Project2: Day-ahead (Short-term) Load Forecasting

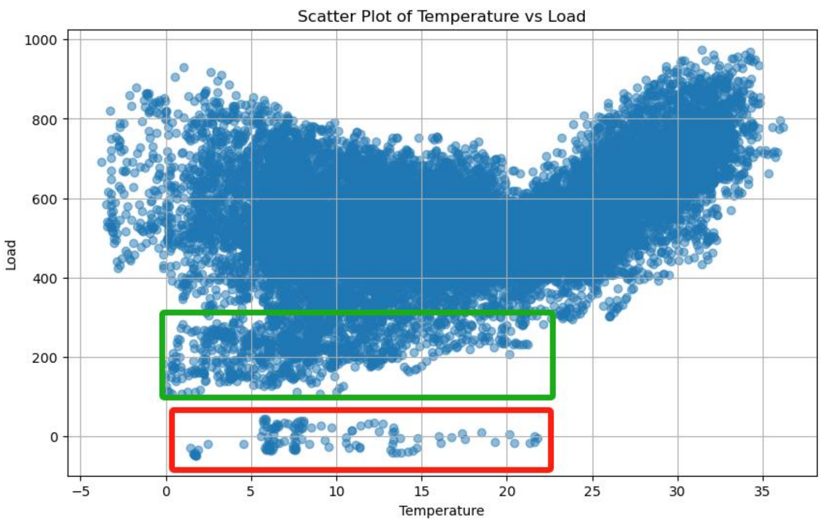
1. Introduction

In any power system, maintaining a balance between electricity supply and demand is crucial, as it significantly affects the stability and frequency of the power grid. Achieving this balance requires accurate day-ahead (short-term) load forecasting, which plays a vital role in supply planning, generation reserves, system security, dispatch scheduling, demand-side management, and other critical decision-making processes. In this project, we focus on city-level day-ahead load forecasting to support these objectives.

2. Data Cleaning

Before the short-term load forecast, data cleaning is essential to reduce the impact of bad data on the model training process. The first step is to identify the anomalous power data. As shown in the figure below, there are some data points (in the red and green boxes) that are far from other points, and some of them even have negative load values. After checking the details about these points, the points in the green box are usually influenced by holidays, while the points in the red box are mostly abnormal.

After identifying the data points that need to be cleaned, the next step is to decide how to fix the bad data in the dataset. In this project, a Multi-layer Perceptron (MLP) is applied to fix the bad data. First, the data that are relatively normal are set as the training data. The MLP model is used to identify the relationship between load and variables such as weather conditions and time features. Then, the well-trained model is used to generate relatively normal data to replace the abnormal data. It is important to note that the data is obtained from the real world and has been scaled, as discussed in my chat with the TA. This means that negative data may not necessarily be abnormal, as some unexpected events (like COVID-19-caused administrative control) may result in extremely low values. However, in my opinion, such conditions cannot be found in the current dataset, so it is difficult to determine whether these unexpected events should be considered. Therefore, it is better to treat these data points as abnormal rather than normal ones for training a more robust and accurate model.



3. Project Analysis

Load forecasting fundamentally involves time series prediction, aiming to estimate future values based on historical time series data and relevant auxiliary variables. Current approaches to load forecasting can be broadly categorized into statistical analysis-based methods and machine learning-based methods. Among the former, classical models such as ARIMA and SARIMA have demonstrated considerable effectiveness in forecasting stationary time series. However, electricity load data often exhibits non-stationary characteristics and is influenced by a multitude of factors with complex non-linear relationships. Consequently, the applicability and predictive accuracy of traditional statistical models in real-world load forecasting remain limited and require further enhancement to address these challenges.

With advancements in machine learning, classical algorithms such as Support Vector Machines (SVM), Random Forest (RF), and XGBoost have been widely applied to load forecasting tasks, achieving notable success in capturing the non-linear relationships between variables. However, these traditional machine learning models often rely heavily on complex feature engineering to perform well, which can limit their practicality and scalability in real-world applications. The emergence of deep learning methods has significantly mitigated these challenges, reducing the dependency on manual feature engineering while delivering improved predictive accuracy. Models specifically designed for time series data, such as RNNs, LSTMs, and Transformers, have proven particularly effective in load forecasting, offering state-of-the-art performance and transforming the way such tasks are approached. In this project, deep learning algorithm will also be applied.

