Cascaded Ensemble-based Short-term Load Forecasting for Smart Energy Management

Joshua Ottens
Electr. and Comp. Eng.
Lakehead University
Thunder Bay, Canada
jaottens@lakeheadu.ca

Thangarajah Akilan
Software Engineering
Lakehead University
Thunder Bay, Canada
takilan@lakeheadu.ca

Amir Ameli
Electr. and Comp. Eng.
Lakehead University
Barrie, Canada
aameli@lakeheadu.ca

Mohammad N. Uddin
Electr. and Comp. Eng.
Lakehead University
Barrie, Canada
muddin@lakeheadu.ca

***Abstract*—The rising adoption of renewable energy generation coupled with the anticipated increase in demand for reliable electricity over the coming years has brought attention to the importance of accurate short-term load forecasting. Short-term load forecasting plays an essential role in the scheduling and planning of the power grid's resources to ensure it operates efficiently and reliably. This paper proposes a short-term load prediction model that exploits `XGboost` and `LightGBM` models under a cascaded ensemble architecture to provide highly accurate predictions. The architecture minimizes the weaknesses of individual predictors by combining advanced feature engineering and feature selection strategies. The proposed model's effectiveness is tested on the publicly available benchmark dataset using mean average percent error (MAPE) to evaluate the models accuracy, while runtime is used to evaluate the model's computational efficiency. The proposed model demonstrates increased performance when compared to both the baseline model and conventional models.**

***Index Terms*—ensemble model, feature engineering, load forecasting, machine learning**

I. Introduction

In 2018, power outages cost the United States economy over $150 billion according to Bloom Energy [1]. These outages were caused by several factors including inadequate power generation due to mismatched load prediction and consumption as well as damage to the grid due to climate-related disasters such as the California wildfires. The recent increase in focus on climate change has led to the phasing out of traditional 'dirty' power generation sources, like coal and oil, and an increase in the adoption of renewable energy sources. While such energy resources reduce the contributions of the electricity generation sector to climate change, it comes with one major drawback. Most renewable generation methods are reliant on unpredictable energy sources—e.g., the wind or the sun. Historically, ensuring reliable power supply involved future load estimation using time series analysis methods such as moving averages or Kalman Filters. These techniques facilitated ramping up energy production during peak hours and scaling down production during off-peak hours. The power system operators could maintain sufficient spinning reserves and fast response generation to deliver energy to the grid on short notice if the customers' energy demands exceeded the predicted demand. While this approach provided a reliable energy supply, it came at the cost of inefficient generation practices.

In order to meet the proposed net-zero emission goals, the demand for electricity in Canada is forecast to grow between 160% and 210% by 2050 [2]. As demand for renewable electricity rises and carbon-emitting generation methods are phased out, the traditional inefficient approach for reliable power supply becomes increasingly precarious. To solve this problem, a highly accurate load prediction model must be developed so that system efficiency can be increased, while still providing reliable power to the customers. Leveraging the capabilities of machine learning (ML) to rapidly analyze complex, non-linear data and predict the most likely outcome has made the use of ML algorithms for load forecasting a prominent research area. This ability to capture patterns in energy demand will lead to proactive optimization of control decisions and allow the power grid to achieve higher efficiency, longer lifetime, and reduced operation costs.

For large-scale implementation of renewable generation methods to be practical, (i) precise weather forecasting is essential to provide grid operators with an accurate estimation of the generated power, and (ii) grid operators need access to highly accurate load prediction algorithms to schedule the required number of generators, ensuring reliable fulfillment of consumer power demands. In general, load forecasting is categorized into three main types: short-term forecasting (STLF) [3], which predicts the load one hour to one day in advance; medium-term forecasting (MTLF) [4], which predicts load one day to several months in advance; and long-term load forecasting (LTLF) [5], which predicts the load one year or more in advance [6]. The STLF prediction presents more challenges compared to MTLF and LTLF due to the higher variability in energy consumption patterns over a small stretch in time [7].

To address the aforementioned challenge, this work focuses on the intricate task of STLF and proposes an ensemble ML model for highly accurate predictions. The key contribution of this work is threefold:

- Introducing a unique Cascaded Ensemble (CE) model that integrates the benefits of two gradient-boosting algorithms to accurately predict future energy loads.

979-8-3503-3120-2/24/$31.00 © 2024 Crown

- Applying sophisticated feature engineering and selection strategies to identify the optimal inputs for enhanced prediction.
- Conducting exhaustive ablation study to benchmark the model and to demonstrate its effectiveness.

The rest of this paper is organized as follows: Section II reviews related works; Section III outlines the proposed methodology; Section IV analyzes the effectiveness of the proposed model; and finally, Section V concludes this paper with the findings of the work and provides directions for future research.

