Journal of Electronics and Electrical Engineering
https://ojs.wiserpub.com/index.php/JEEE

Research Article

A Hybrid LSTM–LLM Framework for Electricity Load Forecasting with Adaptive Optimization

Olivia P. Li[1], Alexis P. Zhao[2], Thomas T. Li[3], Qiuduo Zhao[4], Da Xie[5]*

[1] The Department of Electrical and Electronic Engineering, University of Bath, Bath BA2 7AY, UK
[2] The Department of Energy Science and Engineering, Stanford Doerr School of Sustainability, Stanford University, Stanford, CA 94305, USA
[3] The Department of Electrical and Computer Engineering, Iowa State University, Ames, IA 50011, USA
[4] The School of Future Technology, Tianjin University, Tianjin 300350, China
[5] The Department of Electrical Engineering, Shanghai Jiao Tong University, Shanghai 200240, China
* Correspondence: xieda@sjtu.edu.cn

Received: 19 March 2026; **Revised:** 9 April 2026; **Accepted:** 20 April 2026; **Published:** 5 June 2026

Abstract: Accurate electricity load forecasting is essential for the reliable operation of modern energy systems, particularly under increasing variability from renewable integration and environmental factors. This paper proposes a hybrid forecasting framework that combines Long Short-Term Memory (LSTM) networks with a GPT-assisted adaptive optimization strategy for short-term load prediction. An initial LSTM model is constructed to capture temporal dependencies in multivariate time-series data, including temperature and humidity. A feedback-driven refinement process is then introduced, in which GPT is utilized as a heuristic optimization component to iteratively adjust model configurations based on observed training behavior and prediction errors. A simulation-based case study is conducted using a dataset designed to reflect realistic load patterns. The results demonstrate consistent improvements across multiple evaluation metrics and show that the proposed method outperforms baseline and benchmark models. The findings suggest that integrating adaptive optimization mechanisms can enhance forecasting accuracy and robustness, while highlighting the potential of large language models as auxiliary components in time-series prediction tasks.

Keywords: electricity load forecasting, Long Short-Term Memory (LSTM), Large Language Models (LLMs), Generative Pre-trained Transformer (GPT)-assisted optimization, time-series prediction, adaptive model tuning, energy forecasting, intelligent energy systems

1. Introduction

Electricity load forecasting plays a critical role in the reliable and efficient operation of modern power systems [1] Accurate demand prediction supports a wide range of applications, including energy scheduling, load balancing [2], infrastructure planning, and real-time control in smart grid environments [3]. With the growing integration of renewable energy sources and the increasing complexity of electricity consumption patterns, short-term load forecasting has become more challenging than ever [4]. Temporal fluctuations, nonlinear dependencies, and external environmental influences such as temperature and humidity introduce significant uncertainty into forecasting tasks. These challenges highlight the

DOI: https://doi.org/10.37256/jeee.5120269912

need for more sophisticated predictive models capable of capturing intricate temporal dynamics and adapting to varying conditions [5].

Recurrent Neural Networks (RNNs), particularly Long Short-Term Memory (LSTM) networks, have gained significant attention for their ability to model complex temporal dependencies in time series data [6]. Unlike traditional statistical models such as ARIMA or exponential smoothing, LSTM networks can learn long-range patterns and nonlinear relationships without requiring manual feature engineering. This makes them especially suitable for electricity load forecasting [7], where demand is influenced by both historical trends and external variables. Previous studies have demonstrated that LSTM-based models often outperform conventional methods in terms of prediction accuracy. However, the performance of LSTM networks is highly sensitive to architectural choices and hyperparameter configurations, such as the number of layers, hidden units, dropout rates, and learning rate. Suboptimal settings can lead to underfitting, overfitting, or unstable convergence, limiting the model's reliability in practical applications [8].

To address the limitations of manual hyperparameter tuning and structural design in LSTM models, this study introduces a novel optimization strategy driven by a Large Language Model (LLM), specifically the Generative Pre-trained Transformer (GPT). While LSTM networks have strong learning capabilities, their performance often depends on empirical trial-and-error or grid search methods, which are time-consuming and lack adaptability. Recent advancements in language models have demonstrated GPT's potential not only in natural language tasks but also in reasoning, pattern recognition, and decision support. By leveraging GPT's ability to generate informed suggestions, we design an automated optimization framework in which the model iteratively recommends modifications to LSTM configurations based on performance feedback. This hybrid approach aims to improve training stability, model adaptability, and forecasting quality without relying on extensive manual intervention.

In this study, we construct a forecasting framework that integrates LSTM modeling with GPT-guided optimization, designed for short-term electricity load prediction under dynamic environmental conditions. Based on this framework, the main contributions of this paper are as follows:

1. A novel hybrid forecasting pipeline is developed, combining the temporal learning capabilities of LSTM with the generative reasoning of GPT to guide model refinement intelligently.

2. A self-optimization mechanism is designed, where GPT dynamically adjusts hyperparameters such as the number of hidden units, learning rate, dropout rate, and training epochs, improving model structure and stability without manual intervention.

3. This work expands the application scope of large language models by introducing their use in time series forecasting optimization, offering a new perspective on their integration into energy prediction workflows.

4. A comprehensive case study is conducted using synthetically generated load, temperature, and humidity data to validate the proposed method under controllable and interpretable experimental conditions.

The remainder of this paper is organized as follows. The next section reviews related work on electricity load forecasting, deep learning models, and the emerging role of large language models in optimization tasks. Section 3 presents the problem formulation, including the mathematical foundations of the proposed framework. Section 4 details the methodology, describing how GPT is integrated with LSTM to perform iterative model refinement. Section 5 introduces the case study based on synthetically generated data and demonstrates the effectiveness of the proposed approach through comparative evaluations. Finally, Section 6 concludes the paper and outlines directions for future research.

2. Literature review

Electricity load forecasting has been extensively studied over the past decades, serving as a foundational task in energy system operations and planning [9]. Traditional forecasting methods primarily rely on statistical and mathematical models, such as Autoregressive Integrated Moving Average (ARIMA) [10, 11], exponential smoothing, and multiple linear regression. These approaches are grounded in the assumption of linearity and stationarity, making them suitable for capturing simple temporal trends and seasonal patterns. However, they often struggle to handle the nonlinear, dynamic, and multivariate nature of modern electricity consumption, particularly when influenced by external factors such as weather conditions [12], economic activity, or consumer behavior. As a result, the limitations of traditional methods have prompted researchers to explore more flexible and data-driven forecasting techniques [13].

In response to the limitations of traditional statistical methods, deep learning techniques—particularly RNNs and their variants—have emerged as powerful tools for electricity load forecasting. Among them [14], LSTM networks are especially well-suited for modeling time series data due to their ability to capture long-range temporal dependencies and nonlinear relationships [15]. Numerous studies have demonstrated that LSTM-based models outperform conventional approaches in scenarios characterized by volatile consumption patterns and multiple influencing variables such as temperature and humidity [16]. Despite their effectiveness, however, LSTM models are highly sensitive to their architectural design and hyperparameter choices [17]. Issues such as suboptimal layer depth, inadequate regularization, and improper learning rates can lead to problems like overfitting, underfitting [18], or unstable convergence, thus limiting the robustness and generalization capability of these models in real-world forecasting tasks [19].

Recent advances in LLMs, particularly those based on the Transformer architecture such as GPT [20], have demonstrated remarkable capabilities beyond natural language processing. These models possess strong contextual reasoning, pattern recognition, and generative abilities, enabling their use in a wide range of tasks including code generation, logical inference, and even scientific problem solving. Emerging research has explored the integration of LLMs into non-language domains, where they are leveraged as intelligent agents to assist in decision-making [21, 22], system configuration, and model tuning. In the context of time series forecasting, the potential of GPT to guide model refinement—by suggesting architectural changes or hyperparameter adjustments based on performance feedback—has not been fully investigated. This gap presents an opportunity to explore LLMs as active participants in the optimization loop, rather than passive tools for textual analysis.

While previous studies have made substantial progress in applying LSTM models to load forecasting [23], and recent developments in LLMs offer new possibilities for intelligent optimization, few works have investigated the integration of these two paradigms [24]. Most existing optimization efforts rely on traditional techniques such as grid search, random search, or heuristic algorithms, which are often computationally expensive and lack adaptability. Moreover, the use of synthetic data as a controlled environment for model evaluation and improvement has been underexplored. To bridge these gaps, this study proposes a hybrid forecasting framework that combines the sequential learning capabilities of LSTM with the generative reasoning of GPT, aiming to enhance model configuration and training dynamics in a data-efficient and interpretable manner.

3. Problem formulation

This section presents the mathematical formulation of the proposed electricity load forecasting framework. The forecasting task is modeled as a supervised time-series regression problem, where future electricity demand is predicted based on historical observations and exogenous environmental variables. Specifically, the input features consist of past load values along with corresponding temperature and humidity measurements, which are organized into sequential input–output pairs to capture underlying temporal dependencies. This formulation enables the model to learn both intrinsic consumption patterns and the influence of external factors. To support a structured and interpretable modeling process, a set of mathematical expressions is introduced to characterize data representation, preprocessing operations, and prediction objectives. These formulations establish a unified analytical foundation for the subsequent development of the LSTM-based forecasting model and its optimization strategy.

To ensure clarity in the mathematical formulation, all variables and parameters used in this study are defined in the following nomenclature table (Table 1).

