# Substation-Level Carbon Emission Forecasting for EV-Rich Distribution Systems via Multimodal LLM-Embedded Neural Networks

Zixuan Zhu, Ruoheng Wang, Hongyi Xu, Chenxuan Zhang, Yu Sheng, Siqi Bu*
*Department of Electrical and Electronic Engineering*
*The Hong Kong Polytechnic University*
HKSAR, China

***Abstract*—Conventional substation-level carbon emission forecasting, based on power flow, is increasingly challenged by the uncertainty of electric vehicle (EV) charging behaviors. To overcome this challenge, this paper proposes a novel multimodal large language model (LLM)-based forecasting framework incorporating LLM and feature extraction components to enhance the capability of the forecasting model to handle stochastic carbon emission patterns. The proposed framework first generates a descriptive semantic prompt based on contextual information (e.g., weather, workday), allowing LLM to extract a rich semantic embedding. This embedding is then fused with high-dimensional features concurrently extracted from numerical substation-level data (e.g., load, renewable generation, EV charging data), thereby enabling multi-timescale prediction with respective forecasting components. Validated on a modified EV-rich IEEE 33-bus system, our framework demonstrates exceptional forecasting performance, highlighting the significant potential of fusing LLM for complex EV-integrated power system applications.**

***Index Terms*—Electric vehicles, large language model, time-series forecasting, carbon emission, deep learning**

## I. Introduction

The proliferation of EVs and renewable energy resources is critical for the transition of the modern power system to low-carbon operation [1]. However, the proliferation of EVs introduces highly complex charging-induced load patterns, complicating the corresponding substation-level carbon emission patterns. This inherent stochasticity poses challenges for the accurate prediction of substation-level carbon emissions, even though accurate emission prediction is a prerequisite for carbon-aware grid operations, given that it provides important signals for scheduling charging to low-emission hours, planning operational constraints, and carbon-aware dispatch at the distribution level [2].

While neural network models such as LSTMs and GNNs have shown success in forecasting substation-level operational data directly, including load and carbon emissions [3], their performance is limited without incorporating external context factors [4]. The carbon emission is highly related to weather (e.g., sunny) and day types (e.g., holiday) that impact renewable power generation and power demand. In parallel, recent research investigates LLM-based and language-model-inspired approaches for time-series forecasting, including adapting pretrained LLM backbones with prompt-based mechanisms to incorporate natural-language context to enhance prediction accuracy [5]. This motivates the use of LLMs: by encoding contextual weather and day type information in natural language and extracting semantic embeddings, an LLM can provide richer representations of external information related to carbon emissions, improving generalization and robustness under changing operating conditions.

979-8-3315-8138-1

To incorporate weather and day type information for better forecasting accuracy, we propose a novel framework that leverages the reasoning capabilities of large language models (LLMs) to enhance forecasting accuracy [6]. Our approach generates descriptive semantic prompts based on the system's multifaceted operational context (i.e., weather and day type), which an LLM can interpret to infer the impact on carbon emissions. This allows the LLM to extract semantic embeddings that are far more potent than simply encoding numerical features. The semantic embeddings are fused and processed by feature encoders and multi-timescale forecasting components, thereby enhancing the performance of carbon emission forecasting.

The contributions of this work are as follows.

1) A data-driven EV charging model is proposed to simulate realistic EV-rich distribution system operation scenarios for forecasting model training and validation.
2) A prompting strategy is proposed to generate context-based semantic prompts to enable LLM to interpret complex operational scenarios.
3) A novel multimodal LLM-embedded neural network is proposed, which fuses semantic embeddings with numerical substation-level features, allowing for the leverage of both semantic understanding and numerical patterns to enhance forecasting accuracy.

## II. Data-Driven Model for EV Fleet Charging

To accurately represent the complex charging patterns of EV fleets, we construct a data-driven model based on realistic data to describe realistic large-scale EV charging behavior.

### A. System and State-Variable Definitions

We consider a power distribution network with a set of charging stations, indexed by $j \in \mathcal{J}$. At any discrete time $t$, each station $j$ hosts a set of connected vehicles, denoted by $\mathcal{V}_j(t)$. The state of an individual vehicle $v \in \mathcal{V}_j(t)$ is defined by its battery capacity $C_{\text{batt}}$ and a dynamic state vector $\mathbf{x}_v(t)$:

$$\mathbf{x}_v(t) = [S_v(t), T_v^{\text{rem}}(t)] \tag{1}$$

where $S_v(t)$ is the state-of-charge (SoC) of the EV, which is in $[0, C_{\text{batt}}]$, and $T_v^{\text{rem}}(t)$ denotes the remaining time the vehicle will stay at the charging station.

