The Journal of Supercomputing (2025) 81:1108
https://doi.org/10.1007/s11227-025-07564-5

# Short-term electricity consumption forecasting with deep learning

Emrah Demir[1] · Serkan Gunal[1]

Accepted: 8 June 2025

**Abstract**
Global electricity demand is surging due to population growth, industrialization, and technological advancements. While renewable energy sources are expanding, fossil fuels still remain the primary source of electricity generation, posing challenges due to resource limitations and environmental concerns. To address these challenges and optimize energy use, accurate prediction of electricity consumption is crucial. Therefore, this work introduces novel short-term (24-hour) electricity consumption forecasting models based on customized long short-term memory (LSTM) networks, convolutional neural networks (CNNs), and their ensemble. The models utilize time-series electricity consumption data and meteorological features, including temperature, relative humidity, and wind speed. Trained and evaluated on two geographically distinct datasets spanning 2.5 years, our models utilizing appropriate feature sets surpass the recent studies and achieve significantly high forecasting performance with normalized root mean square error (N-RMSE) reaching 0.16, normalized mean absolute error (N-MAE) reaching 0.13, and mean absolute percentage error reaching 4%. The inclusion of meteorological features contributed notably to prediction performance, demonstrating the benefit of integrating external features in electricity forecasting models. The results highlight the effectiveness of customized deep learning architectures in capturing complex temporal and contextual dependencies within electricity consumption data. Also, these findings offer valuable insights for future research and practical applications in energy management and grid optimization.

**Keywords** Electricity consumption forecasting · Energy management · LSTM · CNN · Ensemble learning · Deep learning

✉ Emrah Demir
emrahdemir@eskisehir.edu.tr

Serkan Gunal
serkangunal@eskisehir.edu.tr

[1] Department of Computer Engineering, Eskisehir Technical University, Eskisehir, Türkiye

Published online: 03 July 2025

## 1 Introduction

With a growing population, ensuring a stable electricity supply is increasingly critical. Electricity powers homes, industries, transportation, healthcare, and cultural institutions. As demand rises, responsible and sustainable energy use is essential. Various sources like wind, water, solar, natural gas, coal, oil, and nuclear contribute to generation, but high storage costs and operational complexities require balancing supply and demand [1]. Efficient energy management relies on accurately predicting electricity needs to prevent overproduction and underproduction, ensuring sustainable resource use.

Insufficient electricity production has severe consequences [2, 3]. Supply shortfalls can cause power outages, disrupting daily life, halting commercial and industrial activities, leading to economic losses and job cuts. It also affects medical equipment, security systems, education, communication, supply chains, and social stability.

On the other hand, excessive electricity production can harm the environment and economy. Overreliance on fossil fuels increases greenhouse gas emissions, while excessive hydroelectric and nuclear production strains water resources and creates nuclear waste challenges. Economically, overproduction lowers energy prices, reduces revenues for energy companies, discourages investment, and stresses transmission and distribution infrastructure, necessitating unnecessary market investments.

The rising demand for electrical energy, especially in developing societies, underscores the need for sustainable solutions. The International Energy Agency (IEA), established in 1974, provides data and analysis to help governments and industries enhance energy security, efficiency, and transition to renewables. In 2021, the IEA reported that the industrial sector was the largest electricity consumer, though its share decreased from 53.5% in 1973 to 41.9% in 2019, while residential, commercial, and public sectors grew.[1] Accurate forecasting of electricity production and consumption is therefore essential given the rising demand and its importance.

Time series forecasting models predict future values based on historical data, playing a crucial role in optimizing electricity usage and mitigating adverse effects. Forecasting horizons are typically categorized as short-term (minutes to days), medium-term (days to months), and long-term (extended periods) [4]. Numerous models have been proposed to address the inherent challenges of time series forecasting. Initial efforts were based on statistical techniques such as autoregressive (AR), autoregressive moving average (ARMA) [5], and autoregressive integrated moving average (ARIMA) [6], which are effective for modeling linear relationships but struggle with complex nonlinear patterns and long-range dependencies.

Recent advancements have shifted toward machine learning and deep learning techniques, which offer greater flexibility in modeling nonlinear dynamics. For instance, support vector machines (SVM) [7], although primarily used for

[1] https://www.iea.org/reports/key-world-energy-statistics-2021/final-consumption.

classification tasks, have been adapted for regression-based forecasting. Similarly, exponential smoothing (ES) [8], which adjusts predictions based on recent data, has found utility in time series analysis. Among deep learning approaches, artificial neural networks (ANNs)—including multilayer perceptrons (MLPs), recurrent neural networks (RNNs), and long short-term memory (LSTM) networks, have gained popularity for their ability to capture nonlinear and temporal dependencies in complex datasets [9].

Ensemble learning, which combines the strengths of multiple models, has also proven effective in forecasting tasks. For example, a robust ensemble-based hybrid model was proposed in [10], integrating five distinct deep learning architectures using an adaptive weighting mechanism to enhance the accuracy and reliability of energy consumption forecasts in China. In parallel, advancements in standalone deep learning architectures have yielded powerful models such as the neural basis expansion analysis for time series (N-BEATS) [11], the deep autoregressive (DeepAR) [12], and the temporal fusion transformer (TFT) [13], all of which have demonstrated high performance in various forecasting applications. Furthermore, architectures involving convolutional neural networks (CNNs) and multi-head attention (MHA) mechanisms have been successfully applied to electricity demand prediction, showcasing their ability to model spatial and sequential features effectively.

Hybrid models that integrate various algorithmic techniques—such as combining ARIMA with LSTM or CNN with LSTM—have demonstrated superior forecasting performance compared to traditional baseline models [14–16]. Building on this trend, several recent studies have proposed innovative hybrid frameworks tailored to specific application domains. For instance, one study enhanced energy efficiency by developing a hybrid renewable energy system that significantly reduced operational costs while improving performance compared to conventional methods [17]. Continuing this research direction, another approach introduced a hybrid forecasting model that combines the Prophet time-series framework with extreme gradient boosting (XGBoost), further refined using Bayesian optimization techniques [18]. Prophet—developed by Facebook—is a univariate deep learning-inspired time-series model that is particularly effective at modeling complex trends and seasonality, even in the presence of missing or irregular data. Meanwhile, XGBoost is a tree-based ensemble learning algorithm known for its robustness and efficiency in capturing nonlinear interactions within structured datasets.

Beyond hybrid modeling, research has increasingly explored emerging paradigms, such as transfer learning, generative modeling, and causal inference. Transfer learning involves adapting pretrained models to new datasets with minimal retraining effort, thereby improving generalization and reducing the need for large labeled datasets [19]. Generative models, including variational autoencoders [20] and generative adversarial networks [21], are used to produce synthetic training data, which can be particularly beneficial in low-resource settings. Meanwhile, causal models aim to uncover cause-effect relationships between variables, offering improved interpretability and actionable insights for decision-making in forecasting.

In parallel with these developments, a limited number of emerging approaches have begun leveraging large language models (LLMs) for forecasting tasks. One such approach, known as BlackInter, presents a memory-efficient framework for building-level electricity load prediction under limited data conditions [22]. By integrating pretrained LLMs with spatiotemporal embeddings and dynamic mode decomposition, the model effectively captures dominant load patterns. It achieves competitive performance in both few-shot and zero-shot settings without requiring access to internal LLM parameters.

Despite the progress made in electricity consumption forecasting, challenges remain in accurately predicting short-term fluctuations. Existing models often struggle to capture the complex nonlinear relationships and dynamic patterns present in electricity consumption data. To address these limitations, this study presents novel short-term (24-h) electricity consumption forecasting models using deep learning—specifically customized LSTM, CNN, and ensemble architectures—integrated with meteorological features to enhance prediction accuracy and robustness. We conduct a systematic evaluation on two datasets containing electricity consumption, meteorological data, and time information from two geographically diverse cities. The models employ a multi-step approach, predicting hourly consumption for the next day based on the past seven days (168 h) of data. Various performance metrics, including normalized root mean square error (N-RMSE), normalized mean absolute error (N-MAE), and mean absolute percentage error (MAPE), assess the models' predictive capabilities. Our work offers the following key contributions:

- Developed innovative forecasting models for short-term electricity consumption.
- Conducted a comprehensive analysis of meteorological and timestamp factors' impact on forecasting accuracy.
- Provided a valuable resource for researchers and practitioners applying deep learning to time series forecasting.
- Enabled more accurate predictions for improved decision-making and resource allocation within the energy sector.

The remainder of the article is organized as follows: Sect. 2 details the proposed forecasting models, including the datasets, performance metrics, and model architectures. Section 3 analyzes the experimental results, evaluating the models' performance. Finally, Section 4 concludes the study and suggests directions for future research.

## 2 Proposed models

In this study, we propose novel multivariate forecasting models based on customized LSTM and CNN architectures, as well as their ensemble. The models are also part of the first author's M.Sc. thesis [23]. Figure 1 depicts the entire forecasting framework, which consists of three hierarchical levels: (i) data collection, preprocessing, and feature engineering, (ii) model development, and (iii) performance evaluation and visualization. At the first level, electricity consumption and meteorological data from two regions are collected. Feature engineering and preprocessing generate 16 distinct feature sets, which are used for training and testing the models. The second level, Model Development, involves designing and implementing custom LSTM and

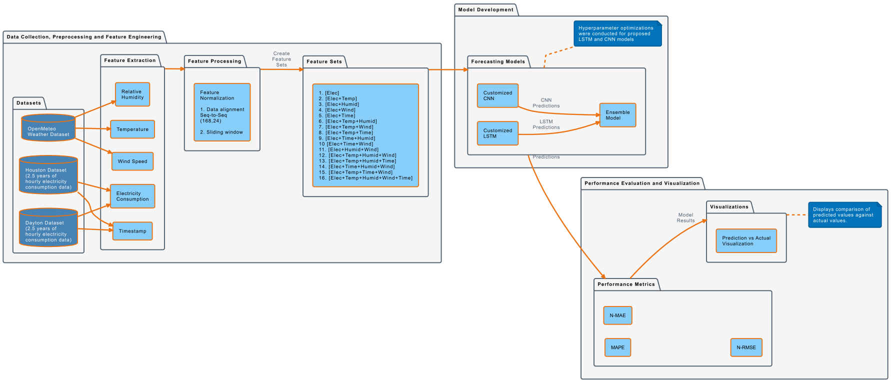

Data Collection, Preprocessing and Feature Engineering
Datasets
OpenMeteo Weather Dataset
Houston Dataset (2.5 years of hourly electricity consumption data)
Dayton Dataset (2.5 years of hourly electricity consumption data)
Feature Extraction
Relative Humidity
Temperature
Wind Speed
Electricity Consumption
Timestamp
Feature Processing
Feature Normalization
1. Data alignment Seq-to-Seq (168,24)
2. Sliding window
Create Feature Sets
Feature Sets
1. [Elec]
2. [Elec+Temp]
3. [Elec+Humid]
4. [Elec+Wind]
5. [Elec+Time]
6. [Elec+Temp+Humid]
7. [Elec+Temp+Wind]
8. [Elec+Temp+Time]
9. [Elec+Time+Humid]
10 [Elec+Time+Wind]
11. [Elec+Humid+Wind]
12. [Elec+Temp+Humid+Wind]
13. [Elec+Temp+Humid+Time]
14. [Elec+Time+Humid+Wind]
15. [Elec+Temp+Time+Wind]
16. [Elec+Temp+Humid+Wind+Time]
Model Development
Hyperparameter optimizations were conducted for proposed LSTM and CNN models
Forecasting Models
Customized CNN
CNN Predictions
Customized LSTM
LSTM Predictions
Ensemble Model
Predictions
Performance Evaluation and Visualization
Visualizations
Model Results
Prediction vs Actual Visualization
Displays comparison of predicted values against actual values.
Performance Metrics
N-MAE
MAPE
N-RMSE

**Fig. 1** The proposed forecasting framework

CNN models, which are optimized through hyperparameter tuning. An ensemble model is then created by combining the LSTM and CNN predictions to enhance forecasting performance. The final level, Performance Evaluation and Visualization, assesses the models' predictive accuracy and presents the results through visualizations for clear and interpretable insights.

