IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Digital Object Identifier

Short Term Load Forecasting Based on Ensemble Model:GRU-LGBM fusion

SHIJON DAS
Idaho State University, Pocatello, CO 83201 USA (e-mail: shijondas@isu.edu)

ABSTRACT This study presents a method for Short-term Load Forecasting (STLF) that combines Light Gradient Boosting Machines (LGBM) and Gated Recurrent Units (GRU) in an ensemble model. The combination of sequential learning exhibited by GRU and the ensemble power of LGBM outperforms traditional approaches, which are frequently inadequate in capturing complicated load patterns. Seasonality, weather, and load patterns are all taken into account while training the model using historical load data. The better forecasting accuracy of the ensemble GRU-LGBM is demonstrated by real-world dataset results, offering a reliable solution for dynamic load fluctuations. This study advances STLF approaches by providing a flexible framework for the best grid management.

INDEX TERMS ELM, GRNN, GRU, LGBM, LSTM, Random Forest, Random Search, Stacked, Look Back Window.

I. INTRODUCTION

Regarding power management, utility firms have two primary responsibilities: responding to demand and guaranteeing consistent performance. Accurately forecasting power demand is a critical responsibility for electricity providers since it directly affects the avoidance of problems like load shortages and interruptions in distribution networks. Electricity providers want to keep costs associated with producing and distributing electricity as low as possible, which is why accurate forecasting is important [1], [2].To plan load generation and distribution units and make educated decisions about energy production and purchase, load forecasting provides advance insights into electricity demand [1]. Power generating and distribution networks are greatly impacted by the accuracy of electrical demand projections; even a little prediction error can result in large financial losses. Notably, as seen in [3], [4], a 1% reduction in prediction inaccuracy leads to £10 million in cost savings for energy operator expenses.

Short-term load forecasting (VSTLF), short-term load forecasting (STLF), medium-term load forecasting (MTLF), and long-term load forecasting (LTLF) are the general categories into which electrical load forecasting models can be divided based on prediction durations [5], [6]. Whereas STLF forecasts load for the next hour to a week, VSTLF focuses on the next few minutes to an hour. For production planning and decision-making in power market tenders, STLF is especially important. The following several weeks to a year and beyond are covered by medium- and long-term forecasting, respectively. These forecasts concentrate on maintenance planning and the construction of electricity generating, transmission, and distribution infrastructure [5], [6].

For this study, an ensemble GRU-LGBM network forecasting is developed for predicting short term load. First, the best GRU with the best hyperparameters is used to predict load and a look back window is used to fiure out the best look out steps and then the predictions from LGBM along with the predictions from GRU are fed into a neural network for the ensemble model. Section IV, presents the data set preparation and the preprocessing along with the implementail details of the ensemble model and in Section VI, the model's performance in comparision to other forecasting models are discussed.

A. ABBREVIATIONS AND ACRONYMS

Gated Recurrent Unit(GRU), Long Short Term Memory(LSTM), General Regression Neural Network(GRNN), Extreme Learning Machines(ELM)

II. MOTIVATION

Modern civilization depends heavily on electricity, and precise short-term load demand forecasting is essential to the dependable and effective operation of power systems. For load forecasting, conventional techniques like as statistical and time series models have been used over the years. However, these traditional methods are severely challenged by the dynamic and nonlinear character of load patterns, which are influenced by a variety of external factors.

The increasing complexity of energy systems combined with the shortcomings of current models in simulating complicated load dynamics highlights the need for more advanced and flexible methods. Neural networks, and deep learning architectures in particular, have emerged recently and have demonstrated incredible promise for capturing complex connections and patterns in load data. The complete potential of neural networks in short-term load forecasting is still largely unrealized, despite their potential.

The motivation for this research stems from the observation that existing load forecasting models often fall short of delivering the required accuracy, especially in scenarios involving rapid changes in demand patterns. The limitations of traditional models in handling non-linear relationships and capturing temporal dependencies necessitate a paradigm shift towards advanced techniques, such as neural networks.