II. Literature Review

The related works are categorized into three groups: feature engineering, gradient boosting models, and ensemble models.

A. Feature Engineering

Prior to model construction, performing feature engineering is crucial to providing high-quality data to the model, resulting in more accurate predictions. For example, Singh *et al.* [8], use a dataset containing features, including dry bulb temperature, dew point temperature, hour of the day, day of the week, working day or holiday classification, loads from the past few hours, loads from the previous day and loads from the previous week. Similarly, a dataset containing paired temperature and load values for several lags for similar hours from the previous days, weeks, and months is exploited by Chen *et al.* [9]. It also includes a one-hot code for the season and a classification of weekdays or weekends. The research by Kondaiah *et* Saravanan [10], attempts to avoid overfitting by only using previous loads while all other features are eliminated. However, this approach is suboptimal, as factors like temperature and weather events can influence consumer load consumption. Therefore, considering them will enhance the model's prediction accuracy. After analyzing the literature with a focus on the features employed and their models' performance, it is observed that the models that use several lag features and some weather data achieve higher accuracy results compared to models that omit some of these features. It should also be highlighted that some models use an excessive number of features and not only attain a lower accuracy due to overfitting but also see an increase in computation time. On the other hand, the models introduced by Yao *et al.* [19] and Irankhah *et al.* [12] use a maximal information coefficient (MIC) to create a heatmap and visually show the relationship between the features and the predicted variable. By eliminating the features with low relationships, the models' performance is seen to improve. In Ur-Rehman *et al.* [13], a recursive feature elimination (RFE) method is used to reduce the feature set. A combination of the heatmap using MIC and the RFE methods is implemented to provide the optimal feature set for the proposed model. To further improve the model's accuracy, the algorithms in Chen *et al.* [9] and Kondaiah *et* Saravanan [16] use data normalization. This allows the data for each feature to be scaled between 0 and 1 to allow the model to operate more efficiently while also improving the overall performance. By researching various feature engineering and feature selection techniques, the most effective methods were determined and implemented to provide information-rich data to the models.

B. Gradient Boosting Models

Gradient boosting models combine multiple weak learners to create a strong overall model. For instance, the extreme gradient boosting (`XGboost`) model is a scalable tree-boosting system first proposed by Chen *et* Guestrin [17]. It is renowned for its rapid computation speed and precise predictions on complex non-linear datasets. This model can handle many features and works very well for predicting continuous time-series values, as is required in the case of this model. The `XGboost` model works by continually adding and training new trees to fit residual errors from the last iteration. The model then separately predicts each tree and then finally combines the prediction results of each tree to determine the final prediction value. Employing decision trees as base learners, `XGBoost` constructs multiple weak learners and applies continuous training through gradient descent. A predicted value is assigned to each instance by adding all the corresponding scores of the leaves together as follows

$$\hat{y}_i = \sum_{k=1}^{K} f_k(x_i) \quad | \quad f_k = w \times q(x), \tag{1}$$

where $\hat{y}_i$ is the predicted value, k is a tree of the decision tree, f_k is an independent function in the function space, $q(x)$ indicates that the sample x is assigned to a leaf node, and w is the leaf node weight. A `XGboost` model is built by Yao *et al.* [19] to provide load prediction that achieves a MAPE of 2.80% and a training time of 14.75 seconds. This offers predictions with an above-average accuracy when compared to other uni-modality load prediction approaches. It should be noted that the runtime for this model is slower compared to other gradient-boosting models observed in the literature, but still operates faster than other common load prediction models implemented using artificial neural networks (ANN), long short-term memory (LSTM) modules, or convolutional neural networks (CNN). The light gradient-boosting machine (`LightGBM`), developed by Ur-Rehman *et al.* [13], is an enhanced framework based on the decision tree algorithm. By providing an efficient design prioritizing speed and memory usage, LightGBM models are recognized for effectively handling large datasets while combating overfitting. These qualities are particularly important for accurate predictions. The `LightGBM` model takes the decision tree as the base learner as defined below

$$H_T(x) = \sum_{t=1}^{T} H_t(x) \quad | \quad H_t \in I \tag{2}$$

where H_t is the t-th learner and I is the collection space of all the learners. This model continuously improves the performance of this learner by completing multiple iterations and using the learner to obtain the mapping function from the input space to the gradient space. The `LightGBM` model built in Yao *et al.* [19] provides load prediction with a MAPE of 2.41% and a training time of 1.98 seconds. This offers results that have an improved accuracy over many other load prediction models. The LightGBM model is known to be very computationally efficient as it can achieve a runtime much lower than the competitors. The combination of high accuracy and computational efficiency makes it an attractive solution.