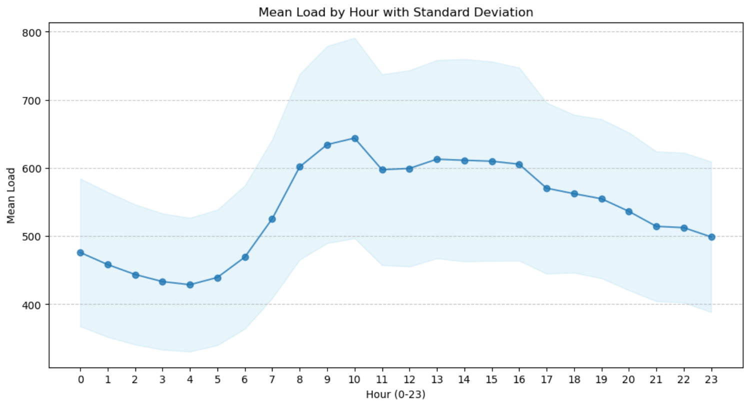


Fig.1 City-Wide Electricity Load Variation: Mean and Standard Deviation Over 24 Hours

Before constructing a forecasting model, it is essential to identify the key factors influencing electricity load. In this study, the objective is to predict the load across an entire city. In addition to historical load levels, time is a well-recognized factor influencing load variations, as electricity demand often follows distinct periodic patterns. Figure 1 illustrates the mean hourly load and its standard deviation over a 24-hour period, based on two years of historical data, highlighting the characteristic diurnal cycle of city-wide electricity consumption. Furthermore, electricity load demonstrates noticeable variations across different days of the week and throughout the seasons (months), reflecting temporal dynamics in consumption behavior. Figure 2 presents the statistical distribution of load across different weekdays, showing minor fluctuations between weekdays and weekends. In contrast, Figure 3 depicts the monthly distribution of load, emphasizing the significant seasonal variations in electricity demand. The analysis reveals that the day of the week has a negligible effect on load variation, whereas months and seasons introduce notable differences in both the mean and variance of the load, underscoring pronounced seasonal patterns. Furthermore, holidays exert a noticeable influence on electricity load, creating deviations from typical patterns. This is visually supported by Figure 4, which compares load distributions across different day types: non-holidays, holidays, and the days following holidays, represented by the numbers 0, 1, and 2, respectively. It is evident from the figure that substantial differences exist in electricity load across these day types. In conclusion, the analysis demonstrates that temporal characteristics, including time of day, season, and holiday effects, are significant determinants of electricity load variations.

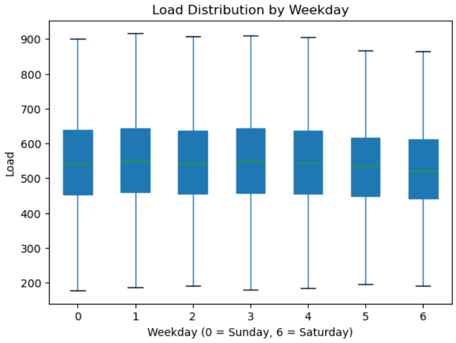


Fig. 2

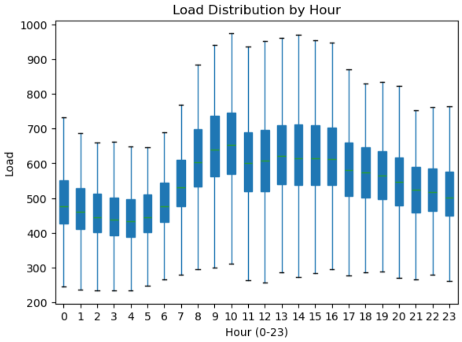


Fig. 3

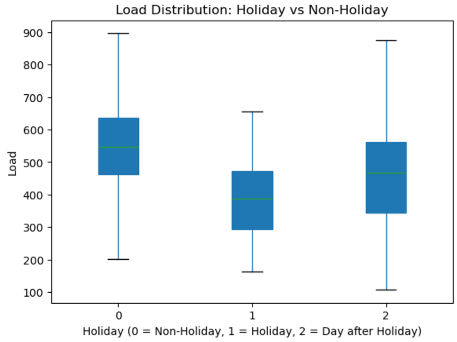


Fig. 4

In addition to temporal factors, weather conditions play a critical role in influencing electricity load. In the dataset, there are four kinds of weather variables, temperature, humidity, precipitation and wind speed . These factors influence electricity consumption directly or indirectly by altering human behavior and energy usage. For instance, high temperatures often lead to increased air conditioning usage during summer, while low temperatures in winter drive heating demand. Similarly, humidity levels can amplify the perception of temperature extremes, further impacting energy consumption. Wind speed and precipitation also contribute by influencing outdoor activities, which can indirectly affect electricity usage patterns.