Recent advances in machine learning, especially the effectiveness of long short-term memory networks (LSTMs) and recurrent neural networks (RNNs), have shown that these models are suitable for short-term load forecasting because they are good at capturing complex temporal patterns. Additionally, the combination of neural networks with ensemble methods, like the Gradient Boosting Machine (GBM), has shown the potential to further enhance predictive performance.

To increase the accuracy of short-term load forecasting, this study intends to take advantage of the developments in neural network designs by investigating the possibilities of ensemble approaches like LightGBM and models like Gated Recurrent Units (GRUs). To help power system operators make better decisions and maintain the stability and dependability of electricity grids, we aim to contribute to the development of more accurate and robust load forecasting models by addressing the shortcomings of earlier forecasting techniques and embracing the capabilities provided by these cutting-edge approaches.

III. RELATED WORKS

Statistical techniques are primarily used in conventional procedures. These include techniques for autoregressive integrated moving normal (ARIMA) [6], exponential smoothing [7], and multiple linear regression [8]. It should be emphasized that the aforementioned methodologies may not consistently produce adequate results in short-term load forecasting (STLF) due to the nonlinear properties of time series univariate load data [9]. Machine learning-based techniques have been developed and widely used in STLF to overcome this constraint [10]. Clustering [11], fuzzy logic frameworks [12], support vector machines (SVM) [13], artificial neural networks (ANN) [13], radial basis functional networks (RBFN) [14], and hybrid methodologies are some of these techniques.

A kernel-based Support Vector Machine (SVM) model for forecasting electric load is proposed in [15], presenting a new method for choosing the kernel function. Real-world examples from the California Power Grid and Australia Power Grid are used to assess the model's performance. To forecast hourly loads, [16] provides a fuzzy logic-based approach that integrates historical load data with time and day (weekends or weekdays). The aim is to ascertain the probable load curve for a given day by utilizing a year's worth of data from a major industry.

Additionally, because recurrent neural networks (RNNs) are good at learning non-stationary load data patterns, they are used for STLF. [26] uses an RNN approach to anticipate domestic load and shows improved root mean square error (RMSE) performance. Using RNN, a contemporary load forecasting methodology is shown in [17], showing respectable performance with less fluctuations than other models in both high and low power demand zones. The lack of vanishing gradient issues has made gated recurrent unit networks (GRUs) more popular [18]. [19] presents a minimum mean average percentage error (MAPE) for STLF with multi-source data that is based on GRU and performs better than other existing techniques.

IV. METHODOLOGY

A. DATASET

For this study, the hourly load data of the Western region from January 1, 2012, to December 31, 2015 by the Electric Reliability Council of Texas(ERCOT) has been used. The dataset is enriched with hourly weather data such as temperature and humidity. The hourly data for temperature and humidity of the Western region is gathered from the website Visual Crossing for the years 2012 to 2015. The sampling frequency is 24 points per day. As a result, a total of 35064 (24 hours * 365 days * 3 years + 24 hours * 366 days) data points were obtained. Apart from the weekends, the dataset also includes National Holidays in its holiday feature. Pearson product-moment correlation coefficient formula is used for figuring out the correlation between load temperature and humidity [20].

$$\rho = \frac{\mathrm{cov}(X, Y)}{\sigma_X \sigma_Y} \tag{1}$$

Here, $\mathrm{cov}(X, Y)$ represents the covariance between variables X and Y, while σ_X and σ_Y are the standard deviations of X and Y, respectively. The coefficient ρ ranges from -1 to 1, where a positive value indicates a positive correlation, a negative value indicates a negative correlation, and a value of 0 indicates no linear correlation between the variables.

Table 1 demonstrates the correlation between electric load temperature and humidity. Electric load shares a positive correlation with temperature while the presence of humidity seems to affect electric load negatively. Hence, the two features were included in the dataset for load prediction in this paper.