In the following subsections, the datasets used for training and evaluating the models are introduced, the performance metrics employed to assess the models are described, and the architectures of the models are explained in detail.

## 2.1 Datasets

The first dataset comprises hourly electricity consumption data for Dayton, Ohio, sourced from Kaggle and PJM Interconnection.[2] The full dataset contains 120,840 h (5,035 days) of records. However, to ensure consistency in experimental design and allow direct comparison between the two case studies, a subset of 20,424 h (851 days) was selected. This period, spanning from 04/01/2016 to 03/05/2018, was deliberately matched to the size and duration of the Houston dataset. The selection ensures that both datasets have the same temporal resolution and length, minimizing bias and promoting fair model comparison across locations. In addition to electricity consumption, meteorological variables such as temperature, relative humidity, and wind speed for Dayton were obtained from Open-Meteo using the coordinates of the city.[3]

Similarly, the second dataset provides an equivalent duration of 20,424 h (851 days) of hourly electricity consumption data for Houston, Texas, sourced from Kaggle,[4] spanning from 01/01/2018 to 30/04/2020. To maintain uniformity in data

[2] https://www.kaggle.com/datasets/robikscube/hourly-energy-consumption.

[3] https://open-meteo.com/en/docs/historical-weather-api.

[4] https://www.kaggle.com/datasets/shubhamkulkarni01/us-top-10-cities-electricity-and-weather-data.

**Table 1** Distribution of training, validation, and testing data for Dayton and Houston datasets

| Data type | Hours |
|---|---|
| Training | 9144 |
| Validation | 2352 |
| Test | 8928 |
| Total | 20424 |

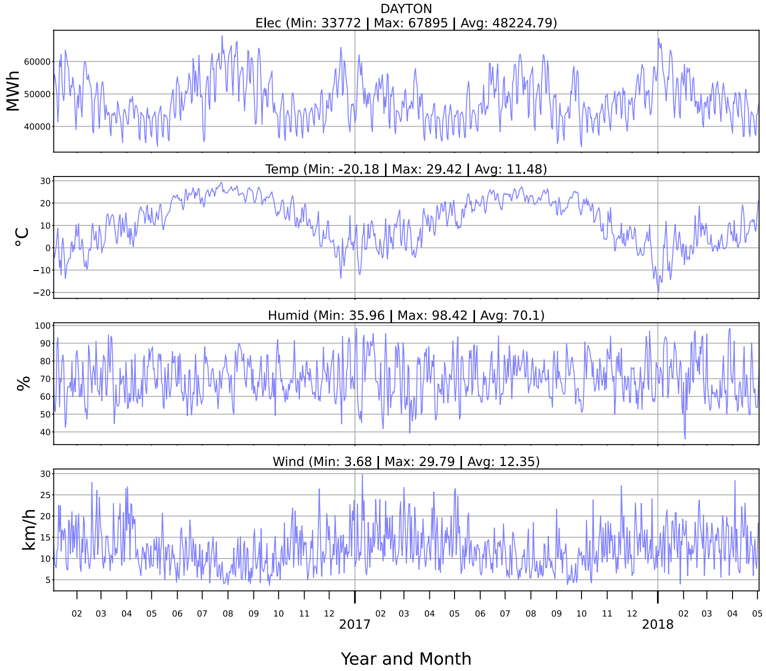


**Fig. 2** Dayton dataset: daily fluctuations in electricity consumption and meteorological data

features, meteorological parameters for Houston were also collected from Open-Meteo using the coordinates of the city.[5]

The dataset design also aimed to capture seasonal variability and daily operational patterns. The training and validation sets were defined to cover approximately 16 months of data, enabling the models to learn distinct seasonal and weekday effects. A dedicated test set of approximately one year was reserved to independently evaluate model generalization under previously unseen seasonal and day-to-day variations. Both electricity consumption and meteorological data were carefully selected to meet this objective and provide a realistic and challenging forecasting scenario. Table 1 shows the distribution of data splits for both datasets.

[5] https://open-meteo.com/en/docs/historical-weather-api.

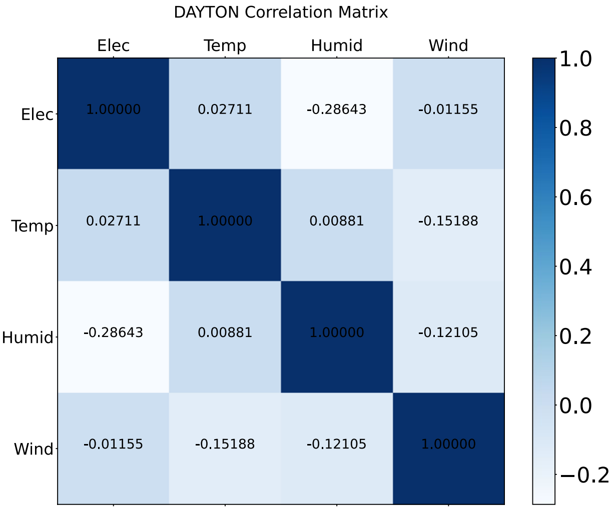


**Fig. 3** Correlation matrix showing relationships between electricity consumption and meteorological variables in Dayton

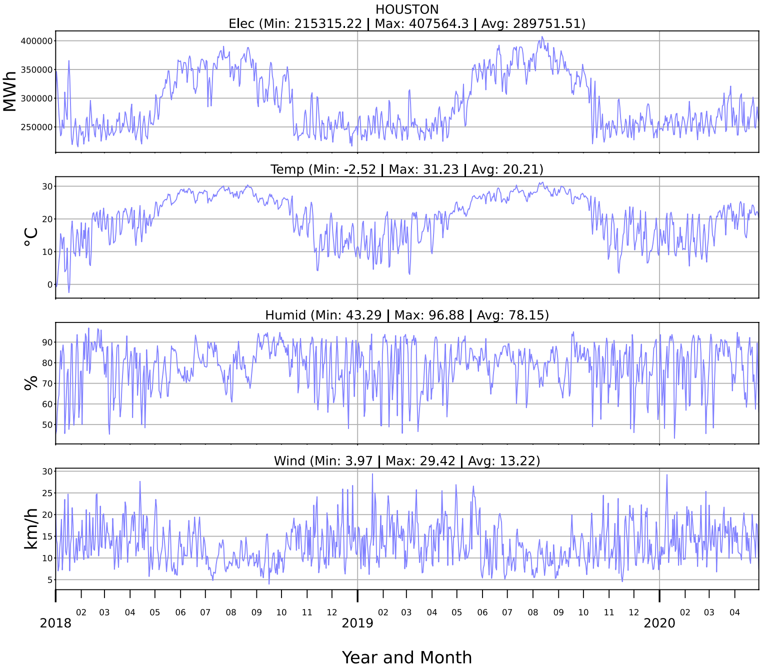


**Fig. 4** Houston dataset: daily fluctuations in electricity consumption and meteorological data

Figure 2 displays daily fluctuations in electricity consumption (Elec), temperature (Temp), relative humidity (Humid), and wind speed (Wind) for Dayton. Figure 3 shows the correlation matrix, indicating weak relationships between electricity consumption and meteorological data.

Similarly, Fig. 4 illustrates daily fluctuations in electricity consumption and meteorological variables for Houston. Figure 5 presents the correlation matrix. Unlike Dayton, Houston shows a strong correlation between electricity consumption and temperature, as evidenced by the correlation values.

## 2.2 Performance metrics

To comprehensively evaluate and compare the performance of the proposed models, three widely used error metrics were employed: N-RMSE, N-MAE, and MAPE [24]. RMSE computes the square root of the average squared differences between predicted and actual values, with units consistent with the original data, making it more sensitive to outliers. MAE measures the average magnitude of errors without squaring them, which makes it less sensitive to outliers. MAPE evaluates forecast accuracy by calculating the average absolute percentage difference between actual and predicted values. The N-RMSE and N-MAE are normalized versions of RMSE and MAE metrics by the data range. Lower values of these metrics indicate better performance. The mathematical expressions for these metrics are presented in (1), (2), and (3), respectively.

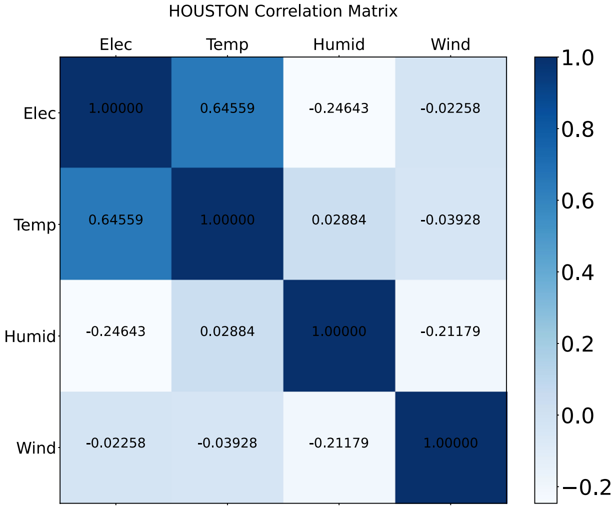


**Fig. 5** Correlation matrix showing relationships between electricity consumption and meteorological variables in Houston

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}, \quad \text{N-RMSE} = \frac{\mathrm{RMSE}}{\max(y) - \min(y)} \tag{1}$$

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|, \quad \text{N-MAE} = \frac{\mathrm{MAE}}{\max(y) - \min(y)} \tag{2}$$

$$\mathrm{MAPE} = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \times 100 \tag{3}$$

## 2.3 Forecasting models

The following subsections detail the architectures of the proposed forecasting models, including the customized LSTM model, CNN model, and their ensemble.