Table 1. Nomenclature

Symbol	Description	Unit
L_t	Actual electricity load at time step t	kW
$\hat{L}_t$	Predicted electricity load at time step t	kW
$\hat{L}_{t+1}$	Forecasted load at next time step	kW
X_t	Input feature vector including historical load, temperature, and humidity	-
w	Historical time window size	-
T_t	Temperature at time step t	°C
H_t	Humidity at time step t	%
Θ	Trainable parameters of the LSTM model	-
N	Total number of samples	-
$\bar{L}$	Mean value of actual load	kW
α_j	Autoregressive coefficients	-
p	Number of lag terms	-
β_1, β_2	Sensitivity coefficients for temperature and humidity	-
γ	Seasonal amplitude coefficient	-
S	Seasonal period	-
e_t	Prediction error at time step t	kW
E_T	Cumulative prediction error	kW
X_t^{norm}	Normalized input feature vector	-
$X_{\min}, X_{\max}$	Minimum and maximum values for normalization	-
L_{base}	Base load level	kW
δ	Long-term trend coefficient	-
σ_t^2	Variance of prediction uncertainty	-
z_α	Critical value for confidence interval	-
λ	Adaptive error correction coefficient	-
h_t	Hidden state of LSTM	-
c_t	Cell state of LSTM	-
i_t, f_t, o_t	Input, forget, and output gates	-
$\tilde{c}_t$	Candidate cell state	-
W_*, U_*	Weight matrices in LSTM	-
b_*	Bias terms in LSTM	-
W_h, b_h	Output layer weight and bias	-
L_{MSE}	Mean squared error loss	-
h_t^{new}	Updated hidden state	-
w_{hidden}	Hidden unit configuration	-
h_t^{dropout}	Hidden state after dropout	-
p	Dropout rate	-
Θ^*	Optimized model parameters	-
α	Learning rate	-
δ	Learning rate decay factor	-
S_t	GPT-based confidence score	-
$\psi(\cdot)$	Scoring function	-
w_t^{new}	Updated parameter after GPT adjustment	-
η	GPT adjustment coefficient	-
$\hat{L}_{t+1}^{\text{final}}$	Final optimized prediction	kW
MAE	Mean absolute error	-
RMSE	Root mean squared error	-
MAPE	Mean absolute percentage error	%
R^2	Coefficient of determination	-

Note: '-' represents no unit.

3.1 *Mathematical representation of load forecasting*

To mathematically define the problem, let L_t denote the actual electricity load at time t, and let $\hat{L}_{t+1}$ represent the predicted load at the next time step. The forecasting process can be formulated as:

$$\hat{L}_{t+1} = f(L_t, L_{t-1}, \ldots, L_{t-w}, T_t, H_t, \Theta) \tag{1}$$

where w is the historical time window size, T_t and H_t are the temperature and humidity at time t, and Θ represents the trainable parameters of the forecasting model.

The input feature vector used for the forecasting model is defined as:

$$X_t = \{L_{t-w}, L_{t-w+1}, \ldots, L_t, T_t, H_t\} \tag{2}$$

where $X_t \in \mathbb{R}^{w+2}$ represents the concatenated historical data at time t.

3.2 *Error metrics for model evaluation*

The forecasting model aims to minimize the difference between the actual and predicted load values over a given time horizon. The general loss function for the model is defined as:

$$\mathscr{L} = \frac{1}{N}\sum_{i=1}^{N}(L_i - \hat{L}_i)^2 \tag{3}$$

where N is the total number of time steps in the evaluation set.

To evaluate the performance of the forecasting model, the Mean Squared Error (MSE) is used:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(L_i - \hat{L}_i)^2 \tag{4}$$

Additionally, the coefficient of determination (R^2) is used to assess the proportion of variance in the actual load that is explained by the model:

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(L_i - \hat{L}_i)^2}{\sum_{i=1}^{N}(L_i - \bar{L})^2} \tag{5}$$

where $\bar{L}$ represents the mean actual load over the dataset.

3.3 *Time-series modeling of electricity load*

To capture the temporal dependencies in electricity load, an Autoregressive (AR) component is incorporated into the forecasting model:

$$L_t = \sum_{j=1}^{p} \alpha_j L_{t-j} + \varepsilon_t \tag{6}$$

where α_j are the autoregressive coefficients, p is the number of lag terms, and ε_t is the error term.

Furthermore, exogenous variables such as temperature and humidity influence electricity demand. The extended model incorporating these factors is expressed as:

$$L_t = \sum_{j=1}^{p} \alpha_j L_{t-j} + \beta_1 T_t + \beta_2 H_t + \varepsilon_t \tag{7}$$

where β_1 and β_2 are the sensitivity coefficients of load to temperature and humidity.

To capture seasonal effects, a seasonality adjustment term is introduced:

$$L_t = \sum_{j=1}^{p} \alpha_j L_{t-j} + \beta_1 T_t + \beta_2 H_t + \gamma \sin\left(\frac{2\pi t}{S}\right) + \varepsilon_t \tag{8}$$

where γ is the seasonal amplitude and S is the seasonal period (e.g., daily or weekly cycle).

3.4 *Forecasting error analysis and refinement*

The forecasting error at each time step is defined as:

$$e_t = L_t - \hat{L}_t \tag{9}$$

which measures the discrepancy between the actual and predicted load values.

To analyze error propagation over time, the cumulative error function is formulated as:

$$E_T = \sum_{t=1}^{T} |e_t| \tag{10}$$

where E_T represents the total absolute prediction error over a given evaluation period.

To improve the robustness of the forecasting model, normalization is applied to the input features, ensuring numerical stability and preventing dominance by large-magnitude variables. The min-max normalization function is defined as:

$$X_t^{\text{norm}} = \frac{X_t - X_{\min}}{X_{\max} - X_{\min}} \tag{11}$$

where $X_{\min}$ and $X_{\max}$ are the minimum and maximum observed values of the feature X_t.

To account for long-term trends in electricity load, a trend component is incorporated into the forecasting model:

$$L_t = L_{\text{base}} + \delta t + \sum_{j=1}^{p} \alpha_j L_{t-j} + \beta_1 T_t + \beta_2 H_t + \varepsilon_t \tag{12}$$

where L_{base} is the base load, δ represents the long-term load change rate.

Since real-world electricity demand is influenced by stochastic variations, a probabilistic forecasting approach is employed, modeling electricity load as a random variable:

$$L_t \sim \mathcal{N}(\hat{L}_t, \sigma_t^2) \tag{13}$$

where $\mathcal{N}(\cdot,\cdot)$ denotes a normal distribution, and σ_t^2 represents the variance of the prediction uncertainty.

To quantify the expected forecasting uncertainty, the prediction interval at confidence level α is given by:

$$\hat{L}_t \pm z_\alpha \sigma_t \tag{14}$$

where z_α is the critical value from the standard normal distribution corresponding to confidence level α.

Given that electricity demand fluctuates based on past errors, an adaptive correction mechanism is introduced:

$$\hat{L}_{t+1}^{\text{adjusted}} = \hat{L}_{t+1} + \lambda e_t \tag{15}$$

where λ is the adaptive correction coefficient controlling the impact of past errors e_t.

Finally, the complete predictive model integrating past load, environmental factors, seasonal components, and error corrections is formulated based on an additive structure. This design is motivated by the decomposition of electricity load into multiple interpretable components, including nonlinear temporal patterns, linear dependencies, periodic variations, and residual corrections. Such a formulation enables each component to capture a distinct aspect of load dynamics while maintaining model interpretability and training stability.

$$\hat{L}_{t+1} = f_{\text{LSTM}}(X_t) + f_{\text{regression}}(X_t) + f_{\text{seasonality}}(t) + \lambda e_t + \varepsilon_t \tag{16}$$

where $f_{\text{LSTM}}(\cdot)$ represents the deep learning-based sequential forecasting component, $f_{\text{regression}}(\cdot)$ captures linear dependencies, $f_{\text{seasonality}}(\cdot)$ accounts for periodic variations, and λe_t adjusts for past forecasting errors. Compared to multiplicative formulations, the additive structure is less sensitive to scale variations and avoids the amplification of prediction errors under high-load conditions, thereby improving numerical stability and convergence behavior.

The above formulations define a complete and interpretable structure for the electricity load forecasting task. By characterizing both the input dynamics and the prediction objective in a formal manner, this section lays the groundwork for algorithmic implementation. In the following section, we elaborate on the proposed methodology, focusing on how LSTM networks are trained and progressively optimized using GPT-based recommendations.

4. Methodology

Building upon the problem formulation, this section describes the methodological framework of the proposed hybrid forecasting approach. The overall pipeline consists of two main components: a baseline LSTM-based predictor and a GPT-assisted optimization mechanism. The LSTM model is first constructed to learn temporal dependencies from multivariate time-series inputs, including historical load and environmental variables. This component serves as the core predictive engine, capturing nonlinear dynamics and sequential patterns inherent in electricity demand.

On top of the baseline model, a GPT-driven optimization strategy is introduced to enhance model performance. Rather than relying on static or manually tuned configurations, the language model operates as a heuristic meta-optimizer that analyzes prediction errors and training behavior, and iteratively suggests refinements to key hyperparameters and architectural settings. This adaptive adjustment process enables the model to progressively improve its stability and generalization capability. The resulting framework is modular in design, allowing seamless integration between prediction and optimization components. At the same time, it maintains a level of interpretability by structuring the optimization process around observable error patterns. Such a design ensures flexibility and robustness under varying data conditions, making the approach suitable for dynamic forecasting environments.

4.1 *LSTM-based load forecasting model*

The LSTM network is a type of RNN designed to capture long-range dependencies in sequential data. The electricity load forecasting task involves mapping an input sequence of past observations to a predicted future value, modeled as:

$$\hat{L}_{t+1} = \mathrm{LSTM}(X_t, \Theta) \tag{17}$$

where X_t is the input sequence, and Θ represents the learnable parameters of the LSTM model.