The collection of all vehicle state vectors, $\{\mathcal{V}_j(t)\}_{j\in\mathcal{J}}$, defines the complete EV fleet state at time $t$.

## B. Vehicle Arrival and Departure

Vehicle arrival and departure events are sampled from probability distributions derived from real-world datasets, ensuring realistic fleet dynamics. Upon an arrival event at time $t_0$, a new vehicle $v$ is instantiated and added to the station's active set, denoted as $\mathcal{V}_j(t)$. The vehicle remains in the active set $\mathcal{V}_j(t)$ until its parking duration expires. Once $T_v^{\text{rem}}(t) \leq 0$, the vehicle is considered to depart. This approach directly utilizes real-world data to simulate realistic EV operation patterns.

## C. EV Charging

For a vehicle $v$ connected at station $j$, its state evolves over each discrete time step $\delta t$. The model assumes each vehicle attempts to charge to its full capacity, $C_{\text{batt}}$, as quickly as possible. The state transition is defined by:

$$S_v(t + \delta t) = S_v(t) + p_v(t) \cdot \eta_c \cdot \delta t \tag{2}$$

$$T_v^{\text{rem}}(t + \delta t) = T_v^{\text{rem}}(t) - \delta t \tag{3}$$

where $\eta_c$ is the charging efficiency. A vehicle is eligible to charge if it has remaining parking time and its battery is not full. This eligibility is captured by an indicator function, $\mathbb{I}_v(t) = \mathbb{I}(T_v^{\text{rem}}(t) > 0 \wedge S_v(t) < C_{\text{batt}})$.

The charging power, $p_v(t)$, is limited by the vehicle's charging curve and the remaining battery capacity. Let $\rho(s)$ be the function defining the charging rate (in kW) at a given SoC, denoted as $S_v(t)$. The charging power of the vehicle is:

$$p_v(t) = \mathbb{I}_v(t) \cdot \min\left(\rho\left(S_v(t)\right), \frac{C_{\text{batt}} - S_v(t)}{\delta t}\right) \tag{4}$$

This formulation ensures that the SoC will not exceed the battery capacity $C_{\text{batt}}$.

## D. Aggregated Power Demand at the Substation Level

The primary output of the model is the total, or aggregated, power demand at each substation $j$. This value, $P_j^{\text{agg}}(t)$, is the linear superposition of the power drawn by all individual vehicles connected to that station at time $t$.

$$P_j^{\text{chg}}(t) = \sum_{v \in \mathcal{V}_j(t)} p_v(t) \tag{5}$$

This aggregated data is used to perform data flow analysis and obtain nodal carbon information in the next section.

# III. Context-Encoded Multi-Timescale Forecasting Framework for Carbon Emission

We formulate the prediction of Nodal Carbon Intensity (NCI) as a supervised learning task. The objective is to train a model that learns a mapping function $f : \mathcal{X} \to \mathcal{Y}$ from a feature space $\mathcal{X}$ representing the system state at time $t$ to a target space $\mathcal{Y}$ representing the grid's carbon characteristics at future time $t+\Delta t$.

## A. Target Space ($\mathcal{Y}$)

The target vector $\mathbf{y}(t) \in \mathcal{Y} \subseteq \mathbb{R}^{2N}$ provides a comprehensive carbon profile of the grid by concatenating three key metrics for all $N$ buses: nodal carbon intensity (NCI), nodal carbon emission (NCE), and direct generation emissions.

$$\mathbf{y}(t) = [\boldsymbol{\lambda}_{\text{nci}}(t)^\top, \mathbf{k}_{\text{nce}}(t)^\top]^\top \tag{6}$$

The direct generation emissions, $\mathbf{k}_{\text{gen}}$, are the product of generator power outputs $\mathbf{p}_{\text{gen}}$ and their static emission factors $\boldsymbol{\lambda}_{\text{gen}}$:

$$\mathbf{k}_{\text{gen}} = \mathbf{p}_{\text{gen}} \circ \boldsymbol{\lambda}_{\text{gen}} \tag{7}$$