TABLE 1. Correlation Between Load and Weather Variables

	Temperature	Humidity
Load	0.4028	-0.4111

A rigorous analysis of the dataset showed great fluctuations in load data and the heatmap correlation showed how the load data is affected by weather, temperature, holidays, and time of the day. During summer and winter as well as on weekends the load values seem to be much higher than the load values on rest of the days.

B. DATA PREPROCESSING

The HourEnding feature from the ERCOT dataset was used to extract the five features for our dataset. The first features Day of the Week(0-6) denotes the day of the week with 0 being Monday and 6 being Sunday. Day of the Month(1-31) represents which day of the month the load data is recorded for while the third feature Month of the Year(1-12) denotes which month of the year it is with 1 being Januray and 12 being December. The fourth feature Time of the Day(0-23) captures the hour when the load data was recorded. For the feature, Holidays, one-hot encoding is used to process the weekday and national holidays information where 1 represents that the day was a holiday and 0 indicates the day was not a holiday. The weekends and national holidays were mapped to 1 and the weekdays were mapped to 0.

This study considered the architecture of the GRU network model and the LGBM network mode and applied different preprocessing techniques for better accuracy in load prediction.

The dataset is loaded into a pandas dataframe and the target variable, which is the Load West column, is converted to numeric values. After converting the Load West column to numeric values, the dataset was cleaned through removal of null values and outliers. Pauta Criterion was used for detecting outliers [21]. The equation shows the formula used for detecting the outliers.

The condition for identifying outliers using the Pauta criterion can be expressed mathematically as:

$$X_{\text{outliers}} = \{x_i \mid x_i > (\mu + 3\theta) \text{ or } x_i < (\mu - 3\theta)\} \tag{2}$$

where:

- X_{outliers} is the set of outliers.
- x_i represents an individual data point.
- μ is the mean of the dataset.
- θ is the median absolute deviation (MAD) from the median.

After cleaning the dataset, the numerical data of the GRU network, including Day of Week(0-6), Day of the Month(1-31), Month of the Year(1-12), Time of Day(0-23), Holidays, Humidity, Temperature and Load WEST is normalized using Scikit Learn's MinMaxScaler.This function normalizes all the feature values to a range between 0 and 1. This normalization ensures that all features contribute equally to the model's learning process and prevents features with larger magnitudes from dominating the learning process. Through normalization, the model can converge quickly during training and can optimize the learning rate.

$$x_{\text{normalized}} = \frac{x - \min(X)}{\max(X) - \min(X)} \tag{3}$$

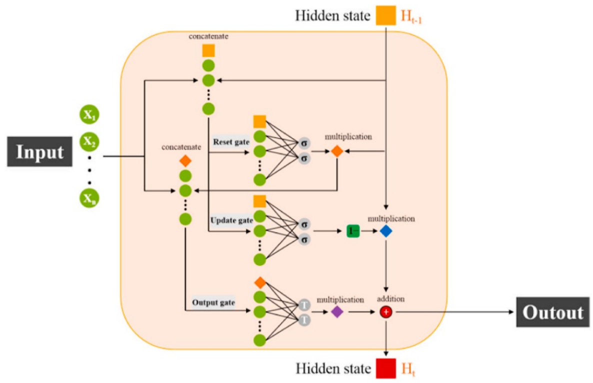

FIGURE 1. Structure of the GRU Model

where:

- $x_{\text{normalized}}$ is the normalized value of the feature x.
- X is the set of all values of the feature.
- $\min(X)$ is the minimum value in the set X.
- $\max(X)$ is the maximum value in the set X.

C. MODEL SPECIFICATION

This section gives an overview of the machine learning models applied on the dataset. GRU and LGBM are the machine learning models chosen for this study.

