Journal of Computer Science and Software Applications
https://www.mfacademia.org/index.php/jcssa
ISSN:2377-0430
Vol. 6, No. 1, 2026

Data-Driven Demand Forecasting Based on an Ensemble LSTM Model

Kaveh Khalilpour[1], Xinying Chen[2]
[1]Charles Darwin University, Darwin, Australia
[2]Charles Darwin University, Darwin, Australia
*Corresponding Author: Xinying Chen; cxy66676@gmail.com

Abstract:

User participation in power grid dispatching can effectively enhance grid flexibility; however, the inherent uncertainty of user behavior constrains the development of demand response. To address this issue, an incentive-based demand response implementation framework is first established, and the mechanism by which load aggregators (LAs) integrate demand-side resources to participate in electricity market operations is analyzed. User behavioral responses under incentive policies are further transformed into demand elasticity.Subsequently, based on the long short-term memory (LSTM) network, a data-driven demand elasticity forecasting method using an ensemble LSTM model is proposed. To improve forecasting performance, the source data are processed through smoothing and scaling operations, and weighting coefficients are incorporated into the loss function to enhance model robustness. Case study results demonstrate that, compared with the conventional LSTM model and the k-nearest neighbor (kNN) forecasting method, the proposed approach reduces the average prediction error of user demand elasticity by 5.33% and 28.8%, respectively. For total load forecasting, the mean absolute percentage error (MAPE) is reduced by 2.06% and 3.09%, respectively. In addition, the effects of smoothing and scaling preprocessing techniques on forecasting accuracy are investigated based on the ensemble LSTM model. The results indicate that appropriate preprocessing of the original data can effectively improve prediction accuracy.

Keywords:

Ensemble long short-term memory (LSTM), demand elasticity, data preprocessing, electricity market, incentive-based demand response, data-driven modeling.

1. Introduction

Demand response technologies can effectively mobilize user-side resources and encourage users to actively participate in electricity markets, thereby alleviating the decline in power system flexibility caused by the large-scale integration of renewable energy sources [1]-[2]. With the increasing diversity of electric loads, it has become difficult for power grids to conduct transactions directly with individual users, while a single user is also unable to independently satisfy the requirements of grid operation. To address this issue, the load aggregator (LA) model has been proposed, in which various types of user-side loads are integrated and managed in a unified manner [3].

By signing contracts with users, LAs can implement control over user behaviors. In order to maximize their own benefits, LAs are required to accurately predict user behaviors. At present, research on load forecasting has become relatively mature, and commonly used methods include neural networks, support vector machines, and k-nearest neighbor (kNN) algorithms [4]. However, these forecasting methods generally do not consider the participation of loads in electricity markets. In recent years, extensive studies have been

conducted on load forecasting under the LA framework. Nevertheless, the uncertainty introduced by user participation in electricity markets significantly constrains forecasting accuracy.

In the literature, fuzzy parameters have been employed to characterize the uncertainty of user participation in demand response under the LA framework [5]; however, this approach still fails to accurately describe user uncertainty. In [6], a shortened forecasting horizon was adopted, and a sequence of models, including data preprocessing, load forecasting, and error forecasting models, was constructed. Although this method improves prediction performance, it involves relatively high computational complexity. In [7], a radial basis function (RBF) neural network was introduced into load forecasting under the LA framework, but the model must be retrained for different load types. Furthermore, short-term load forecasting models based on long short-term memory (LSTM) networks have been established in [8]-[10]. To improve forecasting accuracy, multidimensional input features were quantified; however, the prediction performance still requires further enhancement.

Based on the above analysis, it can be concluded that when users participate in electricity markets, the uncertainty of their behaviors severely limits forecasting accuracy. Therefore, this paper focuses on load forecasting under the LA framework. First, under the LA model, behavioral variations caused by incentive price fluctuations are transformed into demand elasticity. Then, by analyzing the characteristics of the LSTM algorithm, an ensemble LSTM-based incentive-driven demand response load forecasting method is proposed. Finally, case studies demonstrate that the proposed method can effectively improve the forecasting accuracy of both demand elasticity and total load. In addition, it is verified that smoothing and scaling preprocessing of source data, as well as the introduction of weighting coefficients into the loss function, can further enhance forecasting performance.

2. Related Work

Demand response (DR) has long been recognized as an effective strategy to enhance grid flexibility, particularly in scenarios with high penetration of renewable energy. Early research summarized the core principles and classifications of DR programs in electricity markets [11], while subsequent studies quantified its impact on market operation and user-side participation [12]. Classical statistical models, such as triple seasonal methods, have been utilized to handle temporal complexity in short-term load forecasting [13].

With the advancement of deep learning, methods such as long short-term memory (LSTM) networks have demonstrated strong capabilities in capturing temporal dependencies in load data. To further improve accuracy, hybrid approaches have been proposed that combine signal decomposition techniques like discrete wavelet transform with LSTM to capture both high- and low-frequency load patterns [14]. Reviews of neural network-based forecasting methods emphasize their effectiveness in modeling nonlinear load behaviors under dynamic market conditions [15].