C. *Ensemble Models*

An ensemble model operates by combining the operation of several models together to provide an accurate, yet computationally efficient load prediction. They are used for many different applications as they provide a way to combine the strengths of several different models together to reduce prediction error. This technique is commonly used to reduce the effect that noise and individual model biases produce as it does not rely on the prediction of only one model as seen in Alhamid [21]. In Chen *et al.* [9], Lian *et al.* [22], and Hu *et al.* [23], an ensemble approach is implemented to combine the operation of two different models into a single model in an effort to improve the prediction accuracy. Research by Yao *et al.* [19] proposes an ensemble model consisting of an XGboost and LightGBM model and is seen to achieve a MAPE of 1.50% with a computation time of 16.63 seconds. The ensemble model in Wang *et al.* [24] also uses XGboost and LightGBM models to achieve a MAPE of 1.29%, but does not provide a reference runtime. Both these models achieve a higher accuracy than almost all of the other load prediction models investigated. The prediction time of the model by Wang *et al.* is quite fast compared to other ensemble models and popular models, such as the ANN-based solutions, but still provides room for improvement. This work considers the model in Lao *et al.* [19] as the baseline to evaluate the proposed model's performance with respect to prediction accuracy and computational efficiency.

In regards to these existing algorithms, the proposed solution focuses on optimizing three main areas: feature engineering, feature selection, and model construction. To enhance the feature engineering, the model proposed in this paper incorporates additional features absent in other works including the baseline model. These features are temperature and humidity data which provide critical environmental cues, as well as an exponentially weighted moving average (EWMA) that captures non-volatile seasonality information from the data. Another difference from the existing works is that the proposed model employs a dual criterion based on maximal information coefficient (5) and RFE to gain deeper insight into which features should be eliminated. Finally, the proposed framework utilizes a CE architecture, while the existing models including the baseline use a parallel ensemble model.

III. Methodology

Fig. 1 illustrates the proposed STLF pipeline. It subsumes three main stages—i.e., data preprocessing, model development and training, and model evaluation. The steps taken in the pipeline are further discussed below.

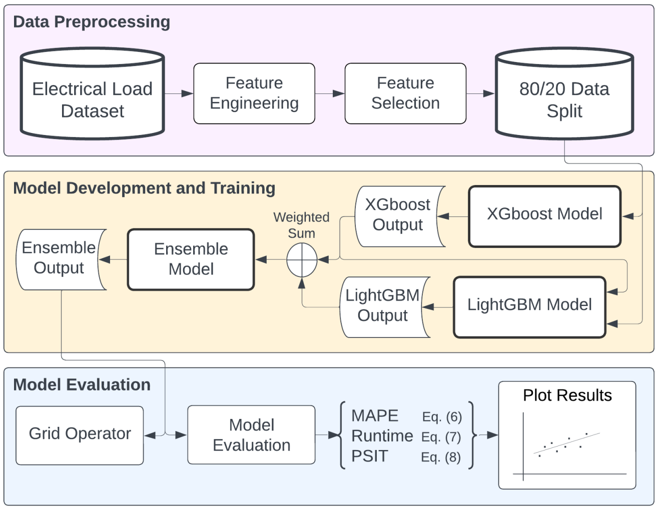


Fig. 1. The operational flow diagram of the proposed STLF.

A. *Datasets*

The primary dataset used for model development and validation contains four years of Spanish electricity grid consumption data from January 2015 to December 2018 provided by the Spanish TSO Red Electria [14]. Another dataset from the LETO Madrid weather station [15] containing hourly weather information is integrated with the primary dataset. The integrated data set contains time-series data observed hourly over the aforementioned duration, totaling 35,064 data points with ten different attributes containing each of the features. From these data samples, mutually exclusive training and test sets were created by splitting them with a ratio of 80:20 as shown in Fig. 1.

B. *Preprocessing*

Data preprocessing is a crucial step to ensure noise and outliers are mitigated and clean data is presented to the model. In this case, it includes data integration, feature engineering, and feature selection. In data integration, a database of load consumption is concatenated with a weather database from the same geographical location and for the same timestamps as the load consumption data points. Feature engineering is applied to create sequential data representing historical information (from the past hours, days, and weeks). The auto-correlation function (ACF) is used to determine a total of four consecutive lags capture valuable details, like the trends and seasonality of the load. Furthermore, all attributes except the target are normalized using the MinMaxScalar as follows

$$X' = \frac{X - X_{\min}}{X_{\max} - X_{\min}}, \tag{3}$$

where X' is the normalized value of X, and $X_{\min}$ and $X_{\max}$ represent the minimum and maximum values of X respectively. The EWMA defined in (4) is also applied to provide feature conditioning while eliminating short-term volatility [25].