1) Gated Recurrent Unit(GRU)

Within the realm of recurrent neural networks (RNNs), the GRU neural network stands out as an optimized model. The GRU network differs from the well-known LSTM network in that it has a less visible forget gate, allowing for a smoother convergence process. It exerts control over the flow of data information via a specialized gate structure. The GRU modifies the calculation mode of a hidden layer state variable in cyclic neural networks by using resetting and updating gates, resulting in an improvement over traditional RNNs. The reset gate governs the extent to which previous state information is ignored, with smaller values indicating a greater degree of neglect. The update gate, on the other hand, manages the transfer of status information from the previous moment to the current one, with larger values implying a greater inclusion of state information. Reset gates are particularly effective at capturing short-term dependencies in temporal data, whereas update gates are particularly effective at capturing long-term dependencies in sequential data [21]. Fig.1 shows the structure of a GRU.

2) LGBM Network

One type of decision tree that isn't dependent on a single tree is the Gradient Boosting Decision Tree (GBDT), which is made up of several decision trees. Its excellent generalization ability and ability to prevent over-fitting make it superior to standard decision trees. Regression and classification issues are frequent uses for it. Iterating the decision tree algorithm is the key concept. The input for the subsequent decision tree will be the residual from the preceding decision tree. Iteration reduces the deviation, and the ultimate result is obtained by adding up the outputs of various decision trees. Assuming that the model has M trees, the predicted value of the model is $\hat{y} = F(x)$, and the mean square error is the loss function. Each stage j $(1 \leq j \leq M)$ initially has an imperfect model F_j, and assume:

$$F_{j+1}(x) = F_j(x) + h(x) \tag{4}$$

where $h(x)$ is the estimator and should try to make:

$$F_{j+1}(x) = F_j(x) + h(x) = y \tag{5}$$

To get equivalently:

$$h(x) = y - F_j(x) \tag{6}$$

Therefore, the weak classifier can get $h(x)$ by fitting the residual $y - F_j(x)$. In addition, it is observed that $y - F_j(x)$ is the negative gradient of the mean square error, so GBDT is a gradient descent algorithm, that obtains different gradients according to different loss functions. The most recent framework that can execute the GBDT algorithm is called LGBM (Light Gradient Boosting Machine). In contrast to certain initial frameworks, LGBM exhibits increased speed, reduced memory usage, and the ability to accommodate dispersed features. Due to its use of a histogram technique and lack of indexing requirements, LGBM significantly minimizes memory and space requirements. To minimize the amount of features, it also employs a mutually exclusive feature method. For tree growth, LGBM selects a lead-wise growth strategy [22].

3) GRU-LGBM Network Model Construction

The GRU-LGBM network model is formed by stacking the predictions from both the GRU and the LGBM models. While GRU performs feature extraction and prediction, LGBM uses integrated machine learning algorithms to fit and predict load data. The predictions are stacked and fed into a neural network for the ensemble model. Here, the stacking algorithm is used for model fusion. Fig. 2 shows the prediction flowchart of the GRU-LGBM model.

The model construction involves a multi-step process that combines deep learning and gradient-boosting techniques for effective load prediction. After the dataset is preprocessed and the features are normalized, the primary deep learning component employs a GRU neural network. Keras Tuner library is used to configure the GRU through a hyperparameter tuning process. The process involves searching for the optimal parameters for the number of GRU units, the number of intermediate layers, and the number of units in the last layer. A random search approach was taken to determine the best hyperparameters. After determining these optimized hyperparameters, the final GRU model was constructed using the best hyperparameters.