### 2.3.1 LSTM model

LSTM networks, a specialized type of recurrent neural networks (RNNs), are designed to effectively model sequential data by capturing long-term dependencies through their unique memory cell architecture. This structure enables the network to selectively store, update, and retrieve information across extended sequences, preserving relevant contextual information during learning [25].

The core of the LSTM architecture consists of three gates: the forget gate, input gate, and output gate. The forget gate determines which information from the previous cell state should be discarded, the input gate controls the extent to which new information is stored in the memory cell, and the output gate decides the portion of the memory cell that contributes to the hidden state at the current time step.

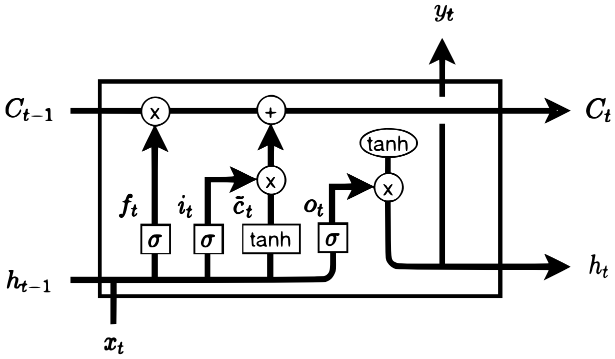


**Fig. 6** Schematic diagram of a typical LSTM cell [25]

**Table 2** Configuration of the proposed LSTM model

| Description | Value |
|---|---|
| Model architecture | Sequential |
| Units | 14 |
| Activation function | ReLU |
| Output dimension | 24 (h) |
| Dropout rate | 0 |
| Batch size | 168 |
| Max epochs | 300 |
| Early stopping patience | 15 |
| Optimizer | Adam |
| Learning rate | 0.001 |
| Loss function | MSE |

These gates operate collectively to regulate information flow across time steps, allowing LSTMs to effectively model complex temporal patterns in data such as electricity consumption time series. A schematic overview of a typical LSTM cell is shown in Fig. 6.

In this study, a custom LSTM model was developed, deviating from traditional LSTM designs by employing the Rectified Linear Unit (ReLU) activation function instead of tanh throughout the gating mechanisms. The substitution of tanh activation with ReLU mitigates vanishing gradient issues and accelerates convergence, resulting in improved training stability for electricity consumption forecasting tasks.

The mathematical operations of the LSTM cell at each time step $t$ are defined in (4) to (9), where $x_t$ represents the input vector, $h_{t-1}$ and $C_{t-1}$ are the hidden state and cell state from the previous time step, respectively.

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f) \tag{4}$$

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i) \tag{5}$$

$$\tilde{c}_t = \mathrm{ReLU}(W_C[h_{t-1}, x_t] + b_C) \tag{6}$$

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{c}_t \tag{7}$$

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \tag{8}$$

$$h_t = o_t \odot \mathrm{ReLU}(C_t) \tag{9}$$

In these expressions, $f_t$, $i_t$, and $o_t$ represent the forget, input, and output gate activations, respectively, computed using the sigmoid activation function $\sigma(\cdot)$. The candidate cell state $\tilde{c}_t$ and the final hidden state $h_t$ are computed using the ReLU activation function, defined as $\mathrm{ReLU}(z) = \max(0, z)$. $W_f$, $W_i$, $W_C$, and $W_o$ denote the weight

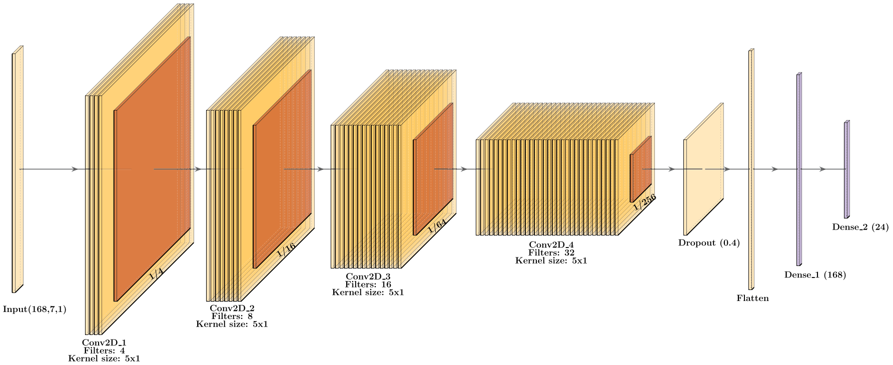


**Fig. 7** Proposed CNN architecture

**Table 3** Configuration of the proposed CNN model

| Description | Value |
|---|---|
| Model architecture | Sequential |
| Input shape | (168, 7, 1) |
| Conv2D_1 | Filters: 4, Kernel size: (5, 1) |
| MaxPooling2D_1 | Pool size: (4, 1) |
| Conv2D_2 | Filters: 8, Kernel size: (5, 1) |
| MaxPooling2D_2 | Pool size: (4, 1) |
| Conv2D_3 | Filters: 16, Kernel size: (5, 1) |
| MaxPooling2D_3 | Pool size: (4, 1) |
| Conv2D_4 | Filters: 32, Kernel size: (5, 1) |
| MaxPooling2D_4 | Pool size: (4, 1) |
| Dropout rate | 0.4 |
| Dense layer | 168 neurons |
| Output dense layer | 24 neurons (h) |
| Activation function | ReLU |
| Batch size | 128 |
| Max epochs | 300 |
| Early stopping patience | 30 |
| Optimizer | Adam |
| Learning rate | 0.001 |
| Loss function | MSE |

matrices, and $b_f$, $b_i$, $b_C$, and $b_o$ are the corresponding bias vectors. The operator $\odot$ denotes element-wise multiplication.

Following the update of the hidden state $h_t$ as defined in (9), the output is passed through a fully connected dense layer to generate the 24-h electricity consumption forecast for the next day, as shown in (10).

$$\hat{Y}_t = \text{Dense}(h_t) \in \mathbb{R}^{24} \tag{10}$$

This architecture allows the model to transform a sequence of 168 hly input vectors into a single-step multi-output forecast, effectively capturing both short-term and long-term temporal dependencies. The configuration of the proposed LSTM model is provided in Table 2.

### 2.3.2 CNN model

CNNs have also been applied to time series forecasting tasks with promising results [15]. A CNN typically consists of convolutional layers, pooling layers, fully connected layers, and activation functions.

In this study, a custom CNN architecture was developed for forecasting, as shown in Fig. 7, with configuration provided in Table 3. The CNN model includes four convolutional and max pooling layers. While Conv1D and MaxPooling1D are common for univariate electricity consumption predictions, they output based on the Filters value, limiting training on multiple attributes. For example, inputs of (168, 7) with Conv1D (Filters = 32) yield (168, 32), and after MaxPooling1D, the input becomes (84, 32), resulting in information loss across features. To overcome this issue, Conv2D and MaxPooling2D layers were used. By inputting data as (168, 7, 1) to Conv2D layers, outputs become (168, 7, 32), preserving attribute information. This modification significantly improved the CNN model's prediction performance with multiple features.

The Conv2D operation at each layer is defined by (11), where $X$ is the input tensor, $W$ is the kernel of size $K_T \times K_F$, $b_k$ is the bias term, and ReLU is the activation function defined as $\max(0, z)$.

$$Y_{i,j,k} = \mathrm{ReLU}\left(\sum_{m=0}^{K_T-1}\sum_{n=0}^{K_F-1} W_{m,n,k} \cdot X_{i+m,j+n} + b_k\right) \tag{11}$$

Each MaxPooling2D layer reduced the time dimension by applying the operation in (12), where $\Omega$ represents the pooling window (4, 1). The pooling step helped reduce computation while retaining important temporal patterns.

$$P_{i,j,k} = \max_{(m,n)\in\Omega} Y_{i+m,j+n,k} \tag{12}$$

### 2.3.3 Ensemble model

Ensemble learning is a powerful machine learning technique that combines multiple models to enhance accuracy, generalizability, and robustness by leveraging their individual strengths. Common ensemble methods include averaging, voting, bagging, boosting, and stacking [10].

In this study, a weighted averaging ensemble approach was used. The individual predictions from the LSTM and CNN models were combined to generate the final forecast. Weighted averaging was preferred over simple averaging, as it consistently produced slightly better results in our experiments.

Let $\hat{y}_{\text{LSTM}}$ and $\hat{y}_{\text{CNN}}$ represent the outputs of the LSTM and CNN models, respectively. The final ensemble prediction $\hat{y}_{\text{Ensemble}}$ is computed as shown in (13), where the weights ($w_{\text{LSTM}}$, $w_{\text{CNN}}$) are calculated based on the inverse of the average N-RMSE of each model on the validation set, as in (14). This error-based weighting assigns higher importance to the model with lower prediction error.

$$\hat{y}_{\text{Ensemble}} = w_{\text{LSTM}} \cdot \hat{y}_{\text{LSTM}} + w_{\text{CNN}} \cdot \hat{y}_{\text{CNN}} \tag{13}$$

$$w_{\text{model}} = \frac{1/\text{N-RMSE}_{\text{model}}}{\sum_i 1/\text{N-RMSE}_i} \tag{14}$$

This ensemble strategy allowed the model to combine the temporal learning capability of the LSTM with the pattern extraction strength of the CNN. As a result, the combined model delivers improved performance and robustness compared to using either model individually.

## 3 Experimental work and discussions

This section presents the experimental results of training and evaluating the proposed models on the Dayton and Houston datasets using metrics given in previous sections. The analysis highlights each model's strengths and weaknesses and examines the influence of feature sets on forecasting performance, focusing on meteorological and time-related factors in short-term electricity consumption prediction.

Before training, a structured preprocessing pipeline was applied to the input time series. This process included handling missing values, normalizing the data, encoding temporal features, and preparing the sequences through a sliding window approach.

Approximately 0.02% of the electricity consumption data contained missing or abnormal values, particularly in the electricity consumption series. These values were corrected using a two-step time-based replacement method. Initially, missing or abnormal values were substituted with the corresponding value from the previous day. If unavailable, the average value for the same hour over the past seven days was used instead. This method preserved daily and weekly patterns while minimizing sudden fluctuations due to data anomalies.