Ensemble learning has emerged as a powerful strategy to improve forecasting robustness and accuracy. Techniques that integrate multiple learning models and address issues such as missing value imputation have shown significant performance improvements [16]. Additionally, time series forecasting methods have incorporated advanced preprocessing mechanisms like second-order differencing to adapt to non-stationary data, contributing to model stability and generalization [17].

In parallel, resource orchestration methods based on reinforcement learning (RL), including multi-agent frameworks, have been applied to dynamic, uncertain environments such as cloud-native systems and microservice clusters. These approaches introduce decision-making strategies that are adaptable and self-optimizing, offering methodological inspiration for adaptive DR control in power systems [18][19].

Further advancements in machine learning include the use of graph-based modeling, representation learning, and dynamic dependency structures. Recent studies have demonstrated the effectiveness of transformer architectures combined with graph integration in tasks such as real-time risk monitoring [20], as well as causal representation learning frameworks to enhance model interpretability and robustness [21][22]. These techniques, though developed in different domains, provide conceptual tools for capturing complex dependencies in user-side energy behaviors.

Retrieval-augmented generation (RAG) and large language model (LLM)-based systems have also gained attention, particularly in automating decision-making and explanation generation under distributional shifts. Applications integrating LLMs with observability and monitoring systems demonstrate how generalized models can contribute to robust, explainable system behaviors [23][24].

Recent research in unsupervised learning and computer vision has introduced robust strategies for anomaly detection and multi-modal adaptation. These techniques utilize contrastive learning, feature aggregation, and test-time adaptation to improve model reliability under real-world noise and uncertainty [25][26][27]. Such robustness mechanisms align well with the goal of ensuring accurate load forecasting under variable user participation in DR.

By integrating these interdisciplinary insights-from ensemble deep learning to RL-based optimization and graph-based modeling-the proposed ensemble LSTM framework leverages smoothing, scaling, and weighted loss to offer a robust and accurate demand elasticity forecasting solution under incentive-driven DR scenarios.

3. Demand Response Under the LA Framework

The implementation process of demand response under the load aggregator (LA) framework is illustrated in Figure 1. After aggregating load resources, the LA participates in the electricity market as an independent entity and obtains demand response indicators and corresponding incentives through bidding or other market mechanisms. During the implementation of demand response, the LA allocates different incentive levels to users according to its own profit objectives and user states in order to achieve the target response.

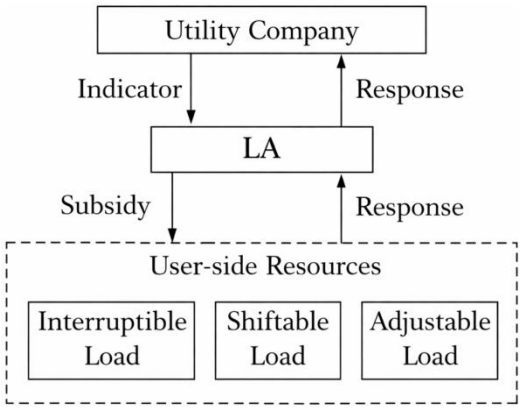

Figure 1. Demand response under the load aggregator (LA) framework

The LA framework involves three main entities, namely power utilities, LAs, and user-side resources capable of participating in demand response. In the operating mechanism, power utilities and LAs predefine electricity prices and sign contracts in advance. Based on their operational conditions, power utilities determine whether user-side participation in load regulation is required. If regulation is needed, load regulation targets are issued to the LA in advance. Since the costs associated with grid regulation vary among

different users, the LA adjusts different types of loads according to its profit objectives and provides corresponding incentives.

Traditional electricity market services impose certain requirements on deviation control. Due to inevitable discrepancies between day-ahead market schedules and actual power consumption, LAs are required to participate in real-time electricity trading or procure ancillary services to compensate for such deviations. To ensure that user-side regulation can meet the requirements of power utilities, the load regulation capacity provided by the LA must include a certain redundancy margin. Therefore, even loads that do not actually participate in regulation during the regulation period are required to undergo testing, so as to ensure that the regulation capacity provided by the LA can respond to unexpected events. Otherwise, penalties may be imposed. If the load capacity reported by the LA is insufficient, its profit will decrease; moreover, if the load regulation target specified by the power utility is not achieved, the LA will also face penalties. Consequently, accurate prediction of user response behavior is essential for the LA when integrating user-side resources to participate in electricity markets.

Throughout the incentive-based demand response process, user responses to different incentives are influenced by multiple factors. These factors mainly include the availability of alternative solutions, the current load status of users, the proportion of electricity expenditure relative to total expenditure, and external environmental conditions. For example, if non-electric equipment can be used as substitutes with comparable effectiveness, user response flexibility tends to be high. When the current load level is relatively high, the proportion of interruptible or transferable load increases, making users more likely to respond to incentives and exhibit higher demand elasticity. In addition, when electricity expenditure accounts for a small proportion of total expenditure, users may be less sensitive to incentives; conversely, sensitivity may increase when this proportion is higher. External environmental conditions also play an important role. For instance, during high-temperature periods in summer afternoons, demand elasticity tends to be relatively low. In such cases, higher incentives are required if the LA aims to reduce load by encouraging users to adjust or shut down temperature control equipment. In contrast, during nighttime periods when ambient temperatures decrease, demand elasticity may increase.