$$X_{EWMA} = \frac{\sum_{i=0}^{t}(1-\alpha)^i X_{t-i}}{\sum_{i=0}^{t}(1-\alpha)^i}, \tag{4}$$

where i is the number of time periods, t is the timestamp of the input sample X, $\alpha = 2/(s+1)$, and s is a defined look-back period. In this case, it is set to twenty-four, chosen intuitively to capture the daily 24-hour seasonality of load and verified through empirical analysis of a range of look-back periods to find the optimal value. To avoid overfitting due to the large input feature set, MIC is used to determine the strength of the relationship between the dependent variable and the independent variable and then eliminate insignificant input features. The MIC between two variables X and Y is defined as

$$MIC(X,Y) = \max_{x,y}\left(\frac{I(X,Y)}{\log(\min(x,y))}\right), \tag{5}$$

where $I(X,Y)$ is the mutual information between X and Y, and x and y are the number of bins used for partitioning X and Y respectively in the construction of the grid for calculating mutual information. The computed MIC for the attributes of the data samples used in this work is visualized in a heatmap as seen in Table I. Besides this, RFE is also implemented to identify the most significant attributes. Based on the MIC and RFE, the least important features are dropped, greatly reducing the input feature dimension from 29 down to 15.

C. Proposed STLF Model

The proposed STLF model is a CE model that exploits `XGboost` and `LightGBM` as illustrated in Fig. 1. At the cascading stage, the intermediate output from the `XGboost` is fed to the `LightGBM` as an internal feature along with the raw input feature set. It improves the learning capability of the `LightGBM` predictor. At the ensemble stage, the target predictions of both models are refined using a weighted average aggregation operation. The hyperparameter settings of these models are summarized in table II. Once the hyperparameters are fine-tuned through the training process using the mutually exclusive training set, the weight for `XGboost` predictor and for the `LightGBM` predictor converged to 0.4 and 0.6 respectively. Fig. 2 demonstrates the model's learning progress wrt to MAPE and the training epoch. The model converges to a generalized solution after the 80th epoch, where both the testing and training errors plateau close to zero demonstrating that the model is neither overfitted nor underfitted. It is worth mentioning that feature selection plays a pivotal role in minimizing the number of features, resulting in a controlled model's complexity that mitigates the overfitting issue. The training procedure is also configured with an early stopping mechanism to help prevent overfitting.

D. Model Evaluation

The proposed model is evaluated on the mutually exclusive test set using evaluation metrics, like MAPE, runtime and per-sample inference time (PSIT). The MAPE metric is defined as follows

$$MAPE = \frac{1}{n}\sum_{t=1}^{n}\left|\frac{A_t - F_t}{A_t}\right|, \tag{6}$$

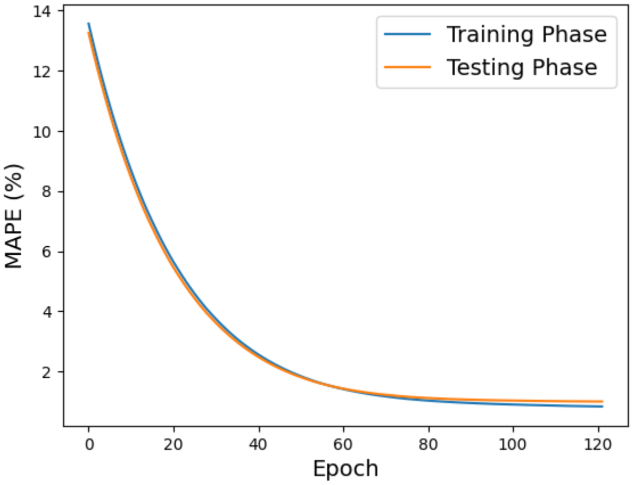


Fig. 2. Training and testing progress of the proposed ensemble-based STLF model.

where A_{t} is the actual value, F_{t} is the predicted value and n is the number of samples present. The runtime is calculated as follows

$$Runtime = t_{stop} - t_{start}, \tag{7}$$

where t_{stop} is the time the model stops running and t_{start} is the time when the model starts running. Finally, the PSIT is calculated as follows

$$PSIT = \frac{t_i}{n_s}, \tag{8}$$

where t_{i} is the total inference time and n_{s} is the total number of samples.

IV. Environment and Experimental Analysis

The performance of the proposed model in the defined environment is analyzed through both qualitative and quantitative methods in the following sections.