Following data cleaning, the data were then rescaled using min-max normalization within the training data range. The same thresholds were applied to the validation and test datasets. To prevent data leakage, any values falling outside the normalized range were clipped to ensure consistency.

Temporal features including month, day of the week, and hour were cyclically encoded using integer values within fixed ranges. Month was encoded in the range [1–12], day of the week in [1–7], and hour of the day in [0–23]. This method helped maintain the periodic nature of temporal features without introducing unnecessary complexity.

Finally, a sliding window method was implemented to create supervised learning samples. For each prediction point, electricity and the other features from the preceding 168 h were used to forecast electricity consumption for the next 24 h. This approach enabled the models to learn both short-term trends and long-term patterns effectively.

The proposed models were trained on 16 distinct feature sets for each dataset. Their forecasting performances were then comparatively analyzed. The list of feature sets used in the experiments is provided in Table 4.

Each dataset was split into 9144 h (381 days) for training, 2352 h (98 days) for validation, and 8928 h (372 days) for testing, yielding 374, 91, and 365 input–output pairs, respectively. Models were trained for up to 300 epochs with early stopping set to 15 for LSTM and 30 for CNN to prevent overfitting.

To ensure reproducibility and reduce random variability in model initialization and training, a fixed random seed value of 47, which controls weight initialization, data shuffling, and other stochastic operations, was used in all experiments. Using the same seed ensures that repeated runs under identical conditions produce consistent results. It also enabled fair evaluation across different feature sets by removing randomness as a confounding factor. While previous studies on electricity load forecasting have often ignored this step, this work emphasizes its importance for reliability and comparability.

All experiments were performed on a workstation equipped with an Intel Xeon E5-1630 v3 CPU at 3.70 GHz and 32 GB of RAM, running a 64-bit Windows 10 operating system. The models were built using Python 3.9 with TensorFlow and Keras libraries in the PyCharm development environment. Hyperparameter tuning was conducted with the Keras-Tuner library.

The following sections provide a detailed analysis of the prediction models trained on the Dayton and Houston datasets, with a critical evaluation of key features affecting predictive accuracy and model effectiveness.

**Table 4** The feature sets used in the proposed models

| No | Feature Set |
|---|---|
| 1 | [Elec] |
| 2 | [Elec+ Temp] |
| 3 | [Elec+ Humid] |
| 4 | [Elec+ Wind] |
| 5 | [Elec+ Time] |
| 6 | [Elec+ Temp+ Humid] |
| 7 | [Elec+ Temp+ Wind] |
| 8 | [Elec+ Temp+ Time] |
| 9 | [Elec+ Time+ Humid] |
| 10 | [Elec+ Time+ Wind] |
| 11 | [Elec+ Humid+ Wind] |
| 12 | [Elec+ Temp+ Humid+ Wind] |
| 13 | [Elec+ Temp+ Humid+ Time] |
| 14 | [Elec+ Time+ Humid+ Wind] |
| 15 | [Elec+ Temp+ Time+ Wind] |
| 16 | [Elec+ Temp+ Humid+ Wind+ Time] |

### 3.1 LSTM model performance

For the Dayton dataset, Fig. 8 illustrates the loss curve of the LSTM model trained exclusively on the electricity consumption feature over 300 epochs. The x-axis represents the number of epochs, and the y-axis denotes the loss values. This loss curve enables the evaluation of the model's convergence, potential overfitting or underfitting, and overall training effectiveness. The model was trained for 300 epochs, taking 55.48 s, without triggering the early stopping mechanism and showed no signs of overfitting. Similarly, other LSTM models on this dataset exhibited comparable loss curves, confirming the absence of overfitting.

Figure 9 presents the average N-RMSE, N-MAE, and MAPE values from LSTM models with different feature sets. Incorporating temperature, relative humidity, wind speed, and timestamp alongside electricity consumption enhances prediction performance, reducing N-RMSE, N-MAE, and MAPE by 19.79%, 20.38%, and 22.31%, respectively.

The N-RMSE, N-MAE, and MAPE values for the model trained solely on electricity consumption were 0.24, 0.20, and 5.96%, respectively. The model performed best in winter and worst in summer. Seasonal average MAPEs were measured as 4.94% in winter, 6.02% in spring, 7.08% in summer, and 5.28% in autumn. Additionally, the annual average MAPE was higher on weekends (7.65%) compared to weekdays (5.28%). Among models with one additional feature, the `[Elec+Time]` set achieved the best performance. This result indicates that the timestamp feature significantly enhances prediction performance compared to other features.

Experiments with multiple feature combinations revealed that the `[Elec+Temp+Humid+Time]` set had the best overall performance, with N-RMSE of 0.19, N-MAE of 0.16, and MAPE of 4.63%. The model maintained the

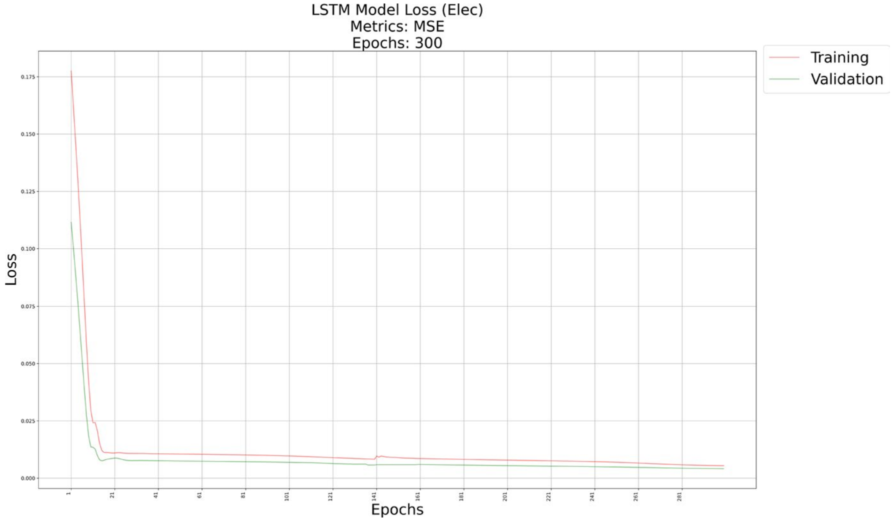
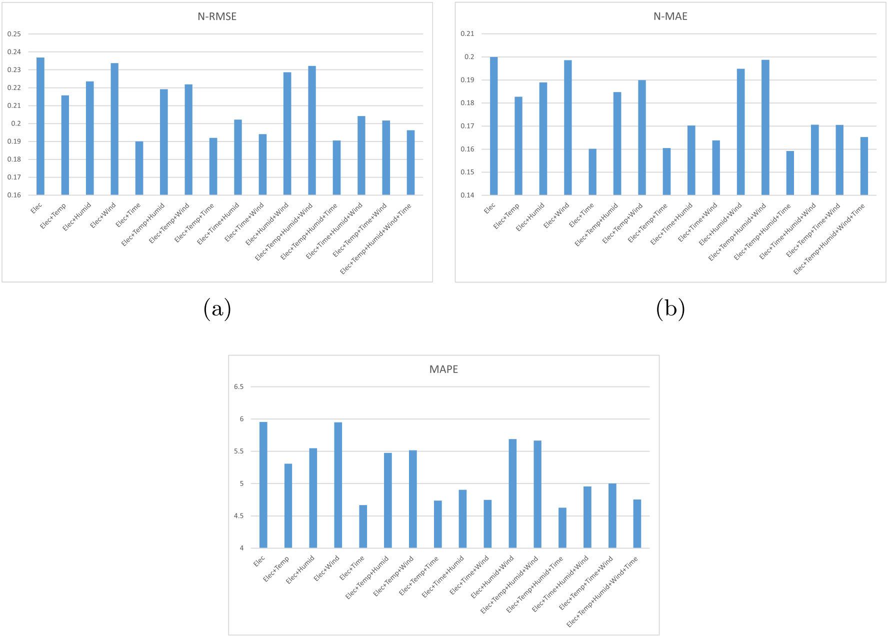


**Fig. 8** Dayton dataset: loss graph of LSTM model trained with electricity consumption feature



(a)

(b)

(c)

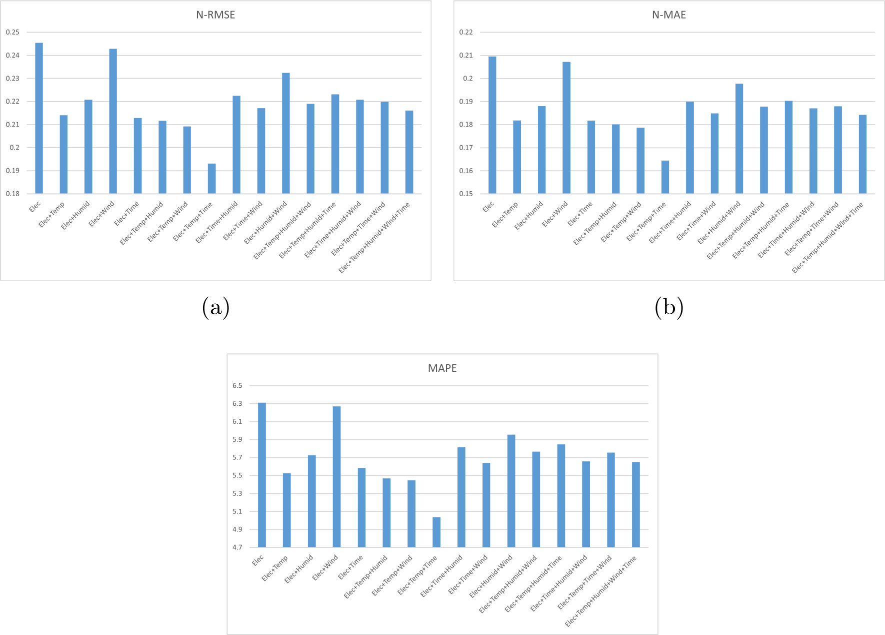
**Fig. 9** Dayton dataset: average **a** N-RMSE, **b** N-MAE, and **c** MAPE values obtained with the LSTM model for different feature sets

best performance in winter and the worst in summer, with seasonal average MAPEs of 4.22% for winter, 4.35% for spring, 5.01% for summer, and 4.91% for autumn. Incorporating temperature, timestamp, and humidity enhanced both seasonal and weekend predictions, resulting in yearly average MAPEs of 4.66% for weekdays and 4.54% for weekends. These improvements highlight the importance of the timestamp feature in the model's prediction performance.

For the Houston dataset, the loss graph of the LSTM model trained exclusively on the electricity consumption feature exhibited characteristics similar to those of the initial model. The training process was completed in 87 epochs within a duration of 15.46 s. Analysis of the loss graph indicated no signs of overfitting, which is consistent with other models trained on this dataset.