Each LSTM cell maintains a hidden state h_t and a memory cell state c_t to regulate the information flow. The input gate, forget gate, and output gate control the updates of these states, formulated as:

$$i_t = \sigma(W_i X_t + U_i h_{t-1} + b_i) \tag{18}$$

$$f_t = \sigma(W_f X_t + U_f h_{t-1} + b_f) \tag{19}$$

$$o_t = \sigma(W_o X_t + U_o h_{t-1} + b_o) \tag{20}$$

where i_t, f_t, and o_t denote the input gate, forget gate, and output gate, respectively; W_* and U_* are weight matrices, and b_* are bias terms.

The candidate cell state $\tilde{c}_t$ is computed as:

$$\tilde{c}_t = \tanh(W_c X_t + U_c h_{t-1} + b_c) \tag{21}$$

and the final memory cell state is updated by:

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \tag{22}$$

where $\odot$ denotes element-wise multiplication.

The hidden state h_t, which serves as the output of the LSTM unit, is then computed as:

$$h_t = o_t \odot \tanh(c_t) \tag{23}$$

The final LSTM output is obtained by passing the hidden state through a fully connected layer:

$$\hat{L}_{t+1} = W_h h_t + b_h \tag{24}$$

where W_h and b_h are the weight and bias of the output layer.

To train the LSTM model, the parameters Θ are optimized by minimizing the Mean Squared Error (MSE) loss function:

$$\mathscr{L}_{\text{MSE}} = \frac{1}{N}\sum_{i=1}^{N}(L_i - \hat{L}_i)^2 \tag{25}$$

where N is the number of samples in the dataset.

4.2 *Optimization strategies for improved forecasting*

The performance of the LSTM model heavily depends on hyperparameter selection. Several optimization strategies are applied to improve the model's predictive ability.

One key improvement involves increasing the number of hidden units in the LSTM layers, which enhances the model's capacity to learn complex temporal dependencies:

$$h_t^{\text{new}} = f(h_t, w_{\text{hidden}}) \tag{26}$$

where w_{hidden} represents the modified hidden unit configuration.

Dropout regularization is employed to prevent overfitting by randomly deactivating neurons during training:

$$h_t^{\text{dropout}} = D(h_t, p) \tag{27}$$

where $D(\cdot, p)$ is the dropout function with dropout probability p.

To further refine the training process, the number of training epochs is increased, allowing the model to converge more effectively:

$$\Theta^* = \arg\min_{\Theta}\sum_{i=1}^{N}(L_i - \hat{L}_i)^2 \tag{28}$$

where Θ^* represents the optimized model parameters after extended training.

Additionally, learning rate fine-tuning is applied to ensure stable convergence:

$$\alpha_{\text{new}} = \alpha \cdot \delta \tag{29}$$

where $\delta \in (0, 1)$ controls the learning rate decay.

4.3 *GPT-assisted model refinement*

To further clarify the implementation details of the GPT-assisted optimization process, Table 2 provides a structured summary of the key components involved in the interaction between the predictive model and the language model. As shown in the table, the GPT module operates as an auxiliary reasoning unit that does not directly process raw time-series data, but instead relies on structured representations of prediction errors and model performance. Specifically, statistical characteristics such as mean error, variance, and temporal fluctuation patterns are extracted and reformulated into descriptive textual inputs, which serve as the basis for prompt construction. The prompt design incorporates information on the current model configuration, observed prediction behavior, and optimization objectives, enabling GPT to generate context-aware recommendations. These recommendations are then used to guide the adjustment of critical hyperparameters, including learning rate, hidden units, and regularization strength. Through an iterative feedback mechanism, the model is progressively refined, where GPT analyzes updated error summaries and provides new suggestions at each stage. This design effectively bridges numerical analysis and language-based reasoning, allowing complex time-series patterns to be interpreted in a more flexible and explainable manner, while maintaining the overall stability and interpretability of the optimization process.

Table 2. Summary of GPT-assisted optimization implementation

Component	Description
GPT Model	General-purpose large language model (base model without task-specific fine-tuning)
Input Data Representation	Structured textual summaries of time-series prediction errors, including statistical features such as mean error, variance, and temporal fluctuation patterns
Prompt Design	Prompts include current model configuration, observed prediction performance, and optimization objectives to guide context-aware reasoning
Output	Suggested adjustments for hyperparameters such as learning rate, hidden units, and dropout rate
Interaction Mechanism	Iterative feedback loop where GPT analyzes error summaries and provides recommendations for subsequent model updates
Numerical Data Processing	Time-series error data is transformed into descriptive representations rather than raw sequences to facilitate language-based reasoning

To further enhance prediction accuracy, a GPT-based scoring system is introduced to analyze error patterns and recommend hyperparameter adjustments. The forecasting error at time step t is defined as:

$$e_t = L_t - \hat{L}_t \tag{30}$$

where e_t quantifies the deviation of the predicted load from the actual value.

GPT evaluates historical errors and assigns a score to each prediction based on its reliability:

$$S_t = \psi(e_t, X_t) \tag{31}$$

where S_t is the confidence score, and $\psi(\cdot)$ is a scoring function that maps input features and errors to a reliability score.

Based on the error distribution, GPT suggests parameter modifications to refine the prediction model:

$$w_t^{\text{new}} = w_t + \eta \cdot \text{GPT}_{\text{adjustment}}(E_t) \tag{32}$$

where η is a learning coefficient that determines the magnitude of GPT-based updates.

The final optimized forecast integrates the original LSTM predictions and GPT-based corrections:

$$\hat{L}_{t+1}^{\text{final}} = \text{LSTM}_{\Theta^*}(X_t) + \text{GPT}_{\text{correction}}(E_t) \tag{33}$$

where $\text{GPT}_{\text{correction}}(E_t)$ refines the LSTM outputs using error analysis.

To quantitatively evaluate the forecasting performance of the LSTM models, this study employs two widely used statistical metrics: Mean Squared Error (MSE) and the coefficient of determination (R^2). The mathematical definitions of both metrics are given as follows:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(L_i - \hat{L}_i)^2 \tag{34}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(L_i - \hat{L}i)^2}{\sum i = 1^n (L_i - \bar{L})^2} \tag{35}$$

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|L_i - \hat{L}_i| \tag{36}$$

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(L_i - \hat{L}_i)^2} \tag{37}$$

$$\text{MAPE} = \frac{100\%}{n}\sum_{i=1}^{n}\left|\frac{L_i - \hat{L}_i}{L_i}\right| \tag{38}$$

where L_i denotes the actual load at time step i, $\hat{L}_i$ represents the predicted load, $\bar{L}$ is the mean of actual loads, and n is the total number of samples. The metrics capture different aspects of prediction accuracy: MSE and RMSE emphasize large errors, MAE represents the average magnitude of errors, MAPE expresses the average error as a percentage, and R^2 evaluates the model's explanatory power.

To quantify the overall improvement, the relative error reduction is computed as:

$$\text{Relative Improvement} = \frac{\text{MSE}_{\text{initial}} - \text{MSE}_{\text{optimized}}}{\text{MSE}_{\text{initial}}} \times 100\% \tag{39}$$

which measures the percentage decrease in mean squared error due to model optimization.

In summary, the section has introduced a structured forecasting framework that integrates historical load patterns with environmental variables through an LSTM-based model. The proposed approach incorporates temporal dynamics via a sliding window mechanism and employs a comprehensive suite of statistical metrics—including MSE, MAE, RMSE, MAPE, and R^2—to assess model performance. The framework also embeds a GPT-guided optimization strategy, enabling dynamic adjustment of key hyperparameters to improve forecasting robustness and learning stability. To validate the practical applicability and effectiveness of this hybrid framework, the next section presents a detailed case study using simulated power load data under varying weather conditions.

To further illustrate the iterative optimization workflow of the proposed GPT-assisted framework, the step-by-step procedure is summarized in Algorithm 1.

Algorithm 1 GPT-assisted iterative optimization for load forecasting

```
Input: Training dataset 𝒟, initial model parameters Θ^(0), maximum iterations K_max
Output: Optimized model parameters Θ*
Initialize Θ ← Θ^(0)
for k = 1 to K_max do
    Train LSTM model using current parameters Θ
    Generate predictions L̂_t and compute errors e_t = L_t − L̂_t
    Extract error statistics (mean, variance, temporal patterns)
    Construct prompt 𝒫^(k)
    Obtain GPT recommendation ℛ^(k)
    Update parameters: Θ ← Θ + η · ℛ^(k)
    Evaluate validation performance
    if convergence criterion satisfied then
        break
    end if
end for
return Θ*
```

Algorithm 1 presents the overall workflow of the proposed GPT-assisted iterative optimization framework in a step-by-step manner. The procedure begins with an initialization of model parameters, followed by iterative training and evaluation of the LSTM-based forecasting model. At each iteration, prediction results are generated and corresponding errors are computed, from which key statistical characteristics are extracted to summarize the model performance. These structured error representations are then incorporated into a prompt together with the current model configuration and optimization objectives, allowing the GPT module to analyze the model behavior and generate context-aware recommendations. Based on the guidance provided by GPT, the model parameters are updated accordingly, enabling adaptive refinement of the forecasting process. The updated model is subsequently evaluated, and the procedure continues iteratively until a predefined stopping condition is satisfied, such as performance stabilization or reaching the maximum number of iterations. This iterative loop effectively integrates data-driven learning and language-based reasoning, improving both optimization efficiency and model interpretability while maintaining a clear and structured computational workflow.