In economics, demand elasticity is used to characterize the sensitivity of users to changes in commodity prices. Similarly, demand elasticity can be employed to describe user responses to incentive prices in demand response programs. Therefore, user responses to different incentives can be transformed into economic demand elasticity, which is expressed as:

$$E = \frac{\Delta R/R}{\Delta I/I} = \frac{\Delta R}{\Delta I} \cdot \frac{I}{R}$$

where E denotes demand elasticity; R and ΔR represent the response quantity of users in demand response and its variation, respectively; I and ΔI denote the incentive received by users and the corresponding variation.

User participation in demand response inevitably leads to a certain loss of comfort or personal benefit. As the magnitude of user response increases, the impact on comfort level and personal benefit becomes more significant. Accordingly, the cost function of user participation in demand response is defined as

$$\begin{cases} U(R) = \frac{1}{2}\beta_{\text{cost}}(R+\varepsilon)^2 + \alpha_{\text{cost}}(R+\varepsilon) \\ \text{s.t. } R \le L \end{cases}$$

where β_{cost} and α_{cost} are response characteristic coefficients determined by user response characteristics, and their values vary among different users. The term ε represents response noise. Since user response noise is

relatively small and exhibits randomness, it is neglected in the analysis of demand elasticity. The parameter L denotes the current load.

By combining (1) and (2), the user demand elasticity can be derived as

$$E = \frac{\mathrm{d}R}{\mathrm{d}U} \cdot \frac{U}{R} = \frac{\beta_{\mathrm{cost}} R + 2\alpha_{\mathrm{cost}}}{2\beta_{\mathrm{cost}} R + 2\alpha_{\mathrm{cost}}}$$

From the above analysis, it can be observed that user demand elasticity exhibits a quadratic functional relationship with user response behavior. However, this relationship is valid only under ideal conditions. In practical scenarios, user response characteristics fluctuate due to real-world uncertainties. Therefore, simple quadratic functions cannot accurately describe user response behavior, and more accurate methods are required to predict user responses.

4. Demand Elasticity Forecasting Based on an Ensemble LSTM Model

4.1 LSTM Model

Recurrent neural networks (RNNs) possess the capability of retaining historical information over a period of time, which makes them suitable for ultra-short-term load forecasting. However, conventional RNNs suffer from long-term dependency issues when processing long input sequences. To address this limitation, the long short-term memory (LSTM) network was proposed in as an improved variant of RNNs. Although both RNNs and LSTM networks adopt chain-like structures, their recurrent modules differ significantly. In a standard RNN, the recurrent unit consists of a single neural network layer, whereas the LSTM architecture introduces interactions among multiple internal layers, enabling more effective information control.

For LSTM-based forecasting models, prediction accuracy generally improves as the relevance between input variables and the target output increases. In practical applications, however, some influential factors are difficult to collect or cannot be continuously structured, such as real-time temperature, humidity, and wind speed. In addition, certain data sources incur high acquisition costs, for example, real-time power consumption data of individual electrical appliances, including air conditioners, electric water heaters, and lighting devices, which require user authorization and the installation of dedicated data acquisition and communication equipment. Although user electricity consumption behavior is influenced by diverse external factors, it typically exhibits strong periodic characteristics. Therefore, to enable the model to identify historical patterns on similar days, daily maximum load, daily minimum load, and time information are selected as input variables.

Moreover, the current electricity consumption behavior of users determines their maximum response potential. Consequently, the current load level of users is also incorporated as an input feature. Considering the incentive-based demand response mechanism, the incentive received by users is included as an additional input variable. In summary, the input vector of the LSTM-based model consists of daily maximum load, daily minimum load, time index, current load, and incentive signal. Figure 2 illustrates the typical structure of an LSTM network.

The LSTM network selectively retains or discards information through three gate units, namely the input gate, the forget gate, and the output gate. The forget gate determines which historical information should be preserved, the input gate controls the incorporation of new information, and the output gate regulates the information transmitted to the next time step. Through the coordinated operation of these gate mechanisms, the LSTM cell maintains a memory state that integrates past information with current inputs, thereby effectively capturing temporal dependencies in sequential data.

By updating the cell state based on both historical memory and current inputs, the LSTM network generates the hidden state output corresponding to the expected user response. This output reflects the predicted demand elasticity or load response at the current time step, providing a foundation for subsequent ensemble modeling and forecasting under incentive-based demand response scenarios.

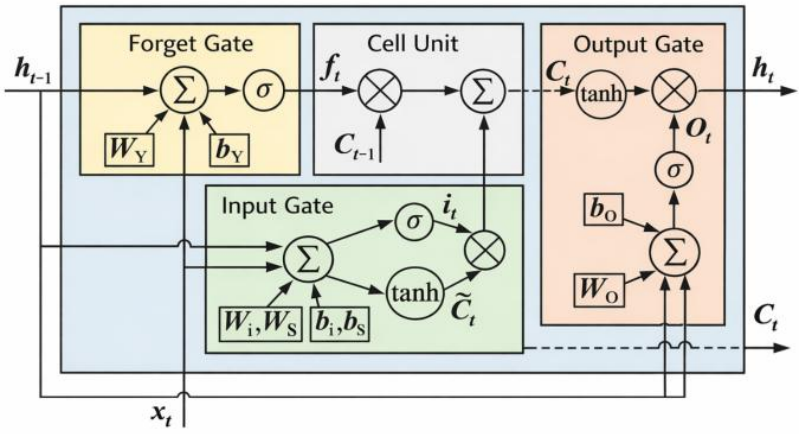

Figure 2. Structure of the LSTM network

4.2 Ensemble LSTM Forecasting Method

The forecasting model is required to support online deployment and continuously generate load predictions. Therefore, it must be capable of effectively generalizing to newly observed data and remain robust to input noise. To this end, an ensemble LSTM forecasting framework is proposed, and its overall structure is illustrated in Figure 3.