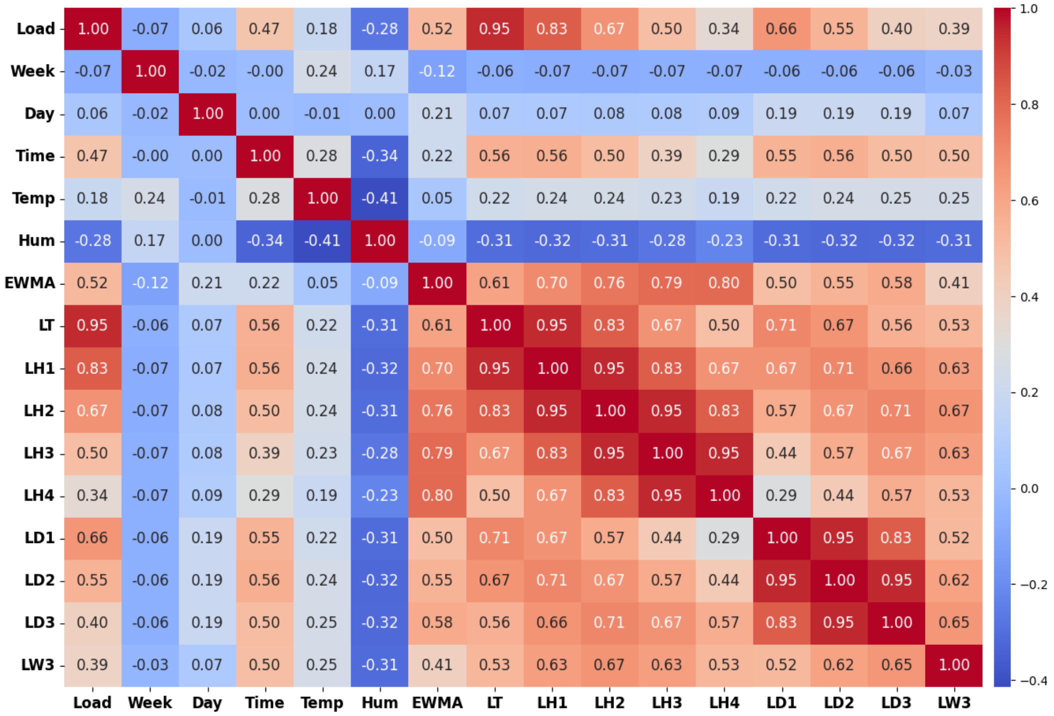
A. Environment

The proposed STLF model is developed on Google Colab notebook using Python version 3.10.12 and open-source libraries such as TensorFlow, Keras, and Scikit-learn. The model training and testing utilize a Tesla T4 GPU with 16 GB of GDDR6 memory and 2,560 CUDA cores.

B. Qualitative Analysis

Fig. 3 graphically analyzes the correctness of the proposed model in comparison to the ground truth consumption data. From the plots for daily and weekly predictions, it is observed that the proposed model captures the daily seasonality of the load very well. However, the model experiences inaccuracies at times when the load pattern has a sharp rate of change, such as near the typical 4 p.m. local minima and the 7 p.m. peak load. A similar issue can be observed from the weekly graph as well, more specifically at peak and local trough values on Tuesday, Wednesday, and Thursday as these days have sharper peaks and local troughs than the other days. Another contributing factor is that between 4 p.m. and 8 p.m. the load has high variability as many people are coming home from work and

