public buildings with sub-meteringdevices,” The 9th International Symposium on Heating, Ventilationand Air Conditioning (ISHVAC) Joint with the 3rd International Conference on Building Energy and Environment (COBEE), 12-15 2015, Tianjin,China, Procedia Eng. 121 (2015) 1016–1022.
[10] O. Valgaev, F. Kupzog and H. Schmeck, “Building power demand forecasting using K-nearest neighbours model – practical application in Smart City Demo Aspern project,” CIRED - Open Access Proceedings Journal, vol. 2017, no. 1, pp. 1601-1604, 10 2017, doi: 10.1049/oap-cired.2017.0419.
[11] Zheng, H.; Yuan, J.; Chen, L. Short-Term Load Forecasting Using EMD-LSTM Neural Networks with a Xgboost Algorithm for Feature Importance Evaluation. Energies 2017, 10, 1168.
[12] M. Mohsen, T. Faraz, S. Esmaeil, G, Noradin, and A, Oveis, “Small-Scale building load forecast based on Hybrid forecast engine,” Neural Processing Letters, 48(1), pp. 329-351, 2018.
[13] T. Dang-Ha, F. M. Bianchi and R. Olsson, “Local short term electricity load forecasting: Automatic approaches,” 2017 International Joint Conference on Neural Networks (IJCNN), Anchorage, AK, 2017, pp. 4267-4274, doi: 10.1109/IJCNN.2017.7966396.
[14] S. Papadopoulos and I. Karakatsanis, “Short-term electricity load forecasting using time series and ensemble learning methods,” 2015 IEEE Power and Energy Conference at Illinois (PECI), Champaign, IL, 2015, pp. 1-6, doi: 10.1109/PECI.2015.7064913.
[15] A. Tokgöz and G. Ünal, “A RNN based time series approach for forecasting turkish electricity load,” 2018 26th Signal Processing and Communications Applications Conference (SIU), Izmir, 2018, pp. 1-4, doi: 10.1109/SIU.2018.8404313.
[16] S. Hochreiter and J. Schmidhuber. “Long Short-Term Memory.” Neural Computation, 9(8), 1997.
[17] Y. Hua, Z. Zhao, R. Li, X. Chen, Z. Liu and H. Zhang, “Deep Learning with Long Short-Term Memory for Time Series Prediction,” IEEE Communications Magazine, vol. 57, no. 6, pp. 114-119, June 2019, doi: 10.1109/MCOM.2019.1800155.
[18] Sean J Taylor and Benjamin Letham. “Forecasting at scale.” The American Statistician, 72(1):37– 45, 2018.
[19] Software available from: http://tensorflow.org
[20] L. Breiman, “Bagging predictors,” Department of Statistics. University of California, Berkley, California 94720, USA, Tech. Rep. 421, September 1994.
[21] D. Dickey and F. Wayne, “Distribution of the Estimators for Autoregressive Time Series With a Unit Root,” Journal of the American Statistical Association 74, no. 366 (1979): 427–31,DOI: 10.1080/01621459.1979.10482531.
[22] M. Schuster and K. K. Paliwal, ”Bidirectional recurrent neural networks,” IEEE Transactions on Signal Processing, vol. 45, no. 11, pp. 2673-2681, Nov. 1997, doi: 10.1109/78.650093.
[23] D. Blei, A. Kucukelbir and J. McAuliffe “Variational Inference: A Review for Statisticians,” Journal of the American Statistical Association, 112:518, 859-877, 2017, DOI: 10.1080/01621459.2017.1285773
[24] R. Neal, “MCMC using Hamiltonian dynamics,” Handbook of Markov Chain Monte Carlo, 2012.