Figure 10 presents the average N-RMSE, N-MAE, and MAPE values for various feature sets. Incorporating temperature, relative humidity, wind speed, and timestamp features alongside electricity consumption data improved the model's learning capacity, reducing N-RMSE, N-MAE, and MAPE by 21.34%, 21.53%, and 20.19%, respectively.

The N-RMSE, N-MAE, and MAPE values for the model trained solely on electricity consumption were 0.25, 0.21, and 6.31%, respectively. The model performed best in winter and worst in summer, with seasonal MAPE values of 5.41%, 7.27%, 5.69%, and 6.88% for winter, spring, summer, and autumn, respectively. The yearly MAPE was 6.02% on weekdays and 7.03% on weekends. Among models with one



(a)

(b)

(c)

**Fig. 10** Houston dataset: average **a** N-RMSE, **b** N-MAE, and **c** MAPE values obtained with the LSTM model for different feature sets

additional feature, the `[Elec+Temp]` set achieved the best performance. The temperature feature had the most significant impact on performance.

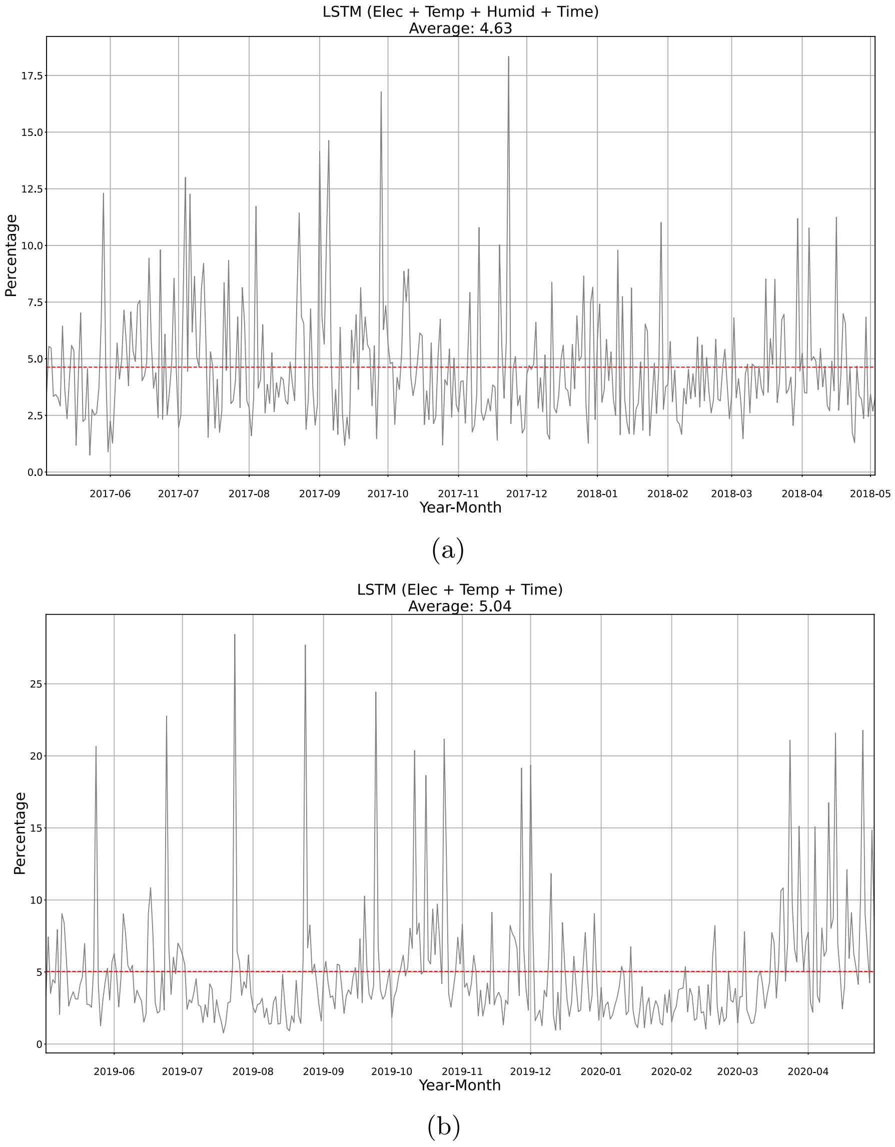
The `[Elec+Temp+Time]` feature set yielded the best overall results with N-RMSE of 0.19, N-MAE of 0.16, and MAPE of 5.04%. It performed best in winter and worst in summer, with seasonal MAPE values of 3.50%, 6.37%, 4.53%, and 5.75%, respectively. Incorporating temperature and timestamp features improved seasonal predictions, achieving minimal differences between weekday and weekend MAPE values at 5.03% and 5.07%, respectively.

Figure 11 displays the MAPE values of the LSTM models with the best performing feature sets on the entire test set of each respective dataset.

### 3.2 CNN model performance

For the Dayton dataset, the CNN model loss graph showed similar characteristics to the LSTM models, with training completed in 231 epochs within 11.78 s. The loss graph analysis indicated no overfitting, consistent with other CNN models trained on this dataset.

Figure 12 presents the average N-RMSE, N-MAE, and MAPE values for various feature sets. Incorporating temperature, relative humidity, wind speed, and timestamp features alongside electricity consumption improved the model's performance,



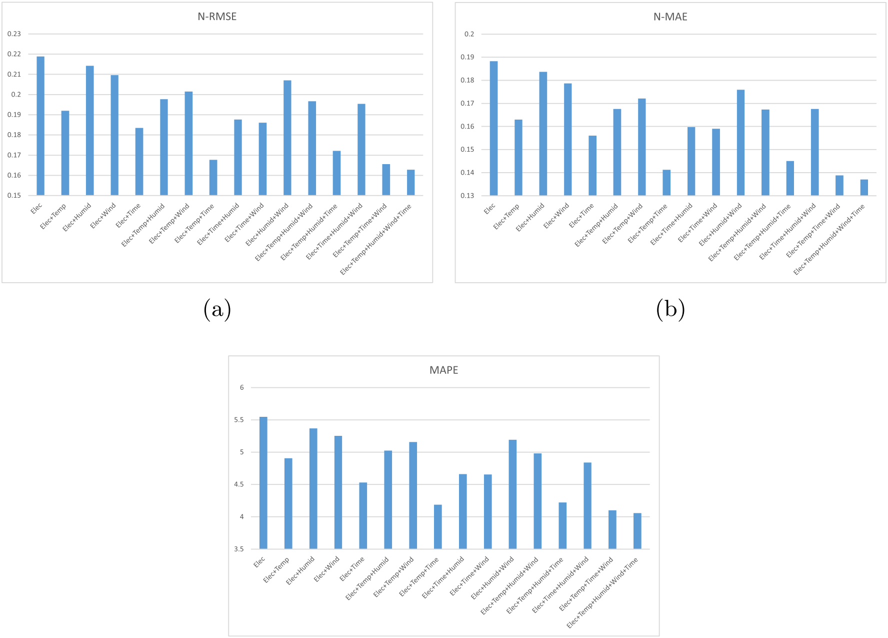
(a)

(b)

**Fig. 11** The MAPE values of the LSTM models with the best performing feature sets on the entire test set of the **a** Dayton and **b** Houston datasets

reducing N-RMSE, N-MAE, and MAPE by 25.58%, 27.24%, and 26.85%, respectively.

The N-RMSE, N-MAE, and MAPE values for the model trained solely on electricity consumption were 0.22, 0.19, and 5.55%, respectively. The model performed best in winter and worst in summer, with seasonal MAPE values of 5.11%, 5.21%,



(a)

(b)

(c)

**Fig. 12** Dayton dataset: average **a** N-RMSE, **b** N-MAE, and **c** MAPE values obtained with the CNN model for different feature sets

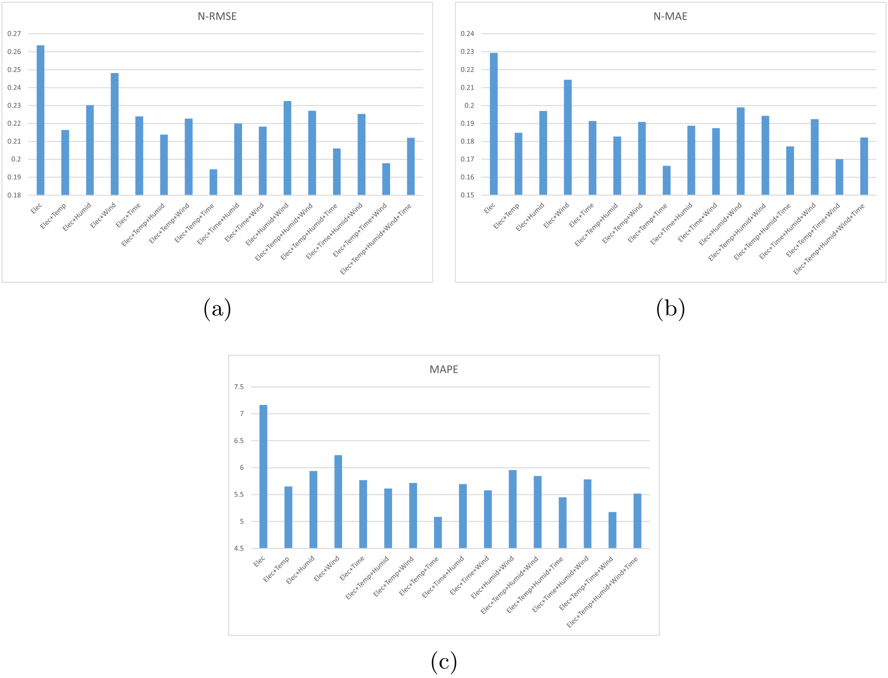
6.16%, and 5.68% for winter, spring, summer, and autumn, respectively. The yearly MAPE was 5.15% on weekdays and 6.53% on weekends. Among models with one additional feature, the `[Elec+Time]` set achieved the best performance.

The `[Elec+Temp+Humid+Wind+Time]` feature set achieved the best overall performance with N-RMSE of 0.16, N-MAE of 0.14, and MAPE of 4.05%. It performed best in winter and worst in summer, with seasonal MAPE values of 3.41%, 3.80%, 4.87%, and 4.13%, respectively. Weekday and weekend MAPE values were 3.99% and 4.22%, respectively.

For the Houston dataset, the CNN model loss graph exhibited similar characteristics to the LSTM models. The training process for this model was completed in 145 epochs within 8.25 s, with no evidence of overfitting, consistent with other CNN models trained on this dataset.