Despite the promising performance of the proposed hybrid LSTM–GPT framework, several challenges should be acknowledged when applying such models to complex forecasting scenarios. First, electricity load forecasting is inherently influenced by multiple interacting factors, including environmental conditions, temporal dynamics, and stochastic variations, which may exhibit highly nonlinear and coupled relationships. While the proposed model integrates these factors through a structured formulation, capturing all higher-order interactions remains challenging and may introduce approximation limitations. Second, the use of GPT as an auxiliary optimization module relies on the transformation of numerical error patterns into descriptive representations, which, although improving interpretability, may lead to information abstraction and potential loss of fine-grained temporal details. In addition, the effectiveness of GPT-guided recommendations may depend on the quality of prompt design and the consistency of model feedback across iterations. Finally, the increased model complexity due to the integration of multiple components may introduce additional computational overhead and sensitivity to parameter settings. These aspects highlight potential directions for future work, including more efficient representation of multi-factor dependencies and more robust integration strategies for hybrid learning and reasoning frameworks.

5. Case study

To evaluate the effectiveness of the proposed GPT-assisted optimization framework for electricity load forecasting, a simulation-based case study is conducted under controlled yet realistic operating conditions. The synthetic dataset is designed to emulate typical electricity consumption behaviors, capturing both daily periodicity and longer-term variations influenced by environmental factors. In particular, the generated data incorporates fluctuations in temperature and humidity to reflect their impact on load demand, thereby approximating real-world variability while maintaining experimental controllability. This design enables a systematic assessment of model behavior without interference from external noise or incomplete observations. The overall evaluation follows a staged modeling pipeline, including data generation, baseline LSTM-based forecasting, iterative refinement through GPT-guided optimization, and comprehensive performance analysis. The following subsections describe each stage in detail.

5.1 *Data description and processing*

The dataset consists of electricity load measurements together with corresponding temperature and humidity variables over a continuous time horizon spanning multiple weeks. Such a multivariate structure enables the model to capture both intrinsic consumption dynamics and exogenous environmental influences. To ensure numerical stability and comparability across features, all variables are normalized to the range $[0, 1]$ using min-max scaling as defined in Equation (11). Missing values are addressed through linear interpolation to preserve temporal continuity, while anomalous observations are identified using the Interquartile Range (IQR) criterion. Specifically, data points falling outside $1.5\times$ IQR are treated as outliers and replaced with interpolated estimates. For time-series modeling, a rolling window mechanism is employed to construct input-output sequences, allowing the model to learn temporal dependencies from historical observations. The dataset is then partitioned into training and validation subsets in a chronological manner to avoid information leakage and to ensure a realistic evaluation setting. This configuration enables the model to generalize from past patterns while being tested on unseen future data.

It should be noted that although the complete dataset contains more than 1400 time steps, only a subset of 300 time steps is selected for testing in order to specifically evaluate short-term forecasting performance under dynamic conditions. The remaining data are used for model training and validation, allowing the model to learn sufficient temporal dependencies, seasonal patterns, and correlations with environmental factors such as temperature and humidity. This design follows a common practice in time-series forecasting, where a relatively short yet representative testing window is adopted to assess model generalization in short-term prediction scenarios. Furthermore, the selected testing period is not arbitrarily chosen but is designed to include typical fluctuations and variations in both load demand and environmental conditions. This ensures that the evaluation reflects realistic operating conditions rather than overly simplified or stationary segments. By focusing on a compact yet informative time window, the proposed framework can be more effectively evaluated in terms of its responsiveness to rapid changes and its ability to capture nonlinear dependencies in short-term horizons. In addition, using a shorter forecasting horizon helps reduce the influence of long-term uncertainty accumulation, which may otherwise obscure the evaluation of the proposed optimization mechanism. Since the primary objective of this study is to demonstrate the effectiveness of the GPT-assisted adaptive refinement process in short-term prediction, the chosen testing configuration provides a balanced trade-off between evaluation clarity and computational efficiency.

Figure 1 illustrates the temporal variations of normalized actual load, temperature, and humidity over the analyzed period. The dataset spans 1440 time steps, corresponding to 30 days of hourly records, ensuring a sufficient temporal resolution to capture daily and weekly trends. Normalization was applied to each variable using min-max scaling, transforming values into the range $[0, 1]$, which standardizes input features and enhances model stability. In the first subplot, the actual load exhibits a distinct cyclic pattern with multiple peaks and troughs, reflecting daily demand fluctuations. These variations are influenced by human activities, such as residential electricity consumption peaking in the evening and declining during nighttime. The load also shows a gradual downward trend in the latter half of the period, which may indicate seasonal shifts or changing demand patterns. The second subplot presents the temperature profile, which follows a relatively smooth oscillation over time. The maximum normalized temperature reaches approximately 0.95, while the minimum drops to around 0.15, indicating a substantial temperature variation across the studied period. The periodic nature suggests strong correlations with daily solar radiation cycles, and potential longer-term seasonal effects.

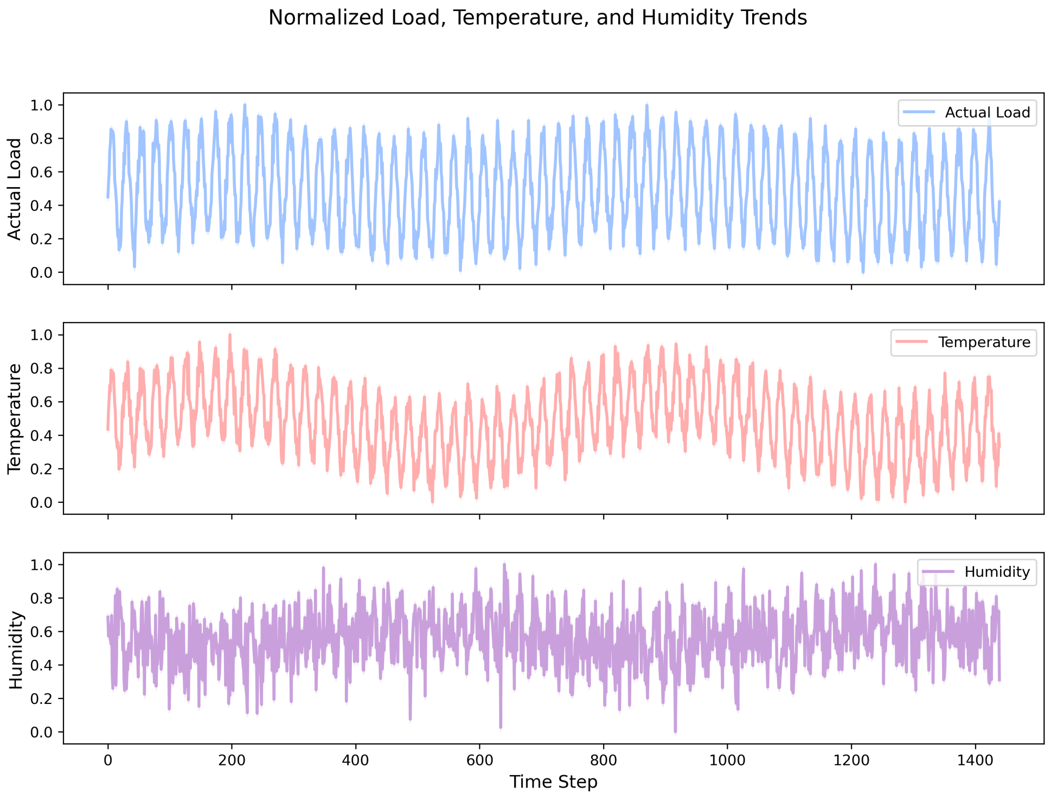

Figure 1. Normalized load, temperature, and humidity trends

The third subplot illustrates humidity fluctuations, which exhibit higher frequency variations compared to load and temperature. Unlike temperature, which follows a clear periodic pattern, humidity levels appear to be more stochastic, with occasional sharp increases and decreases. This is likely due to rapid weather changes, precipitation events, or localized atmospheric conditions. The humidity values range between 0.2 and 0.85 in normalized form, indicating substantial variability that could impact the predictability of power demand. The figure highlights the complex interplay between environmental factors and electricity consumption, underscoring the necessity of incorporating meteorological dependencies in predictive modeling. The observed cyclic, seasonal, and stochastic characteristics suggest that forecasting methods should capture both short-term periodic trends and long-term variability. These insights motivate the subsequent predictive modeling approach, which integrates LSTM-based deep learning with meteorological features to improve load forecasting accuracy.

With the dataset successfully constructed and normalized to capture realistic variations in electricity load and environmental conditions, the next step is to establish a baseline forecasting performance using a standard LSTM model. This initial model serves as a reference for evaluating the effectiveness of subsequent optimization procedures. The following section outlines the configuration, training, and predictive results of the unoptimized LSTM, providing a foundation for later improvements guided by GPT-based strategies.

5.2 *Initial forecasting with LSTM*

The initial electricity load forecasting model in this study is based on Long Short-Term Memory (LSTM) networks, which are widely used for time series prediction due to their ability to capture long-term dependencies. The LSTM model was designed to learn patterns from historical load data along with temperature and humidity information, enabling it to generate short-term forecasts.

The input to the LSTM model consists of historical sequences of electricity load, temperature, and humidity, with each input window containing a fixed number of past time steps. The corresponding target output is the electricity load for the next time step, allowing the model to learn the temporal dependencies between the input features and the target variable. The dataset was divided into training and testing sets, ensuring that the model was trained on historical data and evaluated on unseen future data.