These source emissions are propagated through the network to determine the NCI, $\boldsymbol{\lambda}_{\text{nci}}$, via the carbon flow model [7]:

$$\boldsymbol{\lambda}_{\text{nci}} = (\mathbf{P}_{\text{diag}} - \mathbf{P}_{\text{flow}})^{-1}\mathbf{k}_{\text{gen}} \tag{8}$$

where $\mathbf{P}_{\text{diag}}$ and $\mathbf{P}_{\text{flow}}$ are matrices representing total bus power injections and branch power flows, respectively. Finally, the NCE, $\mathbf{k}_{\text{nce}}$, quantifies the emissions attributable to consumption at each bus by scaling the NCI by the nodal load vector $\mathbf{L}$:

$$\mathbf{k}_{\text{nce}} = \boldsymbol{\lambda}_{\text{nci}} \circ \mathbf{L} \tag{9}$$

## B. Feature Space ($\mathcal{X}$)

The feature vector $\mathbf{x}(t) \in \mathcal{X}$ is engineered to provide a comprehensive representation of the system by combining bus-specific data with system-wide contextual information.

For each bus $i$, we define a nodal feature vector $\mathbf{x}_i(t) \in \mathbb{R}^7$ comprising its local grid and EV fleet status:

$$\mathbf{x}_i(t) = [L_i, P_i^{\text{pv}}, P_i^{\text{wind}}, \lambda_{\text{nci},i}, N_i^{\text{ev}}, P_i^{\text{chg}}, S_i^{\text{sum}}]^\top \tag{10}$$

These components represent, respectively: local load, PV generation, wind generation, current NCI, number of connected EVs, power of charging EVs, and the sum of their SoCs.

A single contextual feature vector $\mathbf{c}(t) \in \mathbb{R}^8$ captures the global state of the system:

$$\mathbf{c}(t) = [w_{\text{today}}, w_{\text{next}}, d_{\text{today}}, d_{\text{next}}, K_{\text{sys}}, P_{\text{sys}}^{\text{ren}}, L_{\text{sys}}, t_{\text{time}}]^\top \tag{11}$$

These components are: current day's weather, next day's weather, day type, total system emissions, total renewable generation, total system load, and the time of day.

## C. Input Sequence Construction

The model's architecture is designed to process three distinct, synchronized input streams. To formalize the temporal dependency, we define an input sequence of time indices $\tau$ corresponding to a look-back window of length $T_{\text{in}}$ ending at time $t$:

$$\tau = (t - T_{\text{in}} + 1, \ldots, t) \tag{12}$$

The three input streams are the historical nodal features, historical contextual features, and a textual prompt, each aligned with this time frame.

First, we define the nodal feature stream. At any instant $k$, the feature vector for a single bus $i$ is denoted $\mathbf{x}_i(k) \in \mathbb{R}^7$. These are aggregated across all $N$ buses to form a comprehensive description of distribution system operation data.

$$\mathbf{X}_{\text{n}}(k) = [\mathbf{x}_1(k)^\top, \mathbf{x}_2(k)^\top, \ldots, \mathbf{x}_N(k)^\top]^\top \in \mathbb{R}^{7N} \tag{13}$$

The input tensor for the nodal feature encoder, $\mathbf{X}_{\text{n}}(\tau) = [\mathbf{X}_{\text{n}}(k)]_{k\in\tau} \in \mathbb{R}^{7N\times T_{\text{in}}}$, is the sequence of these snapshots over the time indices in $\tau$:

$$\mathbf{X}_{\text{n}}(\tau) = [\mathbf{X}_{\text{n}}(t - T_{\text{in}} + 1), \ldots, \mathbf{X}_{\text{n}}(t)] \tag{14}$$

Second, the system-wide contextual feature vector, $\mathbf{c}(k) \in \mathbb{R}^8$, is treated as a separate input stream. The sequence for the contextual feature encoder is constructed as:

$$\mathbf{X}_{\text{c}}(\tau) = [\mathbf{c}(k)]_{k\in\tau} = [\mathbf{c}(t - T_{\text{in}} + 1), \ldots, \mathbf{c}(t)] \in \mathbb{R}^{8\times T_{\text{in}}} \tag{15}$$

979-8-3315-8138-1

The third input is a textual prompt, $T(\tau) \in \mathbb{R}^{T_{\text{in}}}$, which provides qualitative or event-based information of the input sequence, such as alerts about extreme weather or grid maintenance. The processing of this prompt will be detailed in the next section.