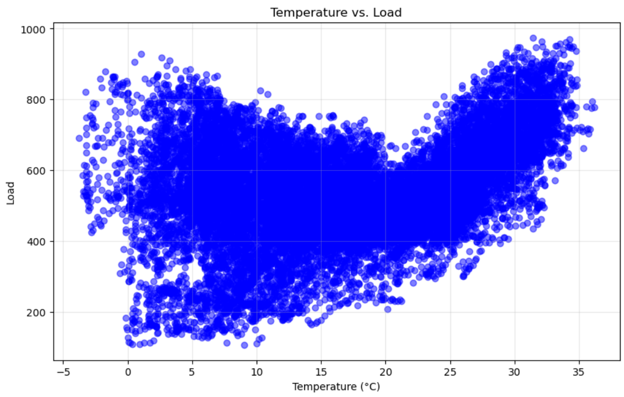


Fig. 5 Relationship Between Temperature and Electricity Load: Non-Linear U-Shaped Pattern

To illustrate the relationship between weather and load, we examine the impact of temperature, which is widely recognized as one of the most influential weather factors. Fig. 5 presents the relationship between temperature and electricity load for the studied region, based on historical data. The figure reveals a non-linear relationship, often characterized by a "U-shaped" or "V-shaped" pattern: electricity demand increases significantly at both high and low temperature extremes due to cooling and heating requirements, respectively, while moderate temperatures correspond to lower energy consumption levels. Additionally, temperature's impact on electricity load often exhibits a lagged effect, as changes in temperature can influence load over subsequent hours or even days. For example, persistent hot weather can lead to cumulative cooling demand, while prolonged cold spells drive extended heating usage. Capturing these lagged effects is critical for improving the accuracy of load forecasting models. However, due to space limitations, a detailed analysis of lagged temperature effects is not provided in this discussion.

This analysis highlights the need to account for weather variables in load forecasting models. Accurately capturing the non-linear and dynamic effects of temperature, humidity, wind speed, and precipitation can significantly enhance forecasting performance, particularly in regions where weather conditions exhibit substantial variability.

4. Proposed Framework

An analysis of the available data indicates that effective load forecasting necessitates the incorporation of historical load data from preceding time periods, temporal characteristics of the target forecasting horizon, and relevant weather variables. In practical engineering applications, these weather features are typically derived from numerical weather prediction models.

In time series forecasting tasks, models like LSTM are widely regarded as the preferred choice due to their superior ability to capture temporal dependencies. However, in load forecasting, a key challenge often arises: time series generated from different features frequently have inconsistent lengths, making it difficult to align load features with auxiliary features along time steps for effective input into the network. This misalignment introduces significant challenges in leveraging auxiliary features for the target forecasting period.

To address this issue, we adopt TimeXer, a novel model introduced at NeurIPS 2024. TimeXer employs a modular architecture that separately processes endogenous variables (e.g., historical load) and exogenous variables (e.g., weather data, temporal features), enabling specialized handling of each type. A cross-attention mechanism is utilized to iteratively integrate semantic information between these variables, enhancing the model's ability to capture complex dependencies and improving forecasting accuracy. The structure of TimeXer is shown in Fig. 6.