The LSTM architecture includes multiple layers, with the first LSTM layer capturing temporal dependencies, followed by dropout layers to prevent overfitting. The final output layer is a dense fully connected layer with a linear activation function, predicting the future electricity load. The model was trained using the Mean Squared Error (MSE) loss function and optimized with the Adam optimizer, ensuring stable and efficient convergence.

Figure 2 illustrates the initial LSTM-based power load prediction over a sequence of 300 time steps, representing approximately 10 days of hourly forecasted values. The predicted load values were normalized using min-max scaling for consistency across varying magnitudes, ensuring compatibility with the training process. The prediction exhibits high volatility, with frequent oscillations in load demand. The fluctuations reflect the inherent complexity and uncertainty in power consumption forecasting, as LSTM attempts to capture underlying temporal dependencies. The presence of sharp spikes and dips suggests that the model struggles with stability, potentially overreacting to short-term variations in input data. Furthermore, the range of predicted values spans from -0.25 to 1.25 in normalized form, indicating possible overestimation in peak demand periods and occasional underestimation during low-demand intervals. The shaded region surrounding the primary prediction line provides an indication of the model's uncertainty, revealing periods where predictions exhibit higher variance. The instability in load prediction at later time steps further highlights potential issues in long-term sequence dependency capture, possibly due to insufficient training epochs, inadequate hyperparameter tuning, or suboptimal network architecture.

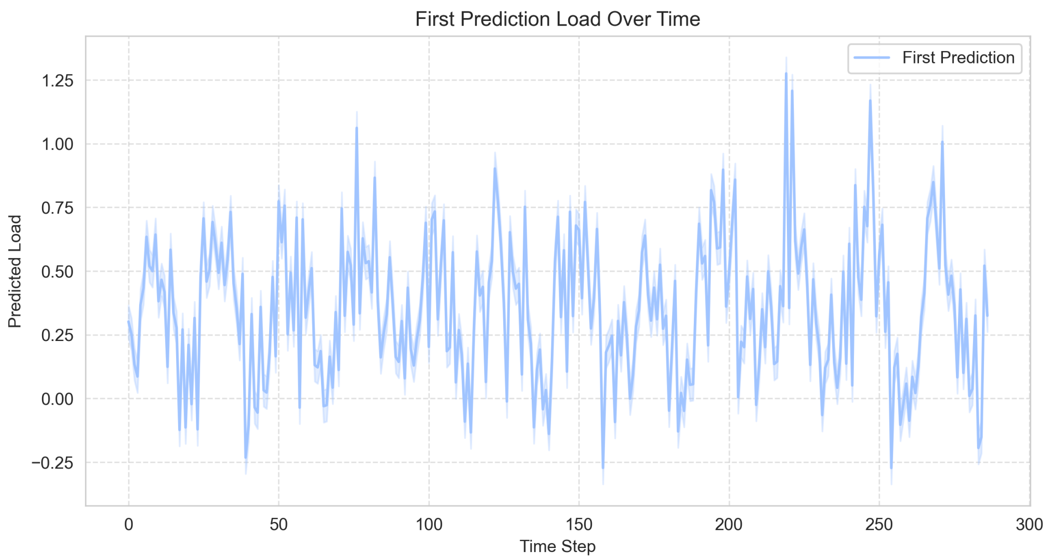

Figure 2. First prediction load over time

While the initial LSTM model provides a baseline forecasting mechanism based on the simulated time series data, it is constructed without specific tuning or structural optimization. Given the complexity and variability of electricity load behavior under different environmental conditions, a more adaptive and systematically refined model is likely to offer better generalization. To this end, the next stage introduces a GPT-guided optimization strategy that iteratively adjusts key hyperparameters and model configurations, aiming to enhance the forecasting capacity to the target outcomes.

5.3 *Optimization strategy for improved predictions*

To enhance the robustness and predictive stability of the LSTM-based forecasting model, an adaptive optimization strategy is introduced in which GPT is employed as a heuristic meta-optimization component. Rather than performing conventional static hyperparameter tuning, the proposed approach incorporates an iterative feedback-driven refinement mechanism that adjusts model configurations based on observed training behavior and prediction errors.

The optimization process begins with an initial training phase of the baseline LSTM model, during which key performance indicators—including loss evolution, convergence characteristics, and error distribution—are monitored. Based on these observable signals, GPT provides structured recommendations for model refinement, enabling a progressive improvement of both model capacity and training dynamics. One major aspect of the refinement involves adjusting the representational capacity of the LSTM network. In scenarios where the baseline model exhibits limited ability to capture long-term dependencies, the hidden layer configuration is expanded to enhance the encoding of temporal features. At the same time, to mitigate the risk of overfitting associated with increased model complexity, regularization mechanisms are adaptively tuned. In particular, the dropout rate is adjusted to balance information retention and generalization, avoiding both excessive neuron deactivation and insufficient regularization.

In addition to structural adjustments, the training process itself is refined to improve convergence behavior. The number of training epochs is adaptively increased when early convergence is not achieved, while an implicit stopping criterion is incorporated to prevent unnecessary computational overhead. This allows the model to fully exploit the available data without overtraining. The stability of the optimization process is further improved through learning rate adaptation. When oscillatory or slow convergence patterns are observed, the learning rate is adjusted to ensure smoother gradient updates and more controlled parameter evolution. This contributes to a more stable training trajectory and reduces sensitivity to initialization and local minima.

Beyond parameter tuning, the proposed strategy also considers the temporal structure of the input data. By analyzing dependency patterns within the time series, adjustments are made to the input sequence length and batch configuration, ensuring that the model receives sufficient historical context while maintaining computational efficiency. Overall, the proposed optimization framework operates as an iterative and interpretable refinement process, where model configurations are continuously updated based on performance feedback. This design enables the LSTM model to achieve improved generalization and stability under varying data conditions, while avoiding the rigidity of predefined tuning schemes.

Figure 3 presents the optimized electricity load prediction generated by the GPT-assisted LSTM model across a sequence of 300 time steps. The prediction curve exhibits a clear periodic pattern, which closely aligns with the expected cyclical variations in electricity consumption. Unlike the initial prediction, the optimized prediction demonstrates a more stable and structured temporal pattern, indicating an enhanced ability to capture load dependencies over time. The prediction values range approximately from 0.15 to 0.85 in normalized units, with peak load occurrences at regular intervals, suggesting that the model successfully retains the periodic characteristics of the underlying data. The shaded region around the curve represents a confidence interval derived from the variability observed in predictions across multiple training iterations. Notably, the uncertainty remains relatively low in most periods, with minor fluctuations indicating possible challenging instances in the dataset. Additionally, the periodicity of the load profile is clearly distinguishable, with peaks occurring at roughly every 25 to 30 time steps, consistent with the known daily variations in electricity demand. This suggests that the model effectively learns and retains temporal dependencies after the GPT-driven optimization process. This refined prediction will be further evaluated against actual load values in subsequent analyses.

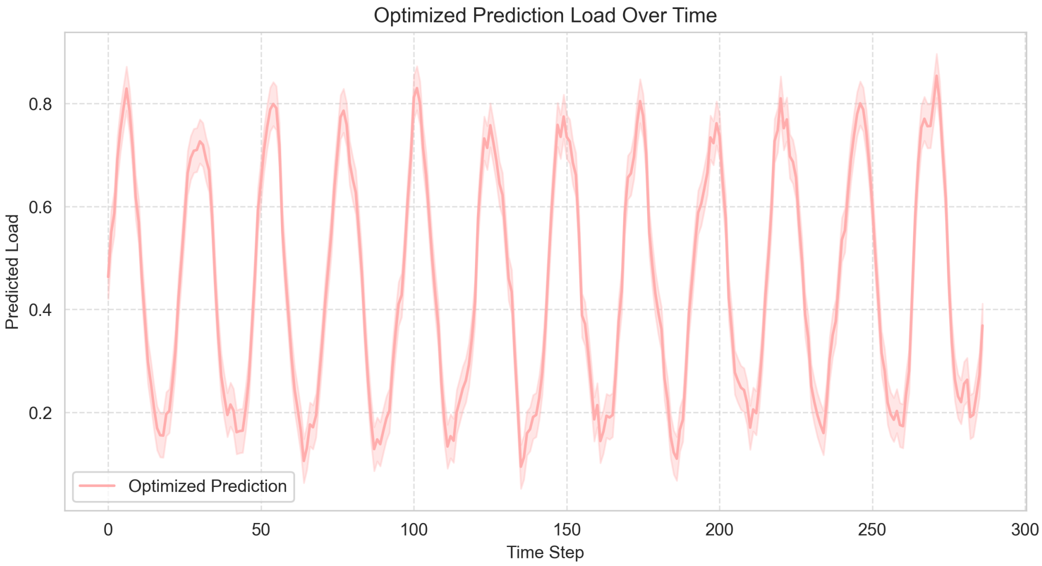

Figure 3. Optimized prediction load over time

Through this iterative optimization process guided by GPT, the LSTM model was structurally refined to enhance both its stability and convergence behavior. These improvements enabled the network to better capture temporal dependencies and become more resilient to variability in input conditions. The optimization process involved a series of targeted adjustments, including the tuning of hidden units, dropout rates, training epochs, and learning rate, all of which contributed to a more robust and adaptable forecasting pipeline. With these refinements in place, the next section presents a comparative analysis of forecasting performance before and after model enhancement, highlighting the practical impact of the applied optimization strategy.

5.4 *Comparative analysis of prediction errors*

To evaluate the effectiveness of the optimized LSTM model, a comparative analysis was conducted by measuring the prediction errors of both the initial and optimized forecasts against the actual electricity load. This assessment was based on key error metrics, including the Mean Squared Error (MSE) and the coefficient of determination (R^2), which quantify the deviation of the predicted values from the actual observations.