### D. *Multi-Timescale Carbon Emission Forecasting*

The supervised learning task is to map these three parallel input streams to a future sequence of target vectors, $\mathbf{Y}_{\text{seq}}(\tau_{\text{out}})$. This is performed across multiple time scales by defining a forecast horizon sequence $\tau_{\text{out}} = (t+1, \ldots, t+T_{\text{out}})$ for hour-ahead ($T_{\text{out}} = 1$), day-ahead ($T_{\text{out}} = 24$), and 72-hour-ahead ($T_{\text{out}} = 72$) predictions. The goal is to learn a mapping function $f$ such that:

$$\mathbf{Y}_{\text{seq}}(\tau_{\text{out}}) \approx f(\mathbf{X}_{\text{n}}(\tau), \mathbf{X}_{\text{c}}(\tau), T(\tau)) \tag{16}$$

where $\mathbf{Y}_{\text{seq}}(\tau_{\text{out}}) = [\mathbf{y}(t+1), \ldots, \mathbf{y}(t+T_{\text{out}})]$. This multi-timescale formulation enables the model to separately process and then fuse LLM-embedded information, allowing it to learn complex temporal patterns for robust forecasting.

## IV. Context-Based Forecasting with Multimodal LLM-Embedded Neural Network

### A. *Context-Based Prompt Generation*

We use a Large Language Model (LLM) to generate a carbon emission forecast that acts as a derived input feature for our main prediction model, $f$. This is achieved using a two-part prompt constructed from the contextual feature vector $\mathbf{c}(t)$.

*a) System Prompt:* A static system prompt defines the LLM's role as a power system forecasting agent and enforces a structured output.

```
ROLE: power system forecasting agent,
INSTRUCT: Your task is to DIRECTLY forecast power
    ↪ system data.
NEVER explain methods, NEVER provide code, NEVER
    ↪ suggest approaches.
You should reasonably forecast future carbon emission
    ↪ pattern based on load and clean energy
    ↪ generation.
OUTPUT FORMAT:
carbon_emission_forecast: [value1, value2, ...]
```

*b) Dynamic Context Prompt:* A dynamic context prompt is generated from the feature vector $\mathbf{c}(t)$. It translates features like weather and day type into descriptive phrases (e.g., sunny, high PV generation) and embeds recent time-series data for system load, renewable generation, and emissions into a human-readable narrative. An example is shown below.

```
Now is workday, high load 10 o'clock.
Current weather is overcast, low PV generation.
Tomorrow is workday, high load.
Tomorrow weather is sunny, high PV generation.
recent data is (per hour):
system clean power generation: ['0.10', '0.20', '0.60',
    ↪ '0.60', '0.80']
system load: ['4.60', '4.90', '5.00', '5.10', '5.20']
system carbon emission: ['3.38', '3.53', '3.30',
    ↪ '3.38', '3.30']
```

The combined prompt is passed to the LLM. Its output, a forecast vector, is appended to the original input vector $\mathbf{x}(t)$, enriching it with a forward-looking, context-aware feature for the final multi-timescale prediction.

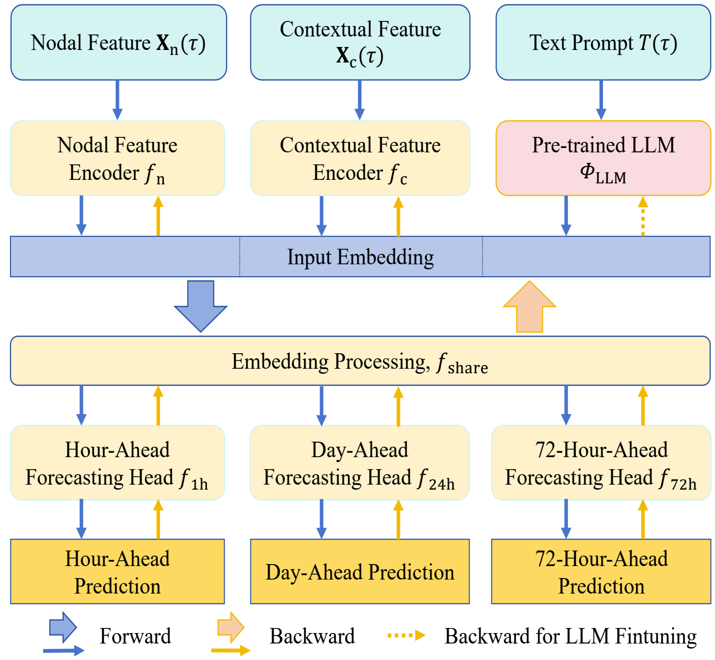


Fig. 1. Multimodal LLM-embedded neural network for multi-time-scale substation-level carbon emission forecasting