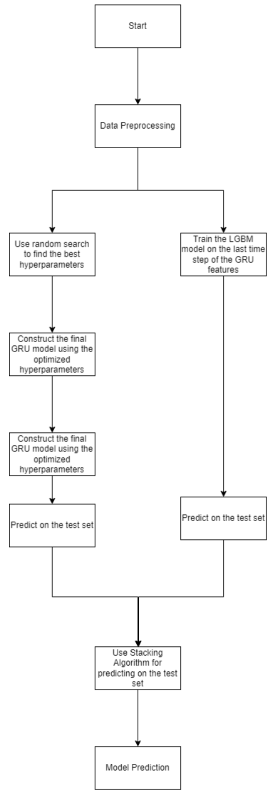

FIGURE 2. Structure of the GRU Model

The final GRU model had 60 units in the first layer. The first layer was followed by two subsequent layers which had 60 units each as well. The last layer had 50 units. The best look back window is determined for the optimal hyperparameters. The look-back window in the context of GRU for load forecasting determines the number of past time steps considered when creating input sequences for the model. A larger look-back window captures more historical context, enabling the LSTM to learn longer-term dependencies in the time series data, while a smaller window emphasizes short-term patterns. The best look-back steps were found to be 10 with a MAPE of 0.89.

Additionally, a LGBM model is integrated into the ensemble. The LGBM model is trained on the last step of the GRU features. The predictions from both the GRU and LGBM models are then stacked and fed into a final neural network layer for the ensemble model. This layer consists of a single dense unit and is trained using the Adam optimizer with mean squared error loss.

V. MODEL PERFORMANCE EVALUATION

For this study, three evaluation metrics have been used to measure the performance of the model. Mean Absolute Error, Mean Absolute Percentage Error(MAPE) and Root Mean Squared Error(RMSE) are the three evaluation metrics used. MAE measures the average absolute difference between the predicted values and the true values. The calculation method is shown in Equation (8). MAPE calculates the percentage difference between predicted and true values using Equation(9). RMSE uses Equation(10) to measure the square root of the average squared difference between predicted and true values.MAPE calculates the percentage difference between predicted and true values.

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}|\hat{y}_i - y_i| \quad (7)$$

$$\mathrm{MAPE} = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{|\hat{y}_i - y_i|}{|y_i|}\right) \times 100 \quad (8)$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2} \quad (9)$$

VI. RESULTS AND DISCUSSIONS

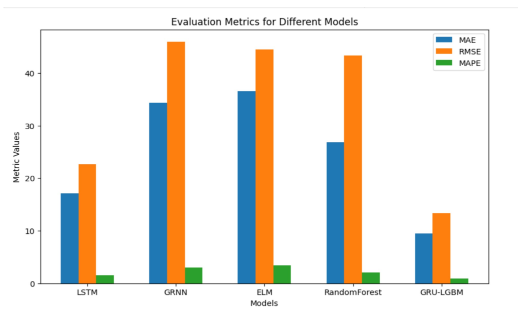

FIGURE 3. Bar chart showing evaluation metrics of LSTM, GRNN, ELM, RandomForest and GRU-LGBM models

Fig. 3 shows how the different models performed in comparison to the GRU-LGBM in predicting the load. The GRU-LGBM model stands out as the most accurate amongst the models evaluated, with significantly lower MAE, RMSE, and MAPE compared to other models.

The LSTM model exhibits a MAE of 17.11, RMSE of 22.65, and MAPE of 1.52, indicating a solid predictive performance. However, the GRNN and ELM models present higher errors, with GRNN yielding a MAE of 34.35, RMSE of 45.98, and MAPE of 3.05, and ELM showing a MAE of 36.59, RMSE of 44.52, and MAPE of 3.44. Random Forest, while outperforming GRNN and ELM, still shows a relatively high MAE of 26.78, RMSE of 43.35, and MAPE of 2.03.

The model results for GRNN, LSTM and EML were obtained from the reference paper titled Short-Term Load Forecasting Using an LSTM Neural Network [20]. This paper used the same ERCOT dataset of the same timeframe with similar features.

It is observed that traditional machine learning models like Random Forest, Extreme Learning Machine (ELM), and General Regression Neural Network (GRNN) are outperformed GRU-LGBM model with a MAE of 9.44, RMSE of 13.32, and MAPE of 0.86.