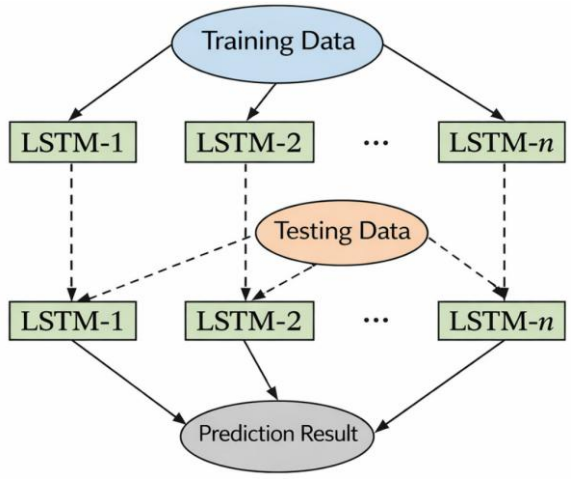

Figure 3. Structure of the ensemble LSTM model

In the proposed ensemble framework, multiple LSTM models with different random initializations are independently trained. During the prediction stage, the outputs of all individual LSTM models are averaged to produce the final forecasting result. This ensemble strategy effectively enhances the robustness and generalization capability of the forecasting model.

To further improve resistance to input disturbances, Gaussian noise is injected into the hidden states between time steps of each LSTM layer during the training process. In the inference stage, noise injection is repeatedly applied to the trained model for evaluation, and the outputs obtained from multiple stochastic forward passes are averaged. This procedure enables the model to better capture uncertainty in the input data and mitigates the impact of noise on prediction accuracy.

4.3 Ensemble LSTM-Based Load Forecasting Procedure

LSTM networks possess memory capabilities that enable them to capture temporal dependencies in load time series data. However, when applying ensemble LSTM models to load forecasting, the large volume and irregularity of raw data, together with the increased number of model parameters, make training more challenging. Consequently, appropriate data preprocessing and parameter configuration are required prior to model training. The detailed load forecasting procedure based on the ensemble LSTM model is described as follows.

1) Data Preprocessing

To enable the model to more effectively capture local features and dominant trends in load data, smoothing is applied to the original time series. Each data point is replaced by the average of a fixed number of consecutive historical observations, including the current value. This operation reduces short-term fluctuations and highlights underlying trends.

Since different input features may exhibit diverse scales and distributions, data scaling is performed prior to model training. All input variables are transformed into a predefined numerical range using min-max normalization. This process accelerates model convergence and improves training stability.

2) Parameter Initialization

All variables in the LSTM network, including weights and bias terms, are initialized prior to training. Weights are initialized using a normal distribution with zero mean and unit variance, while all bias terms are set to small constant values [19]-[20]. Proper initialization helps prevent gradient instability and facilitates effective model training.

3) LSTM Model Training

The ensemble LSTM model is trained using gradient-based optimization methods. Network parameters are updated iteratively in the direction that minimizes the loss function. To reduce training time and improve optimization efficiency, stochastic gradient descent is adopted, where a subset of training samples is randomly selected in each iteration.

Although stochastic optimization accelerates convergence, it may not always guarantee convergence to the global minimum. To balance efficiency and accuracy, a mini-batch training strategy is employed. In each iteration, the loss function is computed over a batch of training samples, which reduces variance in parameter updates while maintaining computational efficiency.

To further improve forecasting performance, a weighted loss function is introduced. By assigning different weights to prediction errors, the influence of large absolute errors on model training is reduced, thereby improving overall prediction accuracy and robustness.

4) Load Forecasting

After sufficient training iterations, the loss function converges to a relatively low level, and the trained ensemble LSTM model is saved. During the forecasting stage, input data are first standardized using the same preprocessing procedures as in the training phase. The trained model is then used to predict user responses and load demand under incentive-based demand response scenarios.

4.4 Evaluation Metric

Considering both statistical error analysis and practical operational requirements, the mean absolute percentage error (MAPE) is adopted as the evaluation metric for the proposed forecasting model. MAPE provides an intuitive measure of relative prediction accuracy and is widely used in load forecasting applications.

5. Case Study and Analysis

5.1 Case Study Data

To verify the effectiveness and reliability of the proposed method, load data provided by a State Grid company are employed. The dataset consists of load measurements from five large industrial plants over a continuous period of 40 days within the same season, with a sampling interval of 15 min. The load data from the first 30 days are used as the training set, while the remaining 10 days are reserved for testing.