### B. *Architecture of LLM-embedded Multimodal Neural Network*

The proposed multimodal LLM-embedded neural network aims to process both numerical data and context prompts, with an architecture illustrated in Fig. 1. First, a frozen Large Language Model, $\Phi_{\text{LLM}}$, encodes the text prompt $T(\tau)$ from the end of the input window into a semantic embedding:

$$\mathbf{e}(t) = \Phi_{\text{LLM}}(T(\tau)) \in \mathbb{R}^{d_{\text{llm}}} \tag{17}$$

Simultaneously, the historical sequences, $\mathbf{X}_{\text{n}}(\tau)$ and $\mathbf{X}_{\text{c}}(\tau)$, are flattened into single vectors to be processed by multi-layer perceptron (MLP) encoders.

Next, two dedicated MLP encoders ($f_{\text{n}}, f_{\text{c}}$) project the flattened sequences and the text embedding into a common latent space. Their outputs are concatenated by another encoder, $f_{\text{share}}$, to produce a single unified representation, $\mathbf{h}_{\text{share}}(t)$:

$$\mathbf{h}_{\text{share}}(t) = f_{\text{share}}\left(\left[f_{\text{n}}(\mathbf{X}_{\text{n}}(\tau))^{\top}, f_{\text{c}}(\mathbf{X}_{\text{c}}(\tau))^{\top}, \mathbf{e}(t)^{\top}\right]^{\top}\right) \tag{18}$$

Finally, this unified representation is fed into three independent, time-specific MLP output heads ($f_{1\text{h}}, f_{24\text{h}}, f_{72\text{h}}$), each dedicated to a distinct forecast horizon: 1-hour ahead, 24-hours ahead, and 72-hours ahead. Each head predicts both the NCI and NCE values for its specific future time step.

The complete model, excluding the frozen $\Phi_{\text{LLM}}$, is trained end-to-end. We also found that fine-tuning the LLM during training can improve the accuracy of forecasting, although using a frozen LLM can outperform traditional approaches.

## V. Case Study

### A. *Experiment Setting*

The proposed method is validated using real-world power system operation data on a modified EV-rich IEEE 33-bus system, as depicted in Fig. 3. The load data is derived from [8], the renewable power generation data is derived from [9], and the EV operation data, which includes charging patterns and schedules, is derived from [10]. The total length of the dataset is 50

979-8-3315-8138-1

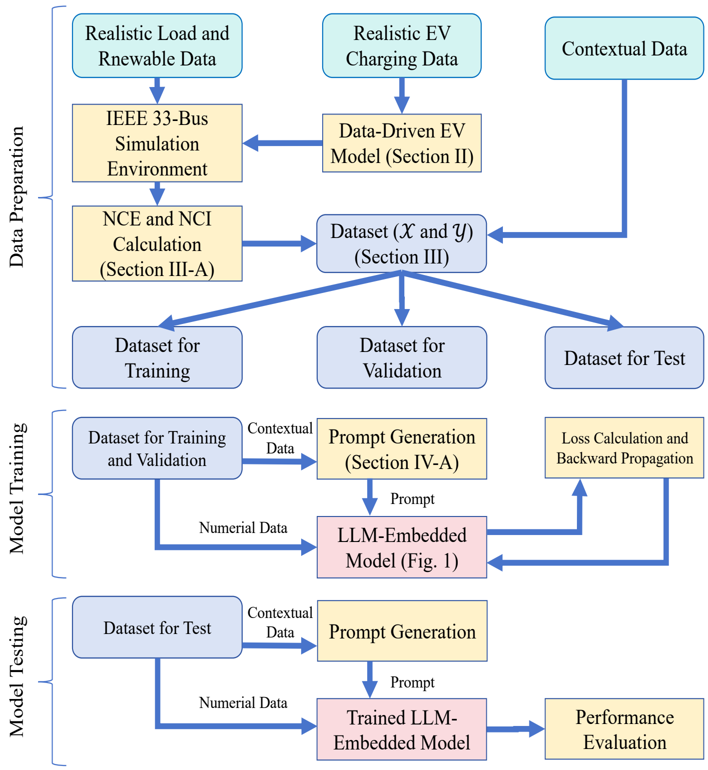


Fig. 2. Procedure for data preparation, model training, and testing

TABLE I
HYPERPARAMETER OF TRAINING

| Hyperparameter | Value | Hyperparameter | Value |
|---|---|---|---|
| Epoch | 100 | Finetuning Learning Rate | 0.00005 |
| Learning Rate | 0.0005 | Finetuning Epoch 1 | 1-5 |
| Batch Size | 5 | Finetuning Epoch 2 | 90-100 |