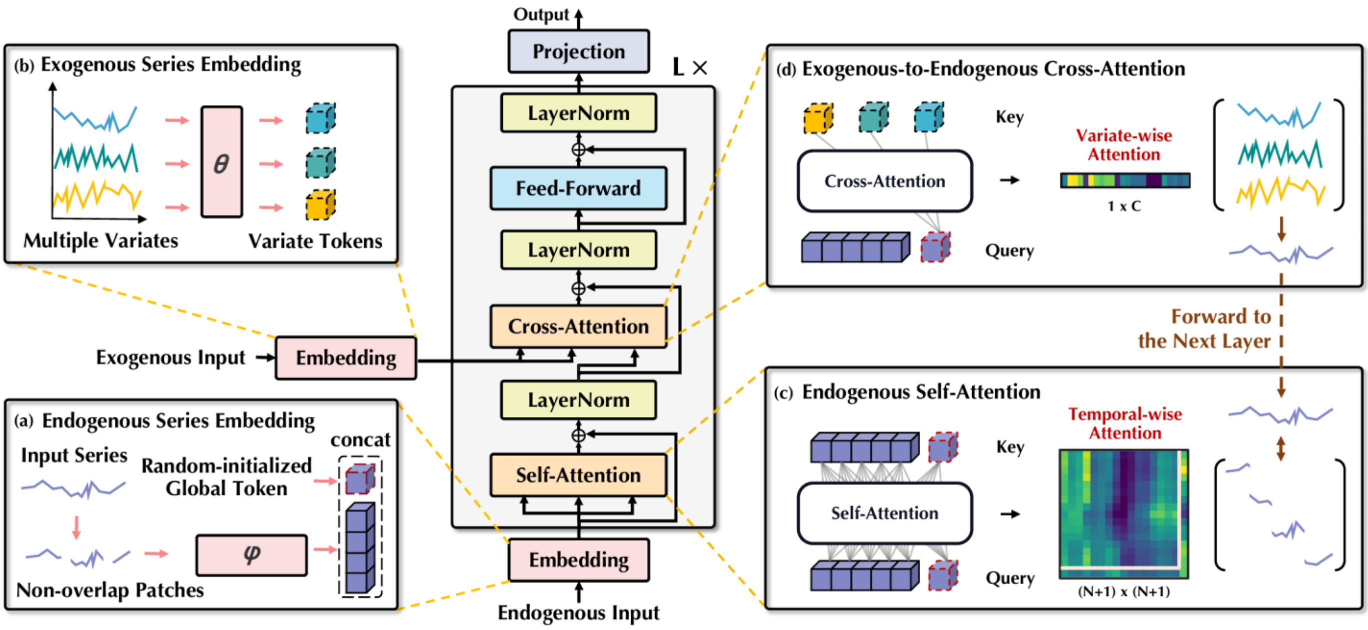


Fig. 6 The Structure of TimeXer

The following is to represent TimeXer in mathematical form. However, due to the limitation of space, something detailed will be simplified. If necessary, please refer the original paper (TimeXer: Empowering Transformers for Time Series Forecasting with Exogenous Variables).

In forecasting with exogenous variables, we are given an endogenous time series $\mathbf{x}_{1:T} = \{x_1, x_2, \ldots, x_T\} \in \mathbb{R}^{T\times 1}$ and multiple exogenous series $\mathbf{z}_{1:T_{\text{ex}}} = \{\mathbf{z}^{(1)}_{1:T_{\text{ex}}}, \mathbf{z}^{(2)}_{1:T_{\text{ex}}}, \ldots, \mathbf{z}^{(C)}_{1:T_{\text{ex}}}\} \in \mathbb{R}^{T_{\text{ex}}\times C}$. Here, x_i denotes the value at the i-th time point, $\mathbf{z}^{(i)}_{1:T_{\text{ex}}}$ represents the i-th exogenous variable, and C is the number of exogenous variables. In addition, T and T_{ex} are the look-back window lengths of the endogenous and exogenous variables respectively. The goal of forecasting model $\mathcal{F}_\theta$ parameterized by θ is to predict the future S time steps $\mathbf{x} = \{x_{T+1}, x_{T+2}, \ldots, x_{T+S}\}$ based on both historical observations $\mathbf{x}_{1:T}$ and corresponding exogenous series $\mathbf{Z}_{1:T_{\text{ex}}}$:

$$\mathbf{x}_{T+1:T+S} = \mathcal{F}_\theta\left(\mathbf{x}_{1:T}, \mathbf{z}_{1:T_{\text{ex}}}\right) \tag{1.1}$$

Endogenous variables embedding:

$$
\begin{aligned}
\{\mathbf{s}_1,\mathbf{s}_2,\ldots,\mathbf{s}_N\} &= \text{Patchify}(\mathbf{x}) \\
\mathbf{P}_{\text{en}} &= \text{PatchEmbed}(\mathbf{s}_1,\mathbf{s}_2,\ldots,\mathbf{s}_N) \\
\mathbf{G}_{\text{en}} &= \text{Learnable}(\mathbf{x})
\end{aligned} \tag{1.2}
$$

Exogenous variables embedding:

$$
\mathbf{V}_{\text{ex,i}} = \text{VariateEmbed}(\mathbf{z}^{(i)}), i \in \{1,\cdots,C\} \tag{1.3}
$$

Endogenous Self-Attention:

$$
\mathbf{P}_{\text{en}}^{l}, \mathbf{G}_{\text{en}}^{l} = \text{LayerNorm}\left(\left[\mathbf{P}_{\text{en}}^{l}, \mathbf{G}_{\text{en}}^{l}\right] + \text{Self}-\text{Attention}\left(\left[\mathbf{P}_{\text{en}}^{l}, \mathbf{G}_{\text{en}}^{l}\right]\right)\right) \tag{1.4}
$$

Exogenous-to-Endogenous Cross-Attention:

$$
\text{Variate-to-Global}: \mathbf{G}_{\text{en}}^{l} = \text{LayerNorm}\left(\mathbf{G}_{\text{en}}^{l} + \text{Cross}-\text{Attention}\left(\mathbf{G}_{\text{en}}^{l}, \mathbf{V}_{\text{ex}}\right)\right) \tag{1.5}
$$

$$
\begin{aligned}
\mathbf{P}_{\