Figure 13 presents the average N-RMSE, N-MAE, and MAPE values for various feature sets. Incorporating temperature, relative humidity, wind speed, and timestamp features alongside electricity consumption improved the model's performance, reducing N-RMSE, N-MAE, and MAPE by 26.24%, 27.45%, and 28.99%, respectively.

The N-RMSE, N-MAE, and MAPE values for the model trained solely on electricity consumption were 0.26, 0.23, and 7.16%, respectively. The model performed best in winter and the worst in summer. The MAPE values were 5.37%, 7.39%, 8.33%, and 7.57% for winter, spring, summer, and autumn, respectively. The yearly



(a)

(b)

(c)

**Fig. 13** Houston dataset: average **a** N-RMSE, **b** N-MAE, and **c** MAPE values obtained with the CNN model for different feature sets

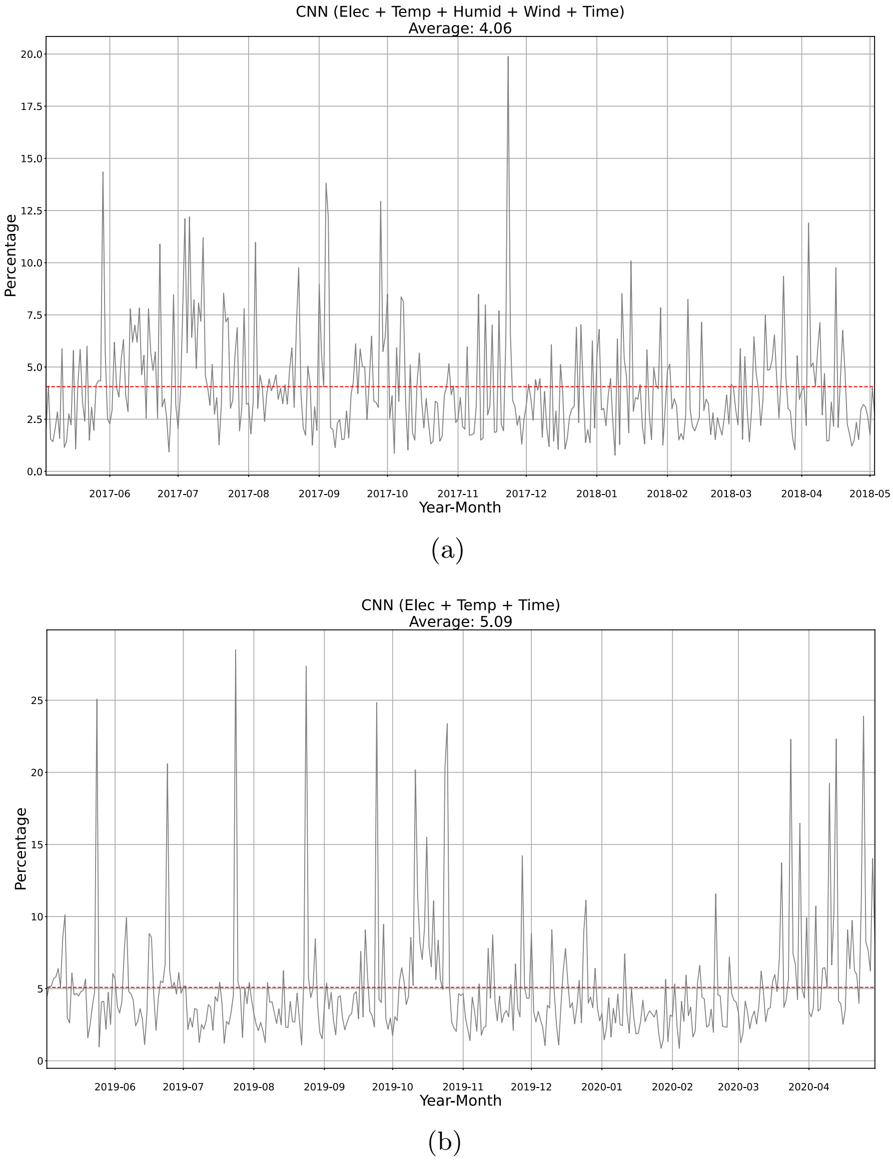
MAPE differences between weekdays and weekends were minimal, with values of 7.18% and 7.13%, respectively. The `[Elec+Temp]` feature set provided the best prediction performance among single additional features.

The `[Elec+Temp+Time]` feature set achieved the best overall performance with N-RMSE of 0.19, N-MAE of 0.17, and MAPE of 5.09%. This suggests that the temperature and timestamp features contribute more significantly to electricity consumption prediction compared to other features in the proposed CNN model. The seasonal MAPE values were calculated as 3.82%, 6.33%, 4.68%, and 5.52% for winter, spring, summer, and autumn, respectively. Weekend predictions were slightly more accurate, with MAPE values of 4.93% compared to 5.15% for weekdays.

Figure 14 displays the MAPE values of the CNN models with the best performing feature sets on the entire test set of each respective dataset.

## 3.3 Ensemble model performance

For the Dayton dataset, Fig. 15 shows the average N-RMSE, N-MAE, and MAPE values from ensemble models trained with different feature sets. Adding temperature, relative humidity, wind speed, and timestamp features alongside electricity consumption improves the model's learning capacity and prediction performance, reducing N-RMSE, N-MAE, and MAPE by approximately 23%.

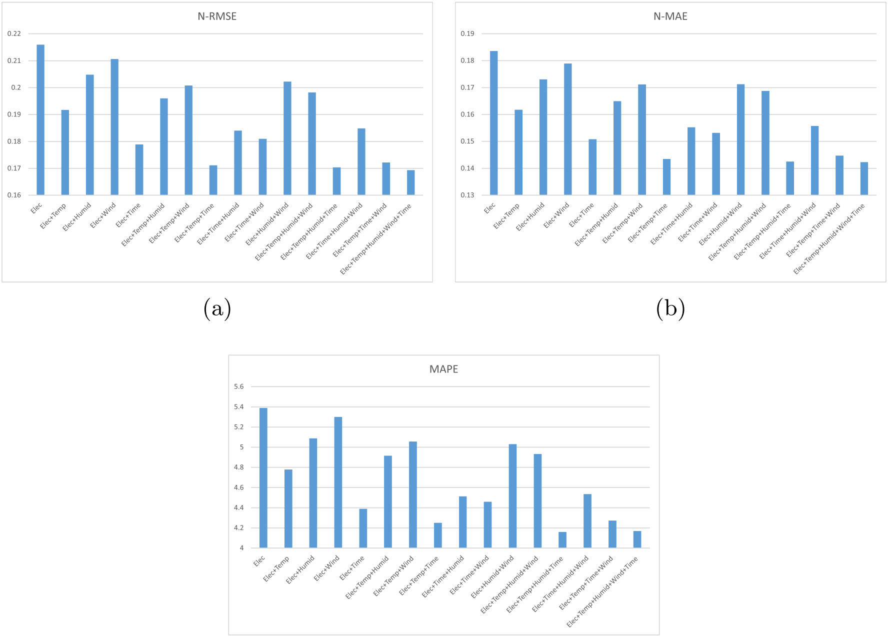


(a)

(b)

**Fig. 14** The MAPE values of the CNN models with the best performing feature sets on the entire test set of the **a** Dayton and **b** Houston datasets

The N-RMSE, N-MAE, and MAPE values for the model trained solely on electricity consumption were 0.21, 0.18, and 5.39%, respectively. The model performed best in winter and worst in summer, with seasonal MAPE values of 4.77%, 5.41%, 5.97%, and 5.37% for winter, spring, summer, and autumn, respectively. The yearly MAPE values for weekdays and weekends were 4.85% and



(a)

(b)

(c)

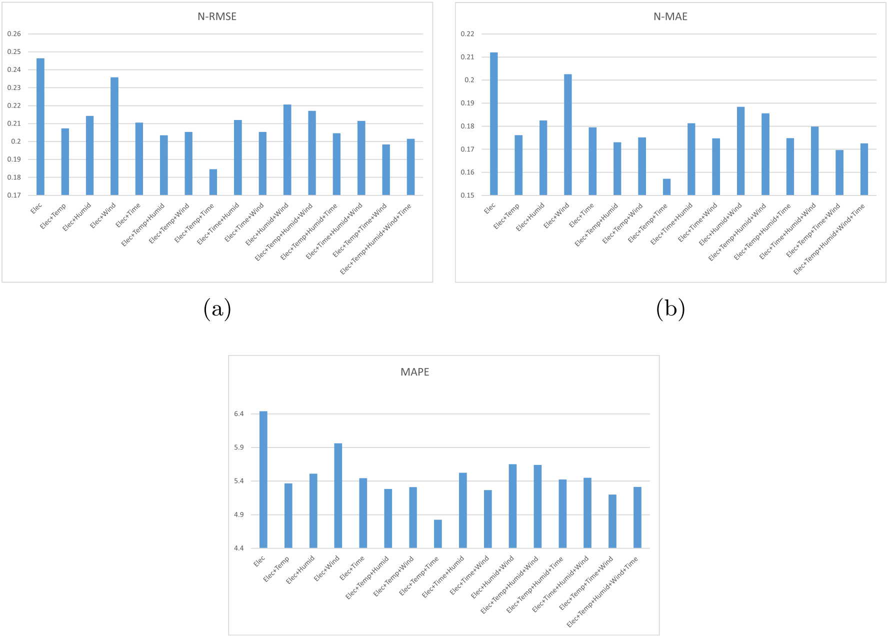
**Fig. 15** Dayton dataset: average **a** N-RMSE, **b** N-MAE, and **c** MAPE values obtained with the ensemble model for different feature sets

6.72%, respectively. The `[Elec+Time]` feature set achieved the best performance among single additional features.

The `[Elec+Temp+Humid+Time]` feature set provided the best overall performance, with N-RMSE, N-MAE, and MAPE values of 0.17, 0.14, and 4.15%. The model showed the best performance in winter and the worst in summer, with seasonal MAPE values of 3.68%, 3.90%, 4.70%, and 4.35%. Including additional features improved seasonal predictions. The MAPE values for weekdays and weekends were 4.14% and 4.19%, respectively, indicating slightly better accuracy for weekday predictions.

For the Houston dataset, Fig. 16 presents the average N-RMSE, N-MAE, and MAPE values obtained from ensemble models trained with various feature sets. Incorporating temperature, relative humidity, wind speed, and timestamp features alongside electricity consumption improves the model's learning capacity and enhances prediction performance, reducing N-RMSE, N-MAE, and MAPE by approximately 26%.