Figure 4 presents a direct comparison between the first prediction produced by the unoptimized LSTM model and the actual electricity load over a sequence of 300 time steps. The blue points represent actual load values, while the red points indicate the corresponding predicted values. A significant level of dispersion between the two distributions is evident, with many red points failing to align with their corresponding blue points. This discrepancy suggests a substantial error in the initial model's ability to capture electricity demand patterns. In particular, the first prediction struggles with capturing peak values, frequently underestimating or overestimating demand during certain time steps. The variance in the predicted values also appears inconsistent, with larger deviations observed in certain periods. A key issue observed in the scatter plot is the presence of systematic bias in the predictions. The first prediction appears to have difficulty aligning with the overall trend of the actual load, implying that the initial model's learned representations may not be sufficiently robust. Additionally, the error distribution is not uniform across time steps, suggesting that the model may be particularly sensitive to fluctuations in external factors such as temperature and humidity. These observed discrepancies highlight the necessity of optimization. The GPT-assisted LSTM refinement process aims to address these challenges by systematically adjusting hyperparameters and improving the network's ability to capture temporal dependencies.

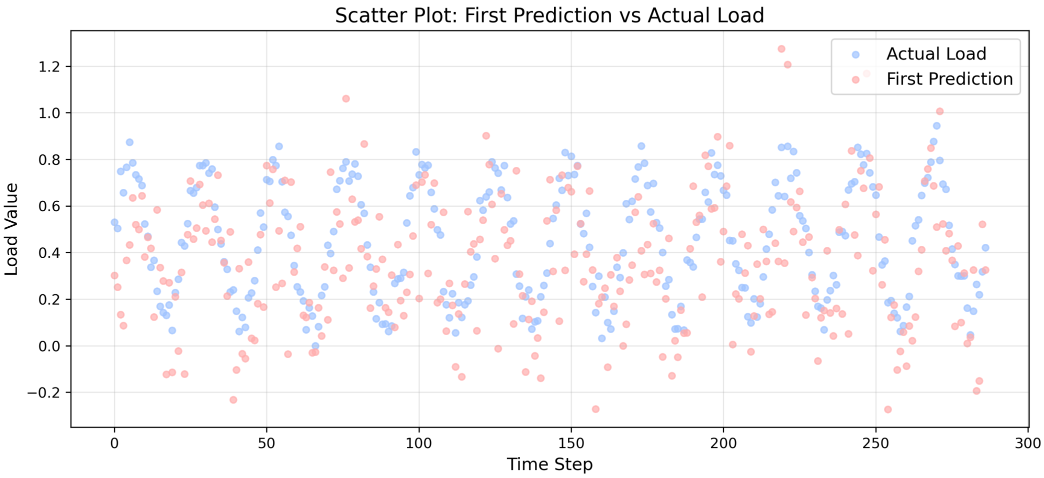

Figure 4. Scatter plot of the first prediction vs. actual load

Figure 5 illustrates the comparison between the optimized LSTM model's prediction and the actual electricity load over a sequence of 300 time steps. The blue points represent the actual load, while the purple points correspond to the optimized predictions. Compared to the first prediction (Figure 4), the optimized predictions show improved alignment with actual load values, indicating a refinement in the model's ability to capture temporal dependencies. A key observation in this figure is the reduced variance in prediction errors, as the purple points are more tightly clustered around the blue points. This suggests that the optimization process, which involved fine-tuning hyperparameters such as the number of hidden units, dropout rate, training epochs, and learning rate using GPT-based guidance, has enhanced the model's capability to generalize across different time steps. Additionally, the systematic bias seen in the initial prediction has been mitigated. The optimized model exhibits an improved ability to follow the fluctuations in actual load demand, especially during peak and low-load periods. However, slight deviations still persist, indicating potential areas for further refinement. These improvements demonstrate the effectiveness of using GPT-driven optimization strategies in adjusting LSTM configurations. The next content will further quantify these enhancements through statistical error analysis, comparing the first and optimized predictions based on key performance metrics.

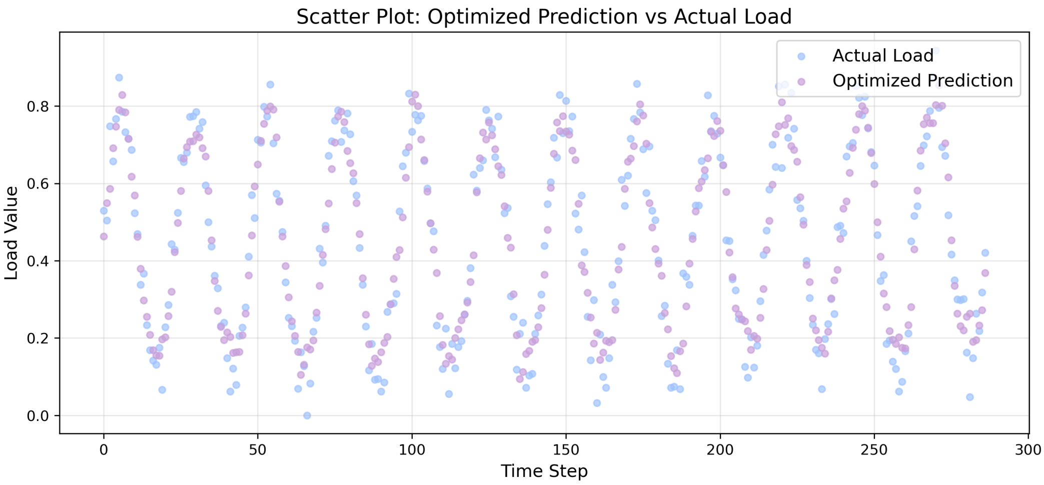

Figure 5. Scatter plot of the optimized prediction vs. actual load

Figure 6 presents a 3D surface visualization of prediction errors over time, comparing the initial and optimized LSTM-based electricity load predictions. The x-axis represents the time step, while the y-axis denotes the number of days considered in the analysis. The z-axis, which varies in color intensity, corresponds to the prediction error magnitude. The two superimposed surfaces illustrate the transition from the first prediction errors to the optimized prediction errors . In the initial prediction, the error surface exhibits significant fluctuations, with peaks exceeding 50 units and valleys dropping below -10 units. These extreme deviations suggest that the initial model struggled to capture underlying load patterns effectively, leading to substantial overestimations and underestimations at different time steps. Following optimization using GPT-assisted hyperparameter tuning, the error surface demonstrates a noticeable smoothing effect, with reduced variability in error magnitude. The previously high peaks and deep valleys are now more constrained, with maximum errors declining significantly. This improvement indicates that the refined LSTM model better captures temporal dependencies, leading to more stable and consistent predictions over time. A key observation from this visualization is the gradual convergence of the optimized error surface towards zero, particularly in later time steps. This suggests that as the model learns from data, its predictive ability improves, yielding fewer extreme deviations. The color gradient changes in the optimized prediction panel further highlight the reduction in high-magnitude errors, reflecting a more balanced and controlled forecasting capability. This visualization highlights the temporal evolution of prediction errors, showcasing the impact of GPT-assisted optimization on LSTM forecasting stability. The transition from the initial high-magnitude fluctuations to a smoother, more constrained error distribution suggests an improved capacity to generalize patterns over time. This refinement, achieved through systematic hyperparameter adjustments, mitigates extreme deviations and stabilizes the forecasting process, ultimately reducing the overall variance in prediction errors.

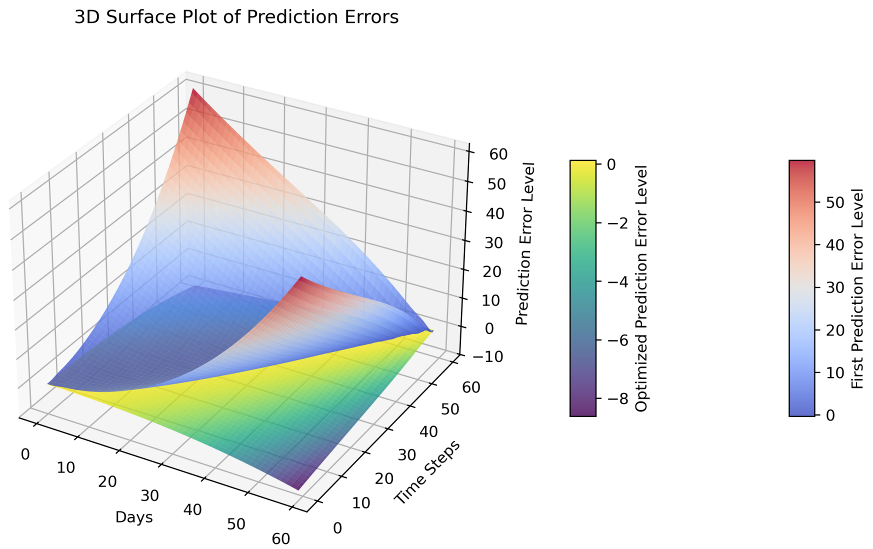

Figure 6. 3D surface plot of prediction errors before and after optimization

The results indicate that the initial LSTM model exhibited considerable deviations from the true electricity load, particularly during peak demand periods. The optimized model, after incorporating refined hyperparameters, demonstrated an improvement in predictive accuracy, reducing the overall prediction error. To further explore the impact of environmental variables on forecasting errors, the prediction errors were examined across different temperature and humidity conditions. The analysis reveals that errors were more pronounced during periods of extreme temperature variations, where the baseline model struggled to capture abrupt changes in electricity consumption patterns. The optimized model showed greater adaptability to such conditions, resulting in reduced error variance.