Since all load data are collected within the same season, daily incentive policies are generally consistent. Specifically, incentive policies are applied from 00:00 to 20:00 each day, while no incentives are provided from 21:00 to 24:00, as illustrated in Figure 4. To reflect stochastic noise in user response behavior, the random noise term ε, as well as the response characteristic parameters βcost and αcost, are assumed to follow normal distributions. The specific parameter settings are summarized in Table 1.

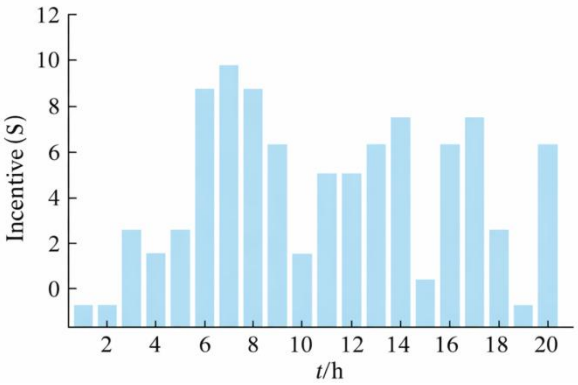

Figure 4. Incentive policies at different time periods

Table 1: Parameters of Demand Response

Parameter	00:00-06:00	07:00-12:00	13:00-18:00	19:00-20:00
Mean value of α(_{cost})	1.5	3	1	1.7
Mean value of β(_{cost})	6	5	6	4.2
Standard deviation	0.2	0.2	0.2	0.2

5.2 Result Analysis

To evaluate the performance of the proposed ensemble LSTM-based forecasting method, comparative experiments are conducted with a conventional LSTM model and the k-nearest neighbor (kNN) forecasting method. The impact of data preprocessing techniques, including smoothing, scaling, and weighted loss functions, on prediction performance is also analyzed.

5) Comparison of Demand Elasticity and Total Load Forecasting Results

First, the demand elasticity forecasting performance of different methods is compared. Based on the incentive policy shown in Figure 4 and the demand response parameters listed in Table 1, the forecasting results of the ensemble LSTM, conventional LSTM, and kNN methods are presented in Table 2 and Figure 5-7. Since no incentive policy is applied from 21:00 to 24:00, demand elasticity forecasting is unnecessary during this period; therefore, only the results from 00:00 to 20:00 are shown.

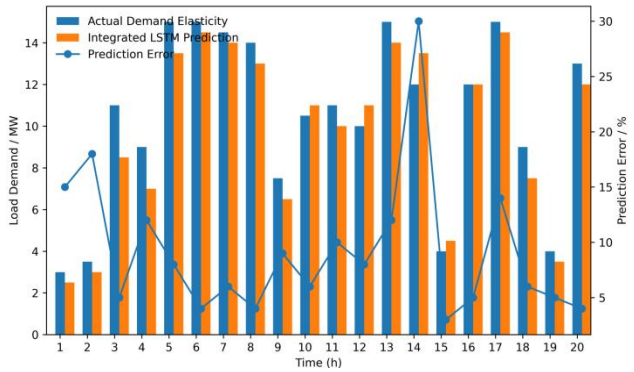

Figure 5. Demand elasticity prediction based on the ensemble LSTM method

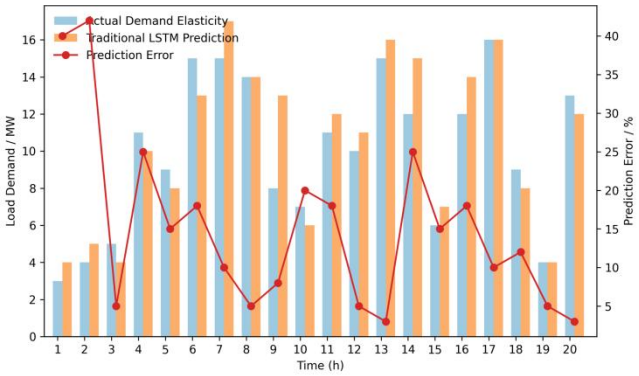

Figure 6. Demand elasticity prediction based on the conventional LSTM method

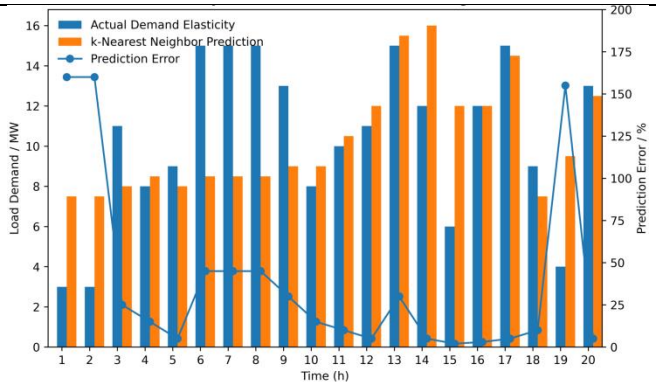

Figure 7. Demand elasticity prediction based on the k-nearest neighbor (kNN) method

Table 2: Comparison of Demand Elasticity Prediction Results

Method	Average Error (%)
Ensemble LSTM	10.14
Conventional LSTM	15.47
k-Nearest Neighbor (kNN)	38.94

As indicated in Table 2, significant differences exist among the average forecasting errors of the three methods. The ensemble LSTM method achieves the lowest average error of 10.14%, which is 28.8% lower than that of the kNN method and 5.33% lower than that of the conventional LSTM method. Figure 5-7 further illustrate the demand elasticity forecasting errors at different time periods. The maximum prediction error of the kNN method exceeds 150%, while the maximum error of the conventional LSTM method is approximately 50%. In contrast, the maximum prediction error of the ensemble LSTM method does not exceed 40%. These results demonstrate that the proposed ensemble LSTM method provides superior accuracy and stability in demand elasticity forecasting.