days. The training dataset spans 40 days, and the remaining 10 days are reserved for validation and for testing the forecasting performance. The procedure is illustrated in Fig. 2.

The proposed LLM-based approach is compared with three different baselines, as specified below:

- *LLM_Context_FT (Proposed)*: Context-enhanced model with an LLM context encoder and LLM fine-tuning.
- *LLM_Context*: Same as above but with a frozen (non-fine-tuned) LLM, to isolate the benefit of fine-tuning.
- *Context*: Context-enhanced model using numerical context features only without LLM.
- *LSTM*: LSTM baseline using only historical carbon-emission data (no external context); hidden size 3000 with separate multi-horizon output heads.
- *MLP*: MLP baseline using only historical carbon-emission data (no external context); hidden size 3000 with separate multi-horizon output heads.

This paper employs the Qwen3-0.6b as the LLM module and finetuned with low-rank adaptation (LoRA), chosen for its strong performance and manageable size for fine-tuning. The key hyperparameters for the training process for all baselines are detailed in Table I. All experiments are conducted on a workstation equipped with an Intel(R) Core(TM) i7-14700KF CPU and a NVIDIA GeForce RTX 4070 Ti SUPER GPU.

## B. Training Performance

The training and validation loss curves for all compared methods are presented in Fig. 4. These curves illustrate the

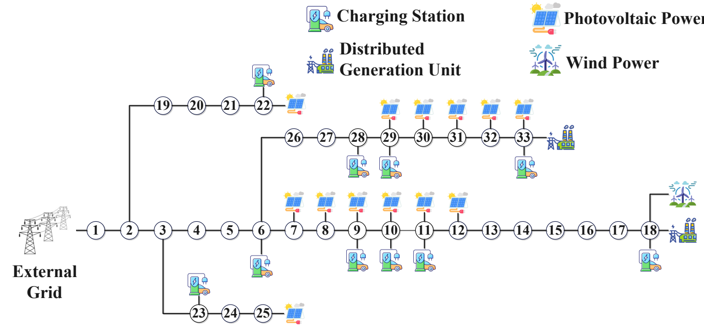


Fig. 3. Modified EV-rich IEEE 33-bus system with multiple renewable resources

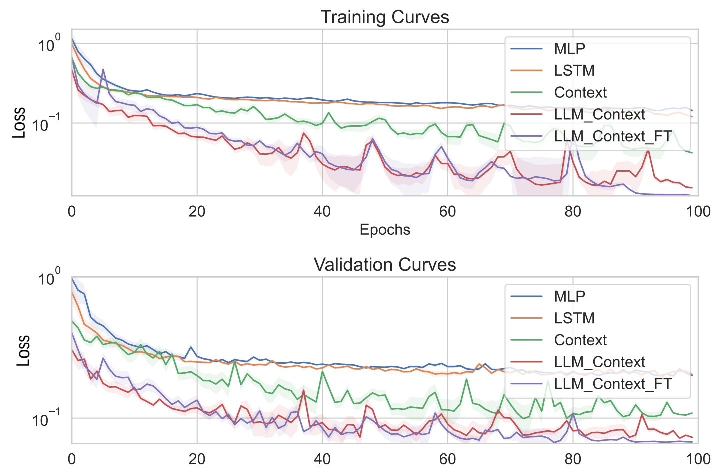


Fig. 4. Training curves and validation curves of all baselines

learning dynamics and generalization capabilities of each model throughout the training epochs.

It is observed that all models converge, indicating that the training process is stable for all baselines. The *MLP* and *LSTM* baselines exhibit similar training and validation loss, highlighting the limitations of forecasting with historical time-series data alone. The *Context* baseline achieves a significantly lower loss, which confirms the hypothesis that incorporating operational context is crucial for improving forecasting accuracy.