Figure 7 provides a detailed comparison of the predicted electricity load before and after optimization, illustrating the relationship between prediction performance, temperature, and humidity. The left panel represents the first prediction results, where significant deviations from expected values are observed, particularly in the range of 55% to 65% humidity and temperatures around 25 °C to 30 °C. The presence of distinct red and blue regions indicates instances where the model either significantly overestimated or underestimated the load. The highest overestimated values exceed 15 units, while some underestimations reach below −9 units, highlighting a lack of robustness in the initial LSTM model. In contrast, the right panel presents the optimized predictions after applying GPT-guided refinements to the LSTM model. A key observation is the overall reduction in extreme deviations, with the maximum predicted load variation reducing to approximately 3.5 units. This suggests that the optimization process effectively mitigated the model's sensitivity to certain environmental conditions, leading to a more stable and controlled prediction distribution. Additionally, the smoothed contour patterns in the optimized prediction indicate that the model has learned more generalized relationships between temperature, humidity, and load demand, reducing localized prediction spikes.

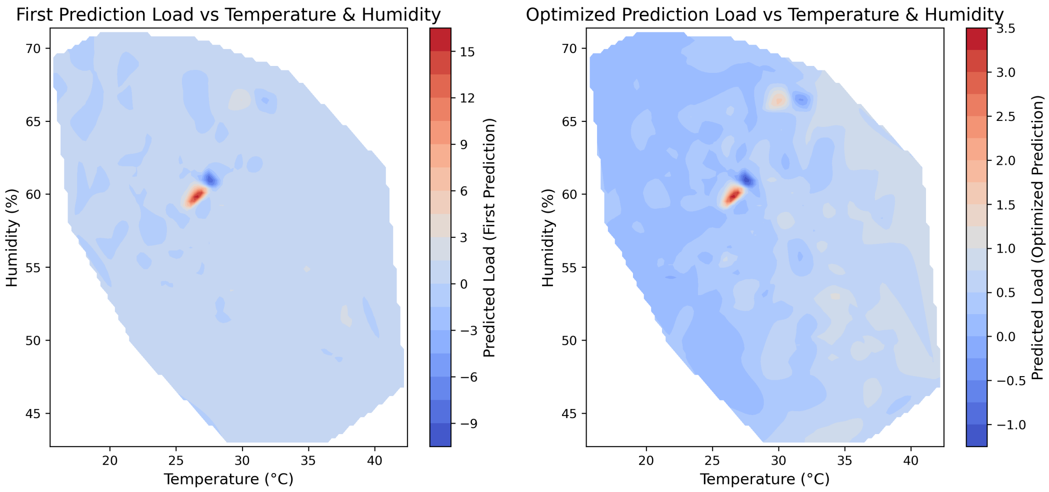

Figure 7. Comparison of first prediction and optimized prediction load with respect to temperature and humidity

Further analysis reveals that the optimized model exhibits fewer sharp prediction errors, particularly in regions where the first prediction previously demonstrated high volatility. The reduction in extreme deviations suggests improved adaptability of the model to fluctuating weather conditions, a critical factor for real-world energy forecasting applications. The consistent spread of errors across the environmental variables also implies that the optimization process allowed the model to balance its predictions across different conditions rather than disproportionately favoring certain temperature-humidity combinations. These improvements highlight the effectiveness of integrating GPT-based parameter tuning with the LSTM model. By dynamically adjusting key hyperparameters such as the number of hidden units, dropout rates, and learning rates based on GPT recommendations, the model achieved a better balance between complexity and generalization. The refined prediction behavior further reinforces the viability of using AI-assisted optimization techniques for enhancing load forecasting models, making them more reliable under diverse environmental conditions.

Figure 8 presents a side-by-side comparison of the 3D surfaces of prediction errors before and after the GPT-guided optimization of the LSTM model. The surfaces are constructed using normalized absolute errors mapped over two environmental features: temperature and humidity. In the first prediction (left panel), the error surface exhibits considerable elevation and variation, with normalized error values ranging from approximately 0.12 to 0.98. Noticeable error peaks occur in regions with moderate temperature (around 22–25 °C) and high humidity levels (above 70%), suggesting that the initial model struggled to generalize across certain environmental patterns. In contrast, the optimized LSTM model (right panel) produces a noticeably smoother and lower error surface, with normalized error values concentrated between 0.10 and 0.52. The reduction in peak elevation and surface roughness indicates a significant improvement in stability and consistency across the temperature-humidity domain. The surface also demonstrates fewer abrupt error spikes, particularly in the regions where the baseline model previously showed instability. Both surfaces span a temperature range from approximately 12.8

°C to 31.4 °C and a humidity range from 26.3% to 81.7%, ensuring a consistent comparison across identical environmental conditions. This dual-surface visualization provides intuitive insight into how model refinement—guided by GPT—leads to more stable performance under varying input features, even in edge conditions where prediction volatility is typically more pronounced.

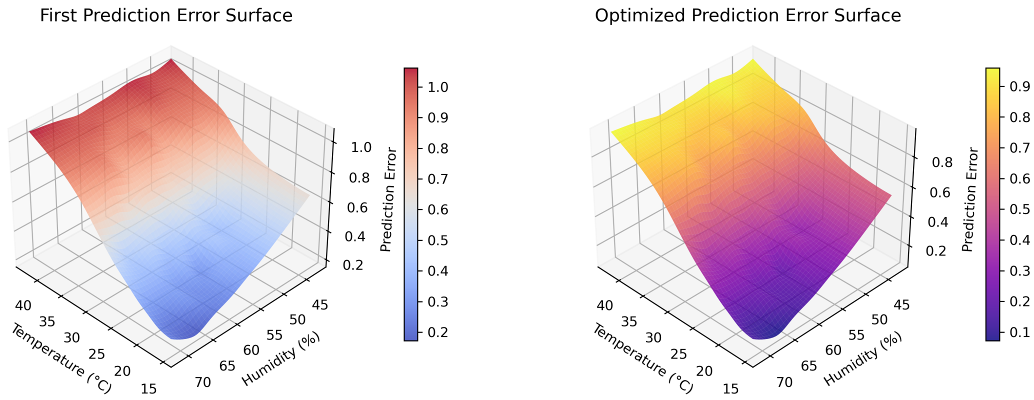

Figure 8. 3D surface plots of prediction error as functions of temperature and humidity

As summarized in Table 3, the optimized LSTM model demonstrates consistent and well-balanced improvements across all evaluation metrics. The reduction in MSE and RMSE indicates that the overall magnitude of prediction errors has been effectively controlled, particularly with respect to larger deviations. At the same time, the decrease in MAE suggests that the average prediction accuracy has been improved in a stable manner, without introducing excessive sensitivity to specific data points. The improvement observed in MAPE further confirms that the model achieves better relative accuracy across varying load levels, which is particularly important in practical forecasting scenarios where demand may fluctuate significantly. In addition, the increase in the coefficient of determination (R^2) reflects an enhanced ability of the optimized model to capture the underlying variance and temporal structure of the load data. It is worth noting that these improvements are not the result of a single parameter adjustment, but rather the combined effect of adaptive refinements in model capacity, regularization, and training dynamics. This coordinated optimization process contributes to a more stable and generalizable forecasting performance. Overall, the results indicate that the proposed framework achieves meaningful performance gains while maintaining robustness and consistency across different evaluation perspectives.

Table 3. Comparison of prediction errors before and after optimization

Metric	Definition	Initial Prediction	Optimized Prediction	Improvement
MSE	Mean Squared Error	0.0383	0.0217	↓ 43.3%
MAE	Mean Absolute Error	0.1372	0.0895	↓ 34.8%
RMSE	Root Mean Squared Error	0.1956	0.1473	↓ 24.7%
MAPE	Mean Absolute Percentage Error	11.32%	7.85%	↓ 30.7%
R^2	Coefficient of Determination	0.3683	0.7126	↑ 93.5%

Table 4 presents a comparative evaluation of the proposed method against several benchmark models. Traditional statistical approaches such as ARIMA exhibit limited capability in capturing nonlinear dependencies, resulting in higher prediction errors. The Multi-Layer Perceptron (MLP) model improves performance by introducing nonlinear mapping, yet lacks the ability to effectively model temporal dependencies. The baseline LSTM model demonstrates improved performance due to its capacity to capture sequential patterns; however, its effectiveness is constrained by fixed hyperparameter configurations. In contrast, the proposed method consistently outperforms all benchmark models across all evaluation metrics, indicating that the adaptive optimization strategy enhances both predictive accuracy and model robustness. These results suggest that the integration of a feedback-driven optimization mechanism provides a meaningful advantage over conventional forecasting approaches.

Table 4. Comparison of forecasting performance across different models

Model	MSE	MAE	RMSE	R^2
ARIMA	0.0521	0.1683	0.2283	0.4215
MLP	0.0447	0.1492	0.2114	0.5128
LSTM (Baseline)	0.0383	0.1372	0.1956	0.3683
Proposed Method	0.0217	0.0895	0.1473	0.7126

This case study presents a complete modeling workflow that integrates simulated load data generation, baseline LSTM forecasting, GPT-guided model optimization, and comprehensive performance evaluation. Starting from synthetically constructed time series that reflect realistic environmental variations, the LSTM model was initially trained without manual tuning to establish a performance baseline. A series of targeted enhancements—suggested through GPT-based reasoning—were then applied to iteratively refine the model structure and training process. The overall improvement in forecasting accuracy is attributed to this optimization strategy, which enabled the LSTM model to better capture complex temporal dependencies while reducing the risk of overfitting. Comparative analyses of the forecast results before and after optimization, supported by both visual and quantitative evaluations, demonstrate the adaptability and effectiveness of the proposed hybrid approach in addressing electricity load forecasting under diverse and complex input conditions.