In addition, Figure 4 shows that demand elasticity exhibits a trend consistent with the incentive policy. When incentives are higher, users provide greater load regulation, whereas lower incentives correspond to smaller load adjustments.

Next, total load forecasting performance is evaluated to further demonstrate the superiority of the ensemble LSTM method. The ensemble LSTM, conventional LSTM, and kNN methods are applied to total load forecasting, and the results are compared in Table 3 and Figure 8 and 9. Since total load forecasting requires predictions over a full 24-hour horizon, the results from 00:00 to 24:00 are presented.

Table 3: Comparison of Total Load Prediction Results

Method	MAPE (%)
Ensemble LSTM	1.08
Conventional LSTM	3.14
k-Nearest Neighbor (kNN)	4.17

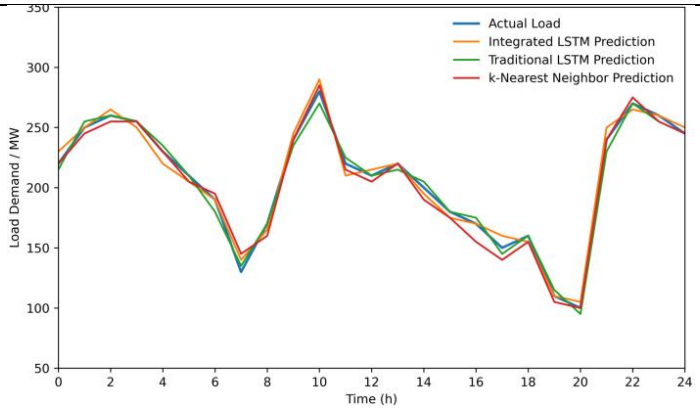

Figure 8. Comparison between predicted load and actual load.

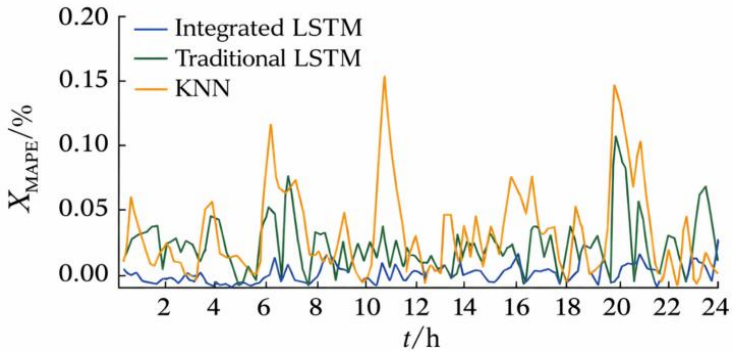

Figure 9. Comparison of MAPE values

A horizontal comparison of forecasting results shows that the ensemble LSTM method achieves a mean absolute percentage error (MAPE) of 1.08%, which is 2.06% lower than that of the conventional LSTM method and 3.09% lower than that of the kNN method. These results indicate that the ensemble LSTM method outperforms both benchmark methods in total load forecasting accuracy.

As shown in Figure 8, the load forecasting results of the ensemble LSTM method are closer to the actual load values compared with those of the conventional LSTM and kNN methods. Figure 9 further confirms that the ensemble LSTM method produces the smallest forecasting error, while the kNN method exhibits the largest error. This is because the kNN method relies on the nearest historical data points for classification and prediction; once noisy or erroneous data exist, its accuracy degrades significantly. In contrast, the LSTM model propagates state information throughout the network structure, ensuring information consistency and improving forecasting accuracy. By training multiple LSTM models with different initializations and averaging their outputs, the ensemble LSTM method further enhances prediction accuracy.

Finally, Table 4 compares the computational time required by different forecasting methods. Although the ensemble LSTM method achieves higher prediction accuracy, it requires longer computation time than the conventional LSTM and kNN methods. This is mainly due to the need to train multiple LSTM models with different initial values in parallel. Even with batch processing strategies, the increased computational complexity results in higher time consumption.

Table 4: Comparison of Computational Time for Different Prediction Methods

Method	Time (min)

Ensemble LSTM	30
Conventional LSTM	26
k-Nearest Neighbor (kNN)	14

6) Impact of Data Preprocessing on Forecasting Performance

The superiority of the proposed ensemble LSTM forecasting method has been demonstrated in the previous subsection. In this subsection, the effects of data preprocessing techniques, including smoothing, scaling, and the introduction of a weighted loss function, on forecasting performance are further investigated. To eliminate the influence of sample randomness on the experimental results, two comparative case studies are constructed.