The N-RMSE, N-MAE, and MAPE values for the model trained solely on electricity consumption were 0.24, 0.21, and 6.43%, respectively. The model performed best in winter and worst in spring, with seasonal MAPE values of 5.33%, 7.25%, 6.15%, and 7.01% for winter, spring, summer, and autumn, respectively. The yearly MAPE values for weekdays and weekends were 6.32% and 6.71%,



(a)

(b)

(c)

**Fig. 16** Houston dataset: average **a** N-RMSE, **b** N-MAE, and **c** MAPE values obtained with the ensemble model for different feature sets

respectively. The `[Elec+Temp]` feature set provided the best prediction performance among single additional features.

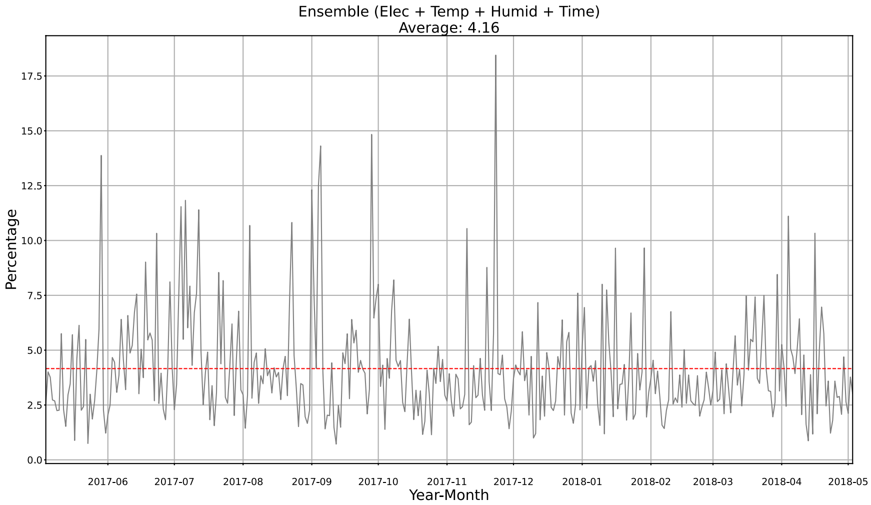
The `[Elec+Temp+Time]` feature set achieved the best overall performance, with N-RMSE, N-MAE, and MAPE values of 0.18, 0.15, and 4.82%, respectively. The seasonal MAPE values were calculated as 3.38%, 6.10%, 4.43%, and 5.38% for winter, spring, summer, and autumn, respectively. Weekend predictions were more accurate, with MAPE values of 4.68% compared to 4.88% for weekdays.

Figure 17 displays the MAPE values of the Ensemble models with the best performing feature sets on the entire test set of each respective dataset.

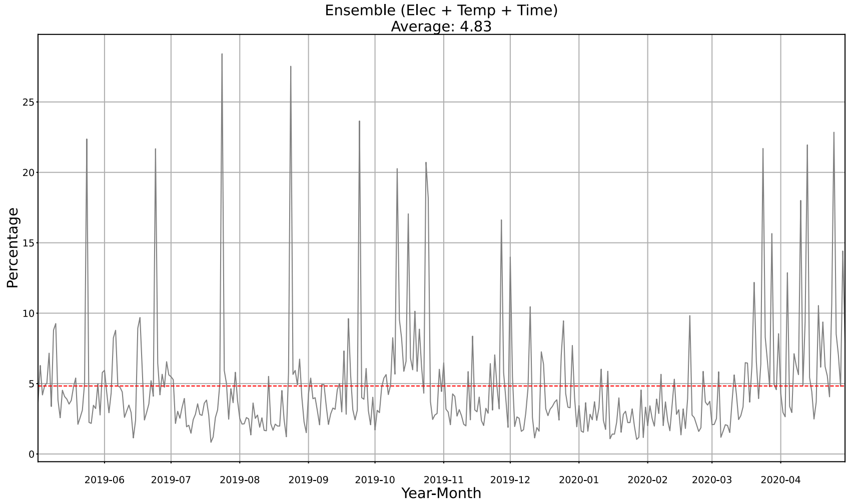
### 3.4 Daily performance analysis

To facilitate a deeper examination and comparative analysis of the proposed models' forecasting performance, representative days were selected from the Dayton and Houston test sets. These particular days serve as reference points, demonstrating that the models generally exhibit consistent forecasting behavior across the majority of the remaining test days.

In evaluating the electricity consumption models under different conditions, both the standalone `[Elec]` feature and each model's optimal feature set were employed. Figure 18 present the actual consumption values, models forecasts, and



(a)

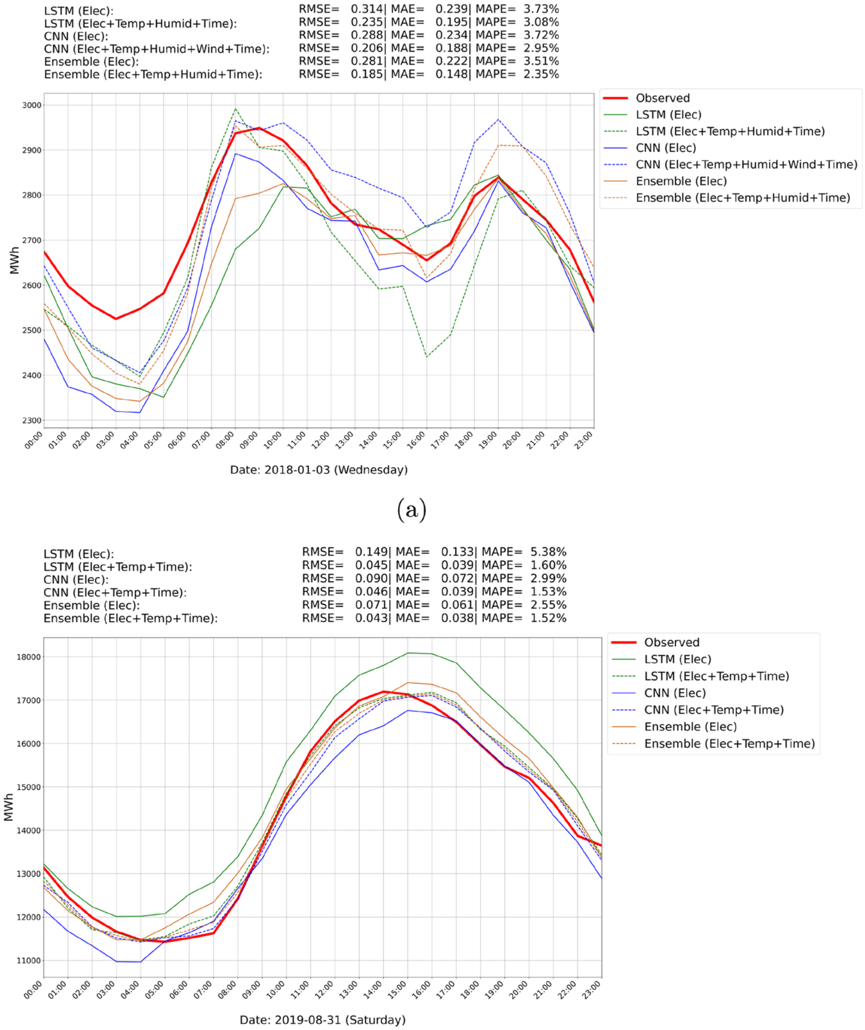


(b)

**Fig. 17** The MAPE values of the Ensemble models with the best performing feature sets on the entire test set of the **a** Dayton and **b** Houston datasets

forecast performance metrics for the chosen days. These graphical representations enable a side-by-side comparison of model outputs and provide insights into each model's accuracy and consistency.

As illustrated in the figure, ensemble models delivered superior forecast performance compared to other approaches for these selected days. Specifically, “Ensemble (Elec+Temp+Humid+Time)" achieved the most accurate results for



(a)

(b)

**Fig. 18** Representative days: **a** Dayton and **b** Houston datasets

the Dayton dataset, while “Ensemble (Elec+Temp+Time)" performed best for the Houston dataset—both displaying the lowest RMSE, MAE, and MAPE values. It is important to note, however, that these outcomes may be specific to the highlighted days and could vary under different conditions. In certain cases, LSTM or CNN models can produce more accurate forecasts.

**Table 5** Average training time (ms) per epoch

| City | LSTM | CNN | Ensemble |
|---|---|---|---|
| Dayton | 178.23 | 73.13 | 251.36 |
| Houston | 178.95 | 75.26 | 254.21 |

**Table 6** Average test time (ms) per day

| City | LSTM | CNN | Ensemble |
|---|---|---|---|
| Dayton | 0.93 | 0.45 | 1.42 |
| Houston | 0.87 | 0.44 | 1.47 |

### 3.5 Computational time analysis

Computational time during training and testing is a relevant factor for evaluating model usability. Table 5 shows the average training time per epoch for the models. CNN models trained faster than LSTM models on both datasets. This difference can be attributed to the structural design of CNNs, which process input in parallel, whereas LSTMs operate sequentially. Ensemble models, which integrate both LSTM and CNN architectures, inherently involve training both sub-models. Therefore, the training time of the ensemble model corresponds to the sum of the training times required by the individual LSTM and CNN models.

Table 6 shows the average test time per daily prediction. Similar to the training phase, CNN models exhibit shorter inference durations due to their efficient use of parallel processing. In contrast, ensemble models require execution of both LSTM and CNN predictions and a subsequent combination step, which moderately increases the inference time. Despite this overhead, the added complexity remains manageable, maintaining suitability for near real-time applications.

In conclusion, CNN models provide a time-efficient alternative to LSTM models, especially when shorter training and inference times are desired. Their computational advantage makes them suitable for time-sensitive forecasting applications without compromising model flexibility. Ensemble models, while introducing a moderate increase in training and prediction durations due to the dual-network structure, yield enhanced accuracy and robustness. This trade-off between execution time and predictive performance makes ensemble architectures particularly appealing for applications where precise load forecasting is critical and a slight increase in runtime is acceptable.