The introduction of the LLM yields a further substantial reduction in loss. The *LLM_Context* baseline, even without fine-tuning, outperforms the *Context* baseline, demonstrating the LLM's superior ability to extract semantic features from the prompt and improve the forecasting accuracy.

Finally, the proposed *LLM_Context_FT* model achieves the lowest training and validation loss of all methods. The performance gap between the *LLM_Context* and *LLM_Context_FT* models validates the benefit of fine-tuning, which adapts the LLM's parameters to the specific forecasting task. The lower training and validation curves indicate the effectiveness of the semantic embeddings extracted by the LLM component.

## C. Test Result

To quantitatively evaluate the forecasting capability of the models, a comprehensive set of performance metrics was calculated on the test dataset. The results, detailed in Table II, provide a clear comparison of the predictive accuracy of different baselines. The metrics include mean squared error (MSE), mean absolute error (MAE), R-squared ($R^2$), mean squared log error (MSLE), and median absolute error (MedAE).

TABLE II
FORECASTING EVALUATION ON TEST DATA

| Method | MSE | MAE | $R^2$ | MSLE | MedAE |
|---|---|---|---|---|---|
| MLP | 0.0117 | 0.0561 | 0.4155 | 0.0065 | 0.0253 |
| LSTM | 0.0127 | 0.0573 | 0.4183 | 0.0071 | 0.0226 |
| Context | 0.0062 | 0.0361 | 0.7153 | 0.0038 | 0.0163 |
| LLM_Context | 0.0050 | 0.0278 | 0.8141 | 0.0027 | 0.0115 |
| LLM_Context_FT | 0.0043 | 0.0257 | 0.8262 | 0.0024 | 0.0091 |

TABLE III
COMPUTATIONAL COST COMPARISON

| Model Type | Training Time (Seconds) | Memory (GB) |
|---|---|---|
| MLP, LSTM, Context | 700 | 2.3 |
| LLM_Context | 2,451 | 5.7 |
| LLM_Context_FT | 3,386 | 15.7 |

The results in Table II show a clear and consistent trend of improvement that aligns with the training performance. The *MLP* model serves as a lower bound on performance. The *Context* model significantly improves upon this baseline across all metrics, for instance, reducing MSE from 0.0117 to 0.0062 and increasing the $R^2$ score from 0.4155 to 0.7153. This underscores the value of contextual data.

The *LLM Context* model further improves performance, achieving a $R^2$ score of 0.8141, which indicates that the LLM's feature extraction is more effective than using raw numerical context. The proposed *LLM_Context_FT* achieves the highest performance, with the lowest error scores (MSE of 0.0043, MAE of 0.0257) and the highest $R^2$ score of 0.8262, which confirms that fine-tuning the LLM enables the model to make more accurate predictions.

Computational cost comparison of different methods is given in Table III. The training times for conventional neural networks, including MLP, LSTM, and context baselines, are around 700 seconds, whereas LLM-based approaches require more memory and longer training times for better forecasting accuracy. Specifically, the proposed LLM_Context_FT needs 15.7 GB of memory and 3386 seconds for LLM finetuning.

Fig. 5 provides a qualitative assessment of the proposed model's performance across different forecasting horizons: hour-ahead, day-ahead, and 72-hour-ahead. For the hour-ahead forecast, the forecasted NCI and NCE curves closely align with their respective real ground-truth curves, capturing the dynamic fluctuations in carbon emissions with high fidelity. This visual result corroborates the excellent quantitative metrics in Table II.

As the forecasting horizon extends to day-ahead and further to 72-hours-ahead, a natural and expected degradation in performance is observed. While the model still successfully captures the general trend and major peaks, the point-to-point accuracy compromises with the increase in prediction lead time, which is a common characteristic in time-series forecasting. Nevertheless, these results collectively validate the robustness and effectiveness of the proposed LLM-based context-enhanced approach for multi-time-scale carbon emission forecasting.

## VI. CONCLUSION

This paper addresses the critical challenge of carbon emission forecasting in EV-rich power systems by using a novel multimodal LLM-embedded neural network. Our framework first generates semantic prompts based on the operation context to enable the LLM to perform semantic reasoning. The extracted semantic embeddings are then concatenated with high-dimensional features of numerical system operation data with multi-timescale forecasting components for the multi-timescale prediction. The exceptional forecasting accuracy achieved on a modified EV-rich IEEE 33-bus system confirms the efficacy of our approach, demonstrating that integrating an LLM with a neural network can effectively improve forecasting accuracy. This work not only proposes an LLM-embedded forecasting method for substation-level carbon emission forecasting but also provides valuable insights for future research in utilizing LLM-based approaches for power system decarbonization with more external information.