6. Conclusion

This paper proposes a hybrid forecasting framework that integrates LSTM-based time-series modeling with a GPT-assisted adaptive optimization strategy for short-term electricity load prediction. The framework is designed to enhance predictive performance through an iterative refinement process, in which model configurations are continuously adjusted based on observable training behavior and prediction errors. By combining the temporal learning capability of LSTM with a feedback-driven optimization mechanism, the proposed approach is able to better capture nonlinear dependencies and evolving patterns in electricity demand. Compared with conventional fixed-configuration models, this design provides a more flexible and adaptive forecasting paradigm that can respond to varying data characteristics without relying on rigid or manually predefined tuning procedures.

The effectiveness of the proposed method is validated through a comprehensive case study conducted under representative operating conditions. The results demonstrate that the optimized model achieves consistent improvements across multiple evaluation metrics, including reductions in MSE, MAE, RMSE, and MAPE, as well as a noticeable increase in the coefficient of determination. These improvements indicate that the model is able to produce more accurate and stable predictions, particularly in the presence of fluctuating environmental factors such as temperature and humidity. Furthermore, comparative analysis with baseline and benchmark models shows that the proposed framework offers enhanced capability in capturing temporal dynamics and reducing prediction variability, thereby improving overall forecasting reliability in a structurally meaningful manner.

From a methodological perspective, this study highlights the potential of leveraging large language models as auxiliary optimization components within data-driven forecasting frameworks. Rather than serving as direct predictors, GPT is utilized to guide model refinement through structured feedback, enabling a more interpretable and adaptive optimization process. This paradigm opens new possibilities for integrating generative models into engineering applications beyond natural language processing. At the same time, considering the deployment of such frameworks in critical energy infrastructure, system-level reliability and security remain important factors. Future implementations should incorporate appropriate protective mechanisms, such as controlled access to model interfaces, isolation of critical components, and secure data processing pipelines, to mitigate potential risks associated with external interference. Future work will focus on validating the proposed framework using real-world power system datasets and extending the model to incorporate additional data modalities, such as textual weather descriptions and operational system information. In addition, further investigation into more robust integration strategies and secure deployment architectures is expected to enhance both the reliability and practical applicability of the proposed approach in real-world energy systems.

Conflicts of interest

The authors declare no conflicts of interest.

References

[1] D. Wu and W. Lin, "Efficient residential electric load forecasting via transfer learning and graph neural networks," *IEEE Transactions on Smart Grid*, vol. 14, no. 3, pp. 2423-2431, 2022. https://doi.org/10.1109/TSG.2022.3208211.

[2] L. Cheng, H. Zang, Z. Wei and G. Sun, "Secure multi-party household load scheduling framework for real-time demand-side management," *IEEE Transactions on Sustainable Energy*, vol. 14, no. 1, pp. 602-612, 2022. https://doi.org/10.1109/TSTE.2022.3221081.

[3] A. Saxena, R. Shankar, E. F. El-Saadany, M. Kumar, O. A. Zaabi, K. A. Hosani, et al., "Intelligent load forecasting and renewable energy integration for enhanced grid reliability," *IEEE Transactions on Industry Applications*, vol. 60, no. 6, pp. 8403-8417, 2024. https://doi.org/10.1109/TIA.2024.3436471.

[4] P. Li, Z. Hu, Y. Shen, X. Cheng, and M. Alhazmi, "Short-term electricity load forecasting based on large language models and weighted external factor optimization," *Sustainable Energy Technologies and Assessments*, vol. 82, p. 104449, 2025. https://doi.org/10.1016/j.seta.2025.104449.

[5] D. Huo, C. Gu, D. Greenwood, Z. Wang, P. Zhao, and J. Li, "Chance-constrained optimization for integrated local energy systems operation considering correlated wind generation," *International Journal of Electrical Power & Energy Systems*, vol. 132, p. 107153, 2021. https://doi.org/10.1016/j.ijepes.2021.107153.

[6] K. Bitirgen and Ü. B. Filik, "A hybrid deep learning model for discrimination of physical disturbance and cyber-attack detection in smart grid," *International Journal of Critical Infrastructure Protection*, vol. 40, p. 100582, 2023. https://doi.org/10.1016/j.ijcip.2022.100582.

[7] X. Fu, M. Wu, S. Ponnarasu, and L. Zhang, "A hybrid deep learning approach for real-time estimation of passenger traffic flow in urban railway systems," *Buildings*, vol. 13, no. 6, 2023. https://doi.org/10.3390/buildings13061514.

[8] A. Mazzoccoli and M. Naldi, "Robustness of optimal investment decisions in mixed insurance/investment cyber risk management," *Risk Analysis*, vol. 40, no. 3, pp. 550-564, 2020. https://doi.org/10.1111/risa.13416.

[9] P. Li, X. Yan, T. Qian, Y. Shen, and M. Alhazmi, "Biodegradable transient data centers: a multi-objective optimization framework for sustainable computing in extreme environments," *IEEE Transactions on Industry Applications*, pp. 1-9, 2026. https://doi.org/10.1109/TIA.2026.3679649.

[10] M.-R. Kazemzadeh, A. Amjadian, and T. Amraee, "A hybrid data mining driven algorithm for long term electric peak load and energy demand forecasting," *Energy*, vol. 204, p. 117948, 2020. https://doi.org/10.1016/j.energy.2020.117948.

[11] K. Ullah, M. Ahsan, S. M. Hasanat, M. Haris, H. Yousaf, S. F. Raza, et al., "Short-term load forecasting: a comprehensive review and simulation study with CNN-LSTM hybrids approach," *IEEE Access*, vol. 12, pp. 111858-111881, 2024. https://doi.org/10.1109/ACCESS.2024.3440631.

[12] A. P. Zhao, S. Li, D. Xie, Y. Wang, Z. Li, P. J.-H. Hu, et al., "Hydrogen as the nexus of future sustainable transport and energy systems," *Nature Reviews Electrical Engineering*, pp. 447-466, 2025. https://doi.org/10.1038/s44287-025-00178-2.

[13] P. Zhao, S. Li, Y. Shen, C. Gu, L. Shan, T. Luo, et al., "Water-Energy Nexus in Power Systems: A Review," *Qeios*. 2023. https://doi.org/10.32388/SFEDFM.

[14] P. Prabu, P. Ganeshkumar, P. Krishnan, S. Dhanasekaran,W. Jiang, J. Logeshwaran, "An improved LSTM based thermal prediction and control algorithm for battery management system in hybrid electric vehicles," *International Communications in Heat and Mass Transfer*, vol. 169, 2025. https://doi.org/10.1016/j.icheatmasstransfer.2025.109930.

[15] M. He, W. Jiang, and W. Gu, "Trichrononet: advancing electricity price prediction with multi-module fusion," *Applied Energy*, vol. 371, p. 123626, 2024. https://doi.org/10.1016/j.apenergy.2024.123626.

[16] H. A. Pai, K. K. Mishra, M. T. R., J. V. M. L. Jeyan, A. Sayal, "Enhanced household energy consumption forecasting using multivariate Long Short-Term Memory (LSTM) networks with weather data integration," *Results in Engineering*, vol. 27, p. 106512, 2025. https://doi.org/10.1016/j.rineng.2025.106512.

[17] P. Li, Y. Shen, Y. Shang, and M. Alhazmi, "Innovative distribution network design using GAN-based distributionally robust optimization for DG planning," *IET Generation, Transmission & Distribution*, vol. 19, no. 1, p. e13350, 2025. https://doi.org/10.1049/gtd2.13350.

[18] G. Manzolini, A. Fusco, D. Gioffrè, S. Matrone, R. Ramaschi, M. Saleptsis, et al., "Impact of PV and EV forecasting in the operation of a microgrid," *Forecasting*, vol. 6, no. 3, pp. 591-615, 2024. https://doi.org/10.3390/forecast6030032.

[19] S. Deng, O. Sprangers, M. Li, S. Schelter, M. de Rijke, "Domain generalization in time series forecasting," *ACM Transactions on Knowledge Discovery from Data*, vol. 18, no. 5, pp. 1-24, 2024. https://doi.org/10.1145/3643035.

[20] X. Liu, Y. Zheng, Z. Du, M. Ding, Y. Qian, Z. Yang, et al., "GPT understands, too," *AI Open*, vol. 5, pp. 208-215, 2024. https://doi.org/10.1016/j.aiopen.2023.08.012.

[21] S. Long, J. Tan, B. Mao, F. Tang, Y. Li, M. Zhao, et al., "A survey on intelligent network operations and performance optimization based on large language models," *IEEE Communications Surveys & Tutorials*, vol. 27, no. 6, pp. 3915-3949, 2025. https://doi.org/10.1109/COMST.2025.3526606.

[22] P. Li, C. Gu, X. Cheng, J. Li, and M. Alhazmi, "Integrated energy-water systems for community-level flexibility: a hybrid deep Q-network and multi-objective optimization framework," *Energy Reports*, vol. 13, pp. 4813-4826, 2025. https://doi.org/10.1016/j.egyr.2025.03.059.

[23] J. Mao, Y. Qian, J. Ye, H. Zhao, and Y. Wang, "GPT-driver: learning to drive with GPT," *arXiv*, 2023. https://doi.org/10.48550/arXiv.2310.01415.

[24] S. Wu, H. Fei, L. Qu, W. Ji, and T.-S. Chua, "Next-GPT: any-to-any multimodal LLM," In Proc. 41st International Conference on Machine Learning, Vienna, Austria, Jul. 21-27, 2024, pp. 53366-53397.