The GRU excels at capturing sequential dependencies in time-series data. It effectively learns the temporal patterns and dependencies present in the historical load data. On the other hand, LGBM is adept at capturing non-linear relationships and interactions among features. The combination of these models allows the GRU to focus on the temporal aspects, providing crucial information to the LGBM model, which, in turn, refines the predictions by considering broader feature interactions.

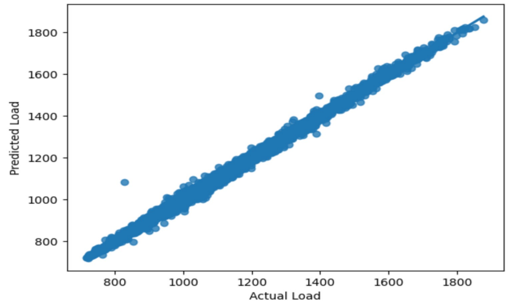

FIGURE 4. Regression plot for load prediction

Fig. 4 shows the regression plot for the actual and the predicted load values. The correlation between actual and predicted load values is 0.9983. The results show that the model can accurately predict load. Fig. 5 shows the Bland-

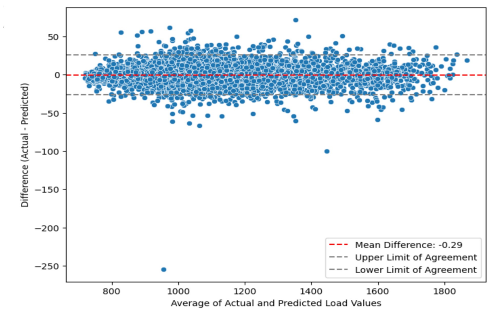

FIGURE 5. Bland-Altman for load prediction

Altman analysis plot for actual and predicted load values. Bland-Altman plot is a tool that supports a visual representation of data consistency. It can be seen from the figure that 98.2 percent of the data is distributed within the consistency boundary. The results show that the predicted load values are in good agreement with the actual values.

VII. CONCLUSION

The superior performance of the GRU-LGBM model suggests that the combination of a recurrent neural network with a gradient-boosting model addresses the intricacies of load forecasting more effectively. The hybrid model leverages the GRU's strength in sequential data processing and temporal pattern recognition, complemented by LGBM's prowess in capturing complex feature interactions. This nuanced understanding of temporal and feature-related aspects contributes to the GRU-LGBM model's exceptional accuracy, making it a compelling choice for load forecasting applications.

REFERENCES

[1] K. Chen, Q. Wang, Z. He, J. Hu, and J. He, "Short-term load forecasting with deep residual networks," *IEEE Trans. Smart Grid*, vol. 10, no. 4, pp. 3943–3952, 2019.

[2] M. Imani, "Deep learning based electricity demand forecasting in different domains," *Iranian J. Energy Environ.*, vol. 11, no. 1, pp. 33–39, 2020.

[3] D. Bunn and E. Farmer, *Comparative Models for Electrical Load Forecasting, 1st Ed.* Wiley, New York, NY, USA, 1985.

[4] G. Cerne, D. Dovzan, and I. Skrjanc, "Short-term load forecasting by separating daily profiles and using a single fuzzy model across the entire domain," *IEEE Trans. Industrial Electron.*, vol. 65, no. 9, pp. 7406–7415, 2018.

[5] S. Fan and R. Hyndman, "Short-term load forecasting based on a semi-parametric additive model," *IEEE Trans. Power Syst.*, vol. 27, no. 1, pp. 134–141, 2012.

[6] Filik, O. Gerek, and M. Kurban, "Neural and mathematical modeling approaches for hourly long term load forecasting," *ICIC Express Lett*, vol. 3, no. 3, pp. 1–6, 2009.

[7] W. Christiaanse, "Short-term load forecasting using general exponential smoothing," *IEEE Trans. Power App. Syst.*, vol. PAS-90, no. 2, pp. 900–911, Mar 1971.