TABLE I
Feature Correlation Heatmap based on MIC



	Load	Week	Day	Time	Temp	Hum	EWMA	LT	LH1	LH2	LH3	LH4	LD1	LD2	LD3	LW3
Load	1.00	-0.07	0.06	0.47	0.18	-0.28	0.52	0.95	0.83	0.67	0.50	0.34	0.66	0.55	0.40	0.39
Week	-0.07	1.00	-0.02	-0.00	0.24	0.17	-0.12	-0.06	-0.07	-0.07	-0.07	-0.07	-0.06	-0.06	-0.06	-0.03
Day	0.06	-0.02	1.00	0.00	-0.01	0.00	0.21	0.07	0.07	0.08	0.08	0.09	0.19	0.19	0.19	0.07
Time	0.47	-0.00	0.00	1.00	0.28	-0.34	0.22	0.56	0.56	0.50	0.39	0.29	0.55	0.56	0.50	0.50
Temp	0.18	0.24	-0.01	0.28	1.00	-0.41	0.05	0.22	0.24	0.24	0.23	0.19	0.22	0.24	0.25	0.25
Hum	-0.28	0.17	0.00	-0.34	-0.41	1.00	-0.09	-0.31	-0.32	-0.31	-0.28	-0.23	-0.31	-0.32	-0.32	-0.31
EWMA	0.52	-0.12	0.21	0.22	0.05	-0.09	1.00	0.61	0.70	0.76	0.79	0.80	0.50	0.55	0.58	0.41
LT	0.95	-0.06	0.07	0.56	0.22	-0.31	0.61	1.00	0.95	0.83	0.67	0.50	0.71	0.67	0.56	0.53
LH1	0.83	-0.07	0.07	0.56	0.24	-0.32	0.70	0.95	1.00	0.95	0.83	0.67	0.67	0.71	0.66	0.63
LH2	0.67	-0.07	0.08	0.50	0.24	-0.31	0.76	0.83	0.95	1.00	0.95	0.83	0.57	0.67	0.71	0.67
LH3	0.50	-0.07	0.08	0.39	0.23	-0.28	0.79	0.67	0.83	0.95	1.00	0.95	0.44	0.57	0.67	0.63
LH4	0.34	-0.07	0.09	0.29	0.19	-0.23	0.80	0.50	0.67	0.83	0.95	1.00	0.29	0.44	0.57	0.53
LD1	0.66	-0.06	0.19	0.55	0.22	-0.31	0.50	0.71	0.67	0.57	0.44	0.29	1.00	0.95	0.83	0.52
LD2	0.55	-0.06	0.19	0.56	0.24	-0.32	0.55	0.67	0.71	0.67	0.57	0.44	0.95	1.00	0.95	0.62
LD3	0.40	-0.06	0.19	0.50	0.25	-0.32	0.58	0.56	0.66	0.71	0.67	0.57	0.83	0.95	1.00	0.65
LW3	0.39	-0.03	0.07	0.50	0.25	-0.31	0.41	0.53	0.63	0.67	0.63	0.53	0.52	0.62	0.65	1.00

TABLE II
Hyperparameter Setting of the Models Used

Hyperparameter	Value
XGboost	
max_depth	9
learning_rate	0.046
n_estimators	122
gamma	2.372
reg_alpha	0.966
reg_lambda	4.953
min_child_weight	6
subsample	0.73
colsample_bytree	0.98
LightGBM	
max_depth	7
learning_rate	0.030
n_estimators	230
n_leaves	41
lambda_l1	0.597
lambda_l2	0.446
min_child_samples	43
subsample	0.717
feature_fraction	0.934
bagging_fraction	0.812
bagging_freq	1
colsample_bytree	0.755
Ensemble Model	
early stopping	true
cost function	MAPE

Heatmap Legend: Load: power consumption, Week: Week of year; Day: Day of week; Time: Time of day; Temp: Temperature; Hum: Humidity; EWMA: smoothed data using EWMA; LT: Load transform; LH1, LH2, LH3, and LH4: Lag for load 1 hour, 2 hours, 3 hours, and 4 hours prior, respectively; LD1, LD2, and LD3: Lag for load 1 day 1 hour, 2 days 1 hour, and 3 days 1 hour, respectively; LW3: Lag for load 1 week 3 hour prior.

TABLE III
Comparison of Proposed Model with Models from Literature. Best Performance is inked in blue. ↑ and ↓ Denotes a Positive and Negative Improvement, Respectively

Model	MAPE	% Improvement	Runtime (s)
Ensemble Model [19]	1.50	Baseline	16.63
Ensemble Model [24]	1.29	↑ 14.0	-
XGboost Model [19]	2.80	↓ 86.6	14.75
LightGBM [19]	2.41	↓ 60.0	1.98
ANN [8]	1.39	↑ 7.3	-
XGboost (ours)	0.997	↑ 33.5	2.82
LightGBM (ours)	0.954	↑ 36.4	4.12
Proposed CE model	0.943	↑ 37.1	4.13

Note: The LightGBM model's runtime is 1.30 seconds, but it uses a feature generated by the XGboost model to improve accuracy. This increased the model's runtime by 2.82 seconds to a total of 4.12 seconds.

engaging in activities that vary from day to day and week to week. This gainsays the model from estimating the exact consumption patterns at those times. However, throughout the morning, the consumers' activities are typically quite similar from day to day allowing the model to capture the pattern precisely, resulting in accurate prediction.