In Case I, the experimental setting follows the configuration described previously, where the load data from the first 30 days are used for training and the remaining 10 days are used for testing. In Case II, the last 30 days of load data are selected as the training set, while the first 10 days are used as the testing set. By comparing the results obtained from these two cases, the impact of data preprocessing on forecasting accuracy can be evaluated while excluding the randomness introduced by sample selection.

Table 5 presents a comparison of forecasting accuracy under different data preprocessing strategies in both Case I and Case II. The results, measured in terms of MAPE, show consistent trends across the two cases, indicating that the effects of data preprocessing on prediction accuracy are stable and not caused by random sample selection. This consistency confirms that the observed improvements in forecasting performance are attributable to the preprocessing methods rather than random variations in the test samples.

Table 5: Comparison of Prediction Results Under Different Data Preprocessing Methods

Preprocessing Method	Case I (MAPE %)	Case II (MAPE %)
Without preprocessing	3.14	3.36
Smoothing	2.65	2.59
Scaling	2.06	2.37
Scaling + weighted loss	1.7	1.88
Smoothing + scaling + weighted loss	1.08	0.94

Figure 10 illustrates the impact of smoothing, scaling, and weighted loss functions on forecasting accuracy in Case I. It can be observed that, based on the conventional LSTM forecasting model, the application of smoothing and scaling preprocessing leads to a reduction in prediction error. When scaling and weighted loss functions are applied simultaneously, the forecasting error is further reduced. Moreover, the proposed method, which integrates smoothing, scaling, and weighted loss functions in the data preprocessing stage, achieves the lowest prediction error among all compared approaches. These results demonstrate that appropriate data preprocessing can effectively enhance forecasting accuracy.

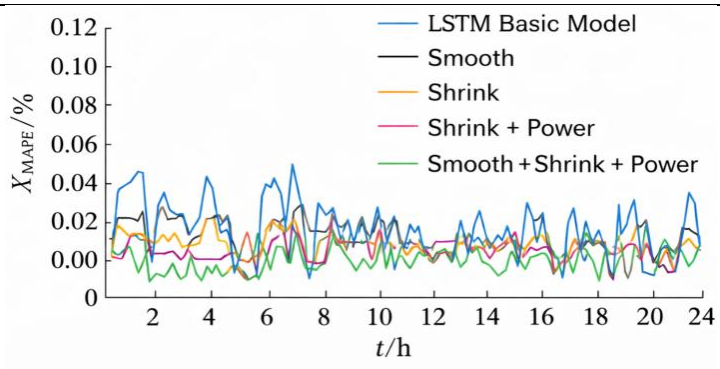

Figure 10. Impact of data preprocessing on forecasting accuracy

6. Conclusion

This paper proposed a data-driven demand elasticity forecasting approach based on an ensemble LSTM model. The main conclusions can be summarized as follows.

Compared with the conventional LSTM model and the k-nearest neighbor (kNN) forecasting method, the proposed ensemble LSTM approach effectively improves the forecasting accuracy of both demand elasticity and total load. Specifically, the average prediction error of demand elasticity is reduced by 5.33% and 28.8%, respectively, while the total load forecasting error is reduced by 2.06% and 3.09%, respectively, demonstrating the superior predictive performance of the proposed method.

Although the ensemble LSTM method achieves a significant improvement in forecasting accuracy, it requires longer computational time than the conventional LSTM and kNN methods. This is mainly because multiple LSTM models with different initial values must be trained in parallel. Even with batch training strategies, the increased computational complexity leads to higher time consumption.

Due to the differences in scale and distribution of the original data, preprocessing techniques such as smoothing and scaling play an important role in improving forecasting performance. The results show that appropriate preprocessing of input data has a positive impact on the accuracy of demand elasticity and load forecasting.

With the large-scale integration of distributed photovoltaic and wind power generation on the user side, users are no longer only electricity consumers but can also act as electricity suppliers. Future work will focus on further investigation of user behavior patterns by incorporating auxiliary service markets and government subsidy policies for renewable energy, in order to achieve more accurate and comprehensive demand response modeling.

References

[1] Zhang J., Peng Y., Wu H., et al. Reliability and risk analysis of load aggregators in demand response programs[J]. Applied Energy, 2019, 252: 113432.

[2] Chen J., Yang P., Chen Y., et al. Optimal operation of multi-energy systems in industrial parks considering integrated demand response[J]. Renewable Energy, 2021, 172: 1061-1074.

[3] H. Ren, H. Lu, J. Lu and Z. Jing, "Analysis of LA demand response characteristics considering cyber physical system coupling and user’ s response difference," Power Syst. Technol., vol. 44, no. 10, pp. 3927-3936, 2020.

[4] A. Zhou, K. Ren, X. Li and W. Zhang, "MMSE: A multi-model stacking ensemble learning algorithm for purchase prediction," 2019 IEEE 8th Joint International Information Technology and Artificial Intelligence Conference (ITAIC), pp. 96-102, May 2019.

[5] V. Kulkarni, S. K. Sahoo, S. B. Thanikanti, S. Velpula and D. I. Rathod, "Power systems automation, communication, and information technologies for smart grid: A technical aspects review," TELKOMNIKA (Telecommunication Computing Electronics and Control), vol. 19, no. 3, pp. 1017-1029, 2021.