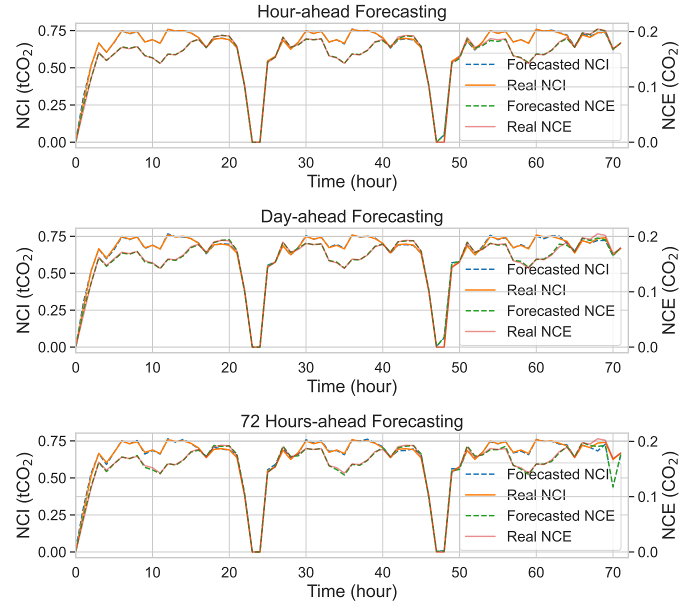


Fig. 5. Multi-time-scale forecasting result

## REFERENCES


[1] N. Shaukat *et al.*, "Decentralized, Democratized, and Decarbonized Future Electric Power Distribution Grids: A survey on the paradigm shift from the conventional power system to micro grid structures," *IEEE Access*, vol. 11, pp. 60957–60987, Jan. 2023.

[2] M. Saklani, D. K. Saini, M. Yadav, and Y. C. Gupta, "Navigating the challenges of EV integration and Demand-Side management for India's sustainable EV growth," IEEE Access, p. 1, Jan. 2024.

[3] Z. Zhu, R. Wang, S. Bu, and R. Guglielmi, "Two-Stage Real-Time Carbon Emission Monitoring for Low-Carbon Power System Operation: a Graph Neural Network-Based approach," Protection and Control of Modern Power Systems, vol. 10, no. 3, pp. 166–183, May 2025.

[4] Y. -K. Wu, Q. -T. Phan and Y. -J. Zhong, "Overview of Day-Ahead Solar Power Forecasts Based on Weather Classifications and a Case Study in Taiwan," in IEEE Transactions on Industry Applications, vol. 60, no. 1, pp. 1409-1423, Jan.-Feb. 2024.

[5] F. Jia, K. Wang, Y. Zheng, D. Cao, and Y. Liu, "GPT4MTS: Prompt-based large language model for multimodal time-series forecasting," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 21, pp. 23343–23351, Mar. 2024, doi: 10.1609/aaai.v38i21.30383.

[6] H. Mirshekali, M. R. Shadi, F. G. Ladani, and H. R. Shaker, "A review of large language models for energy Systems: applications, challenges, and future prospects," IEEE Access, p. 1, Jan. 2025.

[7] C. Kang et al., "Carbon emission flow from generation to demand: A Network-Based model," IEEE Transactions on Smart Grid, vol. 6, no. 5, pp. 2386–2394, Jan. 2015.

[8] Elia Open Data Portal, "Real-time and historical electricity grid data," [Online]. Available: https://www.elia.be/en/grid-data, Accessed: Jan. 15, 2025.

[9] Lin, Z., Zhou, Q., Wang, Z., Wang, C., Bookhart, D. B., & Leung-Shea, M. (2025). A high-resolution three-year dataset supporting rooftop photovoltaics (PV) generation analytics. *Scientific Data*, *12*(1).

[10] K. Baek, E. Lee, and J. Kim, "A dataset for multi-faceted analysis of electric vehicle charging transactions," *Scientific Data*, vol. 11, no. 1, Mar. 2024, doi: 10.1038/s41597-024-02942-9.

[11] A. Yang *et al.*, "QWEN3 Technical Report," *arXiv (Cornell University)*, May 2025, doi: 10.48550/arxiv.2505.09388.