C. Quantitative Analysis

Table III provides a thorough comparative analysis of the proposed XGboost-LightGBM CE based STLF model with state-of-the-art algorithms found in recent literature. The proposed solution achieves the smallest MAPE of 0.943, which is an improvement of 37% compared to the baseline. It also records a low runtime of 4.13 seconds, which is 4 times faster compared to the same baseline. Besides this, the PSIT of the model is found to be 1.496×10^{-4} seconds using the computational environment outlined in Section IV-A. These performances of the proposed CE model and the unimodality XGboost and LightGBM models built in this work surmount the other existing solutions by a significant margin. While the XGboost model built in this research offers an improvement over the baseline model, the proposed CE model offers a greater level of accurate load prediction, achieving a 5.4% lower MAPE than the standalone XGboost model. Similarly, the LightGBM model built in this work also offers an improvement over the baseline model but still does not reach the level of correctness of the CE model. It is important to note that the LightGBM model uses the prediction of the XGboost model as one of its input features to enhance its prediction accuracy. Thus, the performance of the CE model is improved by 1.2% when compared to the unimodality LightGBM model in terms of MAPE, while overhead in runtime is negligible.

V. Conclusion

This paper proposed an advanced load prediction framework by exploiting novel feature engineering and feature selection schemes and a CE model. The proposed solution not only offers a highly accurate load prediction but also a reduction in the computation time compared to the current baseline model. The ablation study conducted in this research demonstrated

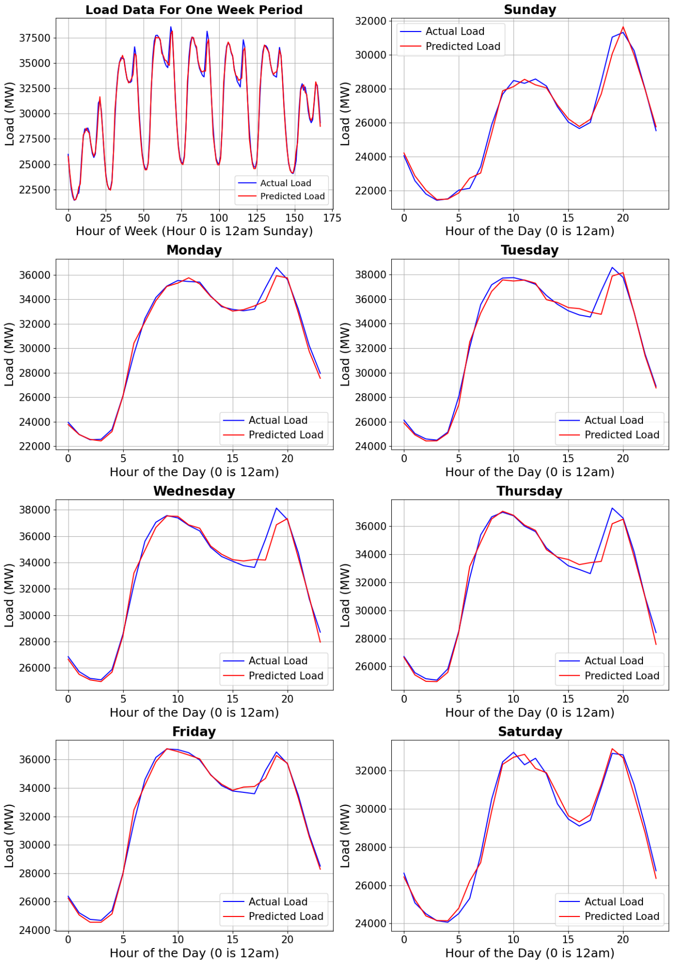


Fig. 3. Model's qualitative performance evaluation on the mutually exclusive test set with respect to daily and weekly load pattern.

the importance of (i) integrating multimodal data (e.g., load pattern and weather information), (ii) selecting key features and capturing seasonality in the time series data, and (iii) harnessing the strengths of weak predictors under a unified architecture. Additionally, this work identified the following as future directions: (i) Enhancing the model to forecast the load far ahead of the time (i.e., a long look-ahead or future time) without any compromising on the prediction accuracy; and (ii) investigating transfer learning approaches to improve the model's generalizability. The former direction will increase the viability of the model for deployment in the industry, while the latter one will allow the model to be implemented in a wide variety of locations and require minimal training to achieve top-notch results.

Acknowledgment

The authors would like to thank Mr. Christopher Silver for his help and guidance throughout the model construction phase of this work.

References

[1] A. Husain, A day without power: Outage costs for businesses, https://www.bloomenergy.com/blog/a-day-without-power-outage-costs-for-businesses

[2] "Electricity transformation Canada," The Big Switch - Electricity in Canada

[3] B.-M. Hodge, D. Lew and M. Milligan, "Short-term load forecast error distributions and implications for renewable integration studies", Proc. IEEE Green Technol. Conf. (GreenTech), pp. 435-442, Apr. 2013.

[4] H. M. Al-Hamadi and S. A. Soliman, "Long-term/mid-term electric load forecasting based on short-term correlation and annual growth", Electr. Power Syst. Res., vol. 74, no. 3, pp. 353-361, Jun. 2005.