[8] W. Charytoniuk, M. S. Chen, and P. Van Olinda, "Nonparametric regression based short-term load forecasting," *IEEE Trans. Power Syst.*, vol. 13, no. 3, pp. 725–730, May 1998.

[9] C.-M. Lee and C.-N. Ko, "Short-term load forecasting using lifting scheme and arima models," *Expert Syst. Appl.*, vol. 38, no. 5, pp. 5902–5911, May 2011.

[10] T. Hong, P. Pinson, Y. Wang, R. Weron, D. Yang, and H. Zareipour, "Energy forecasting: A review and outlook," *IEEE Open Access J. Power Energy*, vol. 7, pp. 376–388, 2020.

[11] A. Jain and B. Satish, "Clustering based short term load forecasting using support vector machines," in *Proc. IEEE Bucharest PowerTech*, 2009, pp. 1–8.

[12] H. H. Çevik and M. Çunkaş, "Short-term load forecasting using fuzzy logic and anfis," *Neural Comput. Appl.*, vol. 26, no. 6, pp. 1355–1367, Aug 2015.

[13] Y. Chen, P. Xu, Y. Chu, W. Li, Y. Wu, L. Ni, Y. Bao, and K. Wang, "Short-term electrical load forecasting using the support vector regression (svr) model to calculate the demand response baseline for office buildings," *Appl. Energy*, vol. 195, pp. 659–670, Jun 2017.

[14] W.-Y. Chang, "Short-term load forecasting using radial basis function neural network," *J. Comput. Commun.*, vol. 03, no. 11, pp. 40–45, 2015.

[15] J. Che and J. Wang, "Short-term load forecasting using a kernel-based support vector regression combination model," *Appl. Energy*, vol. 132, pp. 602–609, Nov 2014.

[16] P. Ganguly, A. Kalam, and A. Zayegh, "Short-term load forecasting using fuzzy logic," in *Int. Conf. Res. Educ. Sci.*, 2017, pp. 1–5.

[17] L.-Y. Wei, "A study of the hybrid recurrent neural network model for electricity loads forecasting," *Int. J. Acad. Res. Accounting, Finance Manage. Sci.*, vol. 7, no. 2, pp. 21–29, Apr 2017.

[18] L. Kuan, Z. Yan, W. Xin, C. Yan, P. Xiangkun, S. Wenxue, J. Zhe, Z. Yong, X. Nan, and Z. Xin, "Short-term electricity load forecasting method based on multilayered self-normalizing gru network," in *Proc. IEEE Conf. Energy Internet Energy Syst. Integr. (EI2)*, 2017, pp. 1–5.

[19] Y. Wang, M. Liu, Z. Bao, and S. Zhang, "Short-term load forecasting with multi-source data using gated recurrent unit neural networks," *Energies*, vol. 11, no. 5, p. 1138, May 2018.

[20] M. S. Hossain and H. Mahmood, "Short-term load forecasting using an lstm neural network," in *2020 IEEE Power and Energy Conference at Illinois (PECI)*, Champaign, IL, USA, 2020, doi: http://doi.org/10.1109/PECI48348.2020.906465410.1109/PECI48348.2020.9064654. pp. 1–6.

[21] Y. Jiang, C. Li, L. Sun, D. Guo, Y. Zhang, and W. Wang, "A deep learning algorithm for multi-source data fusion to predict water quality of urban sewer networks," *Journal of Cleaner Production*, vol. 318, p. 128533, 2021, doi: http://doi.org/10.1016/j.jclepro.2021.12853310.1016/j.jclepro.2021.128533.

[22] W. Qian, C. Gu, C. Zhu, Z. Jiang, B. Han, and M. Yu, "Short-term load forecasting based on multi-model fusion of cnn-lstm-lgbm," in *2021 International Conference on Power System Technology (POWERCON)*, Haikou, China, 2021, doi: http://doi.org/10.1109/POWERCON53785.2021.969761910.1109/POWERCON53785.2021.9697619. pp. 934–939.

• • •