[6] W. Liao, P. Li, Q. Wu, S. Huang, G. Wu and F. Rong, "Distributed optimal active and reactive power control for wind farms based on ADMM," International Journal of Electrical Power & Energy Systems, vol. 129, p. 106799, 2021.

[7] K. Gillingham, D. Rapson and G. Wagner, "The rebound effect and energy efficiency policy," Review of Environmental Economics and Policy, 2016.

[8] S. Sharma, A. Majumdar, V. Elvira and E. Chouzenoux, "Blind Kalman filtering for short-term load forecasting," IEEE Transactions on Power Systems, vol. 35, no. 6, pp. 4916-4919, 2020.

[9] X. Kong, C. Li, F. Zheng and C. Wang, "Improved deep belief network for short-term load forecasting considering demand-side management," IEEE Transactions on Power Systems, vol. 35, no. 2, pp. 1531-1538, 2019.

[10] 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 Transactions on Smart Grid, vol. 10, no. 1, pp. 841-851, 2017.

[11] M. H. Albadi and E. F. El-Saadany, "A summary of demand response in electricity markets," Electric Power Systems Research, vol. 78, no. 11, pp. 1989-1996, 2008.

[12] C. L. Su and D. Kirschen, "Quantifying the effect of demand response on electricity markets," IEEE Transactions on Power Systems, vol. 24, no. 3, pp. 1199-1207, 2009.

[13] J. W. Taylor, "Triple seasonal methods for short-term electricity demand forecasting," European Journal of Operational Research, vol. 204, no. 1, pp. 139-152, 2010.

[14] S. A. Nabavi, S. Mohammadi, N. H. Motlagh, S. Tarkoma and P. Geyer, "Deep learning modeling in electricity load forecasting: Improved accuracy by combining DWT and LSTM," Energy Reports, vol. 12, pp. 2873-2900, 2024.

[15] A. Baliyan, K. Gaurav and S. K. Mishra, "A review of short term load forecasting using artificial neural network models," Procedia Computer Science, vol. 48, pp. 121-125, 2015.

[16] T. Gupta, R. Bhatia and S. Sharma, "An ensemble-based enhanced short and medium term load forecasting using optimized missing value imputation," Scientific Reports, vol. 15, no. 1, p. 21857, 2025.

[17] Y. Ou, S. Huang, R. Yan, K. Zhou, Y. Shu and Y. Huang, "A Residual-Regulated Machine Learning Method for Non-Stationary Time Series Forecasting Using Second-Order Differencing," unpublished.

[18] G. Yao, H. Liu and L. Dai, "Multi-agent reinforcement learning for adaptive resource orchestration in cloud-native clusters," Proceedings of the 2025 2nd International Conference on Intelligent Computing and Data Analysis, pp. 680-687, Aug. 2025.

[19] Y. Zou, N. Qi, Y. Deng, Z. Xue, M. Gong and W. Zhang, "Autonomous resource management in microservice systems via reinforcement learning," 2025 8th International Conference on Computer Information Science and Application Technology (CISAT), pp. 991-995, Jul. 2025.

[20] Y. Wu, Y. Qin, X. Su and Y. Lin, "Transformer-based risk monitoring for anti-money laundering with transaction graph integration," Proceedings of the 2025 2nd International Conference on Digital Economy, Blockchain and Artificial Intelligence, pp. 388-393, Jun. 2025.

[21] J. Li, Q. Gan, R. Wu, C. Chen, R. Fang and J. Lai, "Causal Representation Learning for Robust and Interpretable Audit Risk Identification in Financial Systems," unpublished.

[22]C. F. Chiang, D. Li, R. Ying, Y. Wang, Q. Gan and J. Li, "Deep Learning-Based Dynamic Graph Framework for Robust Corporate Financial Health Risk Prediction," unpublished.

[23]S. Sun, "CIRR: Causal-Invariant Retrieval-Augmented Recommendation with Faithful Explanations under Distribution Shift," arXiv preprint arXiv:2512.18683, 2025.

[24]C. Wang, T. Yuan, C. Hua, L. Chang, X. Yang and Z. Qiu, "Integrating Large Language Models with Cloud-Native Observability for Automated Root Cause Analysis and Remediation," unpublished.

[25]Z. Zhang, W. Liu, J. Tao, H. Zhu, S. Li and Y. Xiao, "Unsupervised Anomaly Detection in Cloud-Native Microservices via Cross-Service Temporal Contrastive Learning," unpublished.

[26]S. Peng, X. Zhang, L. Zhou and P. Wang, "YOLO-CBD: Classroom Behavior Detection Method Based on Behavior Feature Extraction and Aggregation," Sensors, vol. 25, no. 10, p. 3073, 2025.

[27]J. Xie, Y. Wu, Y. Zhang, X. Zhang, Y. Xie and Y. Qu, "PLATO-TTA: Prototype-Guided Pseudo-Labeling and Adaptive Tuning for Multi-Modal Test-Time Adaptation of 3D Segmentation," Proceedings of the 33rd ACM International Conference on Multimedia, pp. 2226-2234, Oct. 2025.