[5] T. Hong, J. Wilson and J. Xie, "Long term probabilistic load forecasting and normalization with hourly information", IEEE Trans. Smart Grid, vol. 5, no. 1, pp. 456-462, Jan. 2014.

[6] W. Kong, Z. Y. Dong, Y. Jia, D. J. Hill, Y. Xu and Y. Zhang, "Short-term residential load forecasting based on LSTM recurrent neural network", IEEE Trans. Smart Grid, vol. 10, no. 1, Jan. 2019.

[7] Groß, A., Lenders, A., Schwenker, F. et al. Comparison of short-term electrical load forecasting methods for different building types. Energy Inform 4 (Suppl 3), 13 (2021).

[8] S. Singh, S. Hussain and M. A. Bazaz, "Short-term load forecasting using artificial neural network," 2017 Fourth International Conference on Image Information Processing (ICIIP), Shimla, India, 2017, pp. 1-5.

[9] K. Chen, Q. Wang, Z. He, J. Hu and J. He, "Short-Term Load Forecasting With Deep Residual Networks," in IEEE Transactions on Smart Grid, vol. 10, no. 4, July 2019.

[10] V. Y. Kondaiah and B. Saravanan, "Short-Term Load Forecasting With Deep Learning," 2021 Innovations in Power and Advanced Computing Technologies (i-PACT), Kuala Lumpur, Malaysia, 2021, pp. 1-5.

[11] X. Yao, X. Fu and C. Zong, "Short-Term Load Forecasting Method Based on Feature Preference Strategy and LightGBM-XGboost," in IEEE Access, vol. 10, pp. 75257-75268, 2022.

[12] A. Irankhah, S. R. Saatlou, M. H. Yaghmaee, S. Ershadi-Nasab and M. Alishahi, "A parallel CNN-BiGRU network for short-term load forecasting in demand-side management," 2022 12th International Conference on Computer and Knowledge Engineering (ICCKE), Mashhad, Iran, Isla Republic of, 2022, pp. 511-516.

[13] H. Ur-Rehman, S. Mujeeb and N. Javaid, "DCNN and LDA-RF-RFE Based Short-Term Electricity Load and Price Forecasting," 2019 International Conference on Frontiers of Information Technology (FIT), Islamabad, Pakistan, 2019, pp. 71-715.

[14] N. Jhana, "Hourly Energy Demand Generation and weather," Kaggle.

[15] D. Herzmann, Iowa Environmental Mesonet, Spanish Weather Data, updated: 2023-11-24, https://mesonet.agron.iastate.edu/.

[16] V. Y. Kondaiah and B. Saravanan, "Short-Term Load Forecasting With Deep Learning," 2021 Innovations in Power and Advanced Computing Technologies (i-PACT), Kuala Lumpur, Malaysia, 2021, pp. 1-5.

[17] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, Aug. 2016, pp. 785–794

[18] Bentéjac, Candice and Csörgő, Anna and Martínez-Muñoz, Gonzalo. (2019). A Comparative Analysis of XGBoost

[19] X. Yao, X. Fu and C. Zong, "Short-Term Load Forecasting Method Based on Feature Preference Strategy and LightGBM-XGboost," in IEEE Access, vol. 10, pp. 75257-75268, 2022.

[20] A. Preciado, "LIGHTGBM algorithm: An end to end review on the trees to DataFrame method," LightGBM Algorithm: An end to end review on the trees to DataFrame method,

[21] M. Alhamid, "Ensemble models: What are they and when should you use them?," https://builtin.com/machine-learning/ensemble-model

[22] H. Lian et al., "A Short-Term LOAD forecasting Method Based on EEMD-LN-GRU," 2021 3rd Asia Energy and Electrical Engineering Symposium (AEEES), Chengdu, China, 2021, pp. 666-673.

[23] Y. Hu, B. Qu, J. Wang, J. Liang, Y. Wang, K. Yu, Y. Li, K. Qiao, "Short-term load forecasting using multimodal evolutionary algorithm and random vector functional link network based ensemble learning," Applied Energy, Volume 285, 2021, 116415, ISSN 0306-2619

[24] L. Wang, S. Mao, B. M. Wilamowski and R. M. Nelms, "Ensemble Learning for Load Forecasting," in IEEE Transactions on Green Communications and Networking, vol. 4, no. 2, pp. 616-628, June 2020.

[25] S. Hosseini, A. Shahbandegan, and T. Akilan, "Deep Neural Network Modeling for Accurate Electric Motor Temperature Prediction," in 2022 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE), pp. 170–175, October 2022.