### 3.6 Comparison with literature

The performance assessment of the proposed models in comparison to existing literature, as summarized in Table 7, underscores their superior predictive capabilities. The table highlights the best results achieved with the relevant models. As can be seen from the table, the proposed LSTM, CNN, and ensemble models

**Table 7** Comparison of the performances of the proposed models with the literature

| Model | N-RMSE | N-MAE | MAPE (%) | Horizon (h) |
|---|---|---|---|---|
| CNN [15] | 0.502 | 0.373 | – | 1 |
| SVR [15] | 0.620 | 0.493 | – | 1 |
| GRU [15] | 0.614 | 0.488 | – | 1 |
| MLP [15] | 0.519 | 0.382 | – | 1 |
| LSTM [15] | 0.383 | 0.251 | – | 1 |
| Linear Regression [15] | 0.570 | 0.404 | – | 1 |
| Decision Tree [15] | 0.571 | 0.359 | – | 1 |
| ELM [15] | 0.563 | 0.427 | – | 1 |
| MLP [26] | – | – | 4.97 | 24 |
| LSTM [26] | – | – | 5.17 | 24 |
| CNN [26] | – | – | 5.62 | 24 |
| SDA [27] | – | – | 12.39 | 24 |
| MLR [27] | – | – | 11.24 | 24 |
| BEM [27] | – | – | 9.09 | 24 |
| ARIMAX [27] | – | – | 14.74 | 24 |
| ARIMAX+BEM [27] | – | – | 10.41 | 24 |
| BEM+ARIMAX [27] | – | – | 9.03 | 24 |
| BlackInter LLM (zero-shot) [22] | – | – | 7.87 | 3 |
| BlackInter LLM (few-shot) [22] | – | – | 7.39 | 3 |
| *Proposed models* | | | | |
| LSTM | 0.191 | 0.159 | 4.63 | 24 |
| CNN | 0.163 | 0.137 | 4.06 | 24 |
| Ensemble | 0.170 | 0.142 | 4.16 | 24 |

consistently outperform the models presented in the literature across all error metrics: N-RMSE, N-MAE, and MAPE. The CNN model developed in our study achieved the lowest N-RMSE, N-MAE, and MAPE values, which were recorded at 0.163, 0.137, and 4.06%, respectively. In contrast, the LSTM model exhibited slightly higher values of 0.191, 0.159, and 4.63%, respectively. The ensemble model, combining LSTM and CNN predictions, achieved balanced performance with N-RMSE, N-MAE, and MAPE values of 0.170, 0.142, and 4.16%, respectively. Notably, these models significantly outperform the best-performing model of [15], showcasing a 57.4% and 45.5% reduction in N-RMSE and N-MAE, respectively, even with a longer forecasting horizon. Furthermore, our proposed CNN model demonstrated a notable improvement in MAPE compared to the MLP, LSTM, and CNN models developed by [26], with reductions of 18.3%, 21.5%, and 27.8%, respectively.

The proposed models also exhibit substantially lower MAPE values than those reported by [27], with the best-performing model from their study (BEM+ARIMAX) having a MAPE of 9.03%. Recently, Zhou et al. [22] introduced BlackInter, a novel electricity forecasting framework built upon a pretrained LLM. BlackInter leverages zero-shot and few-shot learning to predict electricity demand without extensive retraining. In their experiments, the zero-shot and few-shot versions of BlackInter achieved MAPE values of 7.87% and 7.39%, respectively, over a 3-h horizon. While BlackInter demonstrated strong adaptability and generalization in short-term forecasting tasks, our proposed CNN and ensemble models provided superior accuracy for the more challenging and practically relevant 24-h forecasting horizon.

In summary, the comprehensive comparison confirms the robustness and reliability of our proposed models across multiple benchmarks. The consistent improvements in predictive accuracy over a wide range of models in the literature highlight their suitability for real-world electricity consumption forecasting applications. Future research may explore further integration of pretrained LLM approaches with traditional deep learning architectures to combine the strengths of both paradigms.

## 4 Conclusions

This study proposed novel and robust models, incorporating deep learning as well as meteorological and time-related features, for short-term electricity consumption forecasting. The proposed LSTM, CNN, and ensemble-based models, trained on datasets from Dayton, Ohio, and Houston, Texas, achieved notable performance, with N-RMSE values as low as 0.16, N-MAE values as low as 0.14, and MAPE values as low as 4.06%.

The study's key findings can be summarized as the significant impact of meteorological and time attributes on model performance, the ensemble model's superior stability compared to individual models, the varying performance across seasons and weekdays, and the influence of dataset choice. Among all models, the CNN model achieved the best overall forecasting accuracy across all evaluation metrics, demonstrating its strong ability to extract meaningful patterns from multivariate time series data. The ensemble approach, which combines the strengths of both LSTM and CNN architectures, provided slightly lower accuracy than CNN alone but offered more consistent results across different scenarios. Notably, the weighted average ensemble method slightly outperformed the simple average approach, indicating the benefit of optimizing model contributions based on individual performance.

This work distinguishes itself from previous studies through several methodological contributions. First, a customized LSTM architecture using ReLU activations was developed to improve training stability and reduce vanishing gradient problems, which is rarely addressed in electricity forecasting. Second, a customized CNN model with Conv2D layers was used to maintain multivariate feature relationships and avoid information loss common in Conv1D approaches. Third, a structured experimental design was applied using fixed random seeds to ensure fair and reproducible comparisons across feature sets and model types. Additionally, the evaluation was conducted on two geographically distinct datasets to examine model generalization. The analysis of 16 different feature set combinations provided a thorough exploration of the influence of input features on forecasting performance.

Overall, the research contributes to the field of electricity consumption forecasting by proposing effective deep learning models and providing valuable insights into how meteorological and time-related variables affect prediction accuracy. The customized models demonstrated strong generalization abilities and showed potential for practical applications in real-world forecasting tasks.

Future work may include investigating more advanced deep learning models, incorporating additional external data sources, and addressing specific challenges such as extreme weather events and irregular consumption patterns. An interesting research direction is to use large pretrained language models (LLMs), which can be fine-tuned for electricity demand forecasting. This approach could benefit from the strong generalization and transfer learning capabilities of LLMs to further improve forecasting accuracy.

**Acknowledgements** This work was supported by Eskisehir Technical University Scientific Research Projects Grant 22LOT247. Open access funding was provided by the Scientific and Technological Research Council of Turkiye (TUBITAK). The authors used Grammarly and Google Gemini for improving language and readability of this work.

**Funding** Open access funding provided by the Scientific and Technological Research Council of Türkiye (TÜBİTAK).

## References

1. Karrabi M, Jabari F, Foroud AA (2025) A green ammonia and solar-driven multi-generation system: thermo-economic model and optimization considering molten salt thermal energy storage, fuel cell vehicles, and power-to-gas. Energy Convers Manag 323:119226
2. Moraglio, F., Ragusa, C.: Day-ahead electricity price forecasting in the contemporary italian market. IEEE Access (2024)
3. Shen B, Hove A, Hu J, Dupuy M, Bregnbæk L, Zhang Y, Zhang N (2024) Coping with power crises under decarbonization: the case of china. Renew Sustain Energy Rev 193:114294
4. Qureshi M, Arbab MA, Rehman SU (2024) Deep learning-based forecasting of electricity consumption. Sci Rep 14(1):6489
5. Pappas SS, Ekonomou L, Karamousantas DC, Chatzarakis G, Katsikas S, Liatsis P (2008) Electricity demand loads modeling using autoregressive moving average (ARMA) models. Energy 33(9):1353–1360
6. Bowden N, Payne JE (2008) Short term forecasting of electricity prices for miso hubs: evidence from Arima–Egarch models. Energy Econ 30(6):3186–3197
7. Yan X, Chowdhury NA (2014) Mid-term electricity market clearing price forecasting: a multiple SVM approach. Int J Electr Power Energy Syst 58:206–214
8. Rendon-Sanchez JF, Menezes LM (2019) Structural combination of seasonal exponential smoothing forecasts applied to load forecasting. Eur J Oper Res 275(3):916–924

9. Asghar R, Fulginei FR, Quercio M, Mahrouch A (2024) Artificial neural networks for photovoltaic power forecasting: a review of five promising models. IEEE Access
10. He J, Li Y, Xu X, Wu D (2025) Energy consumption forecasting for oil and coal in China based on hybrid deep learning. PLoS ONE 20(1):0313856
11. Oreshkin BN, Dudek G, Pełka P, Turkina E (2021) N-beats neural network for mid-term electricity load forecasting. Appl Energy 293:116918
12. Salinas D, Flunkert V, Gasthaus J, Januschowski T (2020) Deepar: probabilistic forecasting with autoregressive recurrent networks. Int J Forecast 36(3):1181–1191
13. Lim B, Arık SÖ, Loeff N, Pfister T (2021) Temporal fusion transformers for interpretable multi-horizon time series forecasting. Int J Forecast 37(4):1748–1764
14. Saeed F, Paul A, Seo H (2022) A hybrid channel-communication-enabled CNN-LSTM model for electricity load forecasting. Energies 15(6):2263
15. Khan ZA, Ullah A, Haq IU, Hamdy M, Mauro GM, Muhammad K, Hijji M, Baik SW (2022) Efficient short-term electricity load forecasting for effective energy management. Sustain Energy Technol Assess 53:102337
16. Alsharekh MF, Habib S, Dewi DA, Albattah W, Islam M, Albahli S (2022) Improving the efficiency of multistep short-term electricity load forecasting via R-CNN with ML-LSTM. Sensors 22(18):6913
17. Zehao W, Zile C, Simin Y, Huanhuan D, Junling W, Ghadimi N (2025) Optimal economic model of a combined renewable energy system utilizing modified. Sustain Energy Technol Assess 74:104186
18. Zeng S, Liu C, Zhang H, Zhang B, Zhao Y (2025) Short-term load forecasting in power systems based on the Prophet–BO–XGBoost model. Energies 18(2):227
19. Kamalov F, Sulieman H, Moussa S, Avante Reyes J, Safaraliev M (2024) Powering electricity forecasting with transfer learning. Energies 17(3):626
20. Moradzadeh A, Moayyed H, Zare K, Mohammadi-Ivatloo B (2022) Short-term electricity demand forecasting via variational autoencoders and batch training-based bidirectional long short-term memory. Sustain Energy Technol Assess 52:102209
21. Bendaoud NMM, Farah N, Ahmed SB (2021) Comparing generative adversarial networks architectures for electricity demand forecasting. Energy Build 247:111152
22. Zhou Y, Wang M (2025) Empower pre-trained large language models for building-level load forecasting. IEEE Trans Power Syst https://doi.org/10.1109/TPWRS.2025.3548891
23. Demir E (2024) Electricity consumption forecasting with deep learning models. Master's thesis, Eskisehir Technical University
24. Chai T, Draxler RR (2014) Root mean square error (RMSE) or mean absolute error (MAE)? Arguments against avoiding RMSE in the literature. Geosci Model Dev 7(3):1247–1250
25. Torres JF, Martínez-Álvarez F, Troncoso A (2022) A deep LSTM network for the Spanish electricity consumption forecasting. Neural Comput Appl 34(13):10533–10545
26. Butt FM, Hussain L, Mahmood A, Lone KJ (2021) Artificial intelligence based accurately load forecasting system to forecast short and medium-term load demands. Math Biosci Eng 18(1):400–425
27. Almaleck P, Massucco S, Mosaico G, Saviozzi M, Serra P, Silvestro F (2024) Electrical consumption forecasting in sports venues: a proposed approach based on neural networks and ARIMAX models. Sustain Cities Soc 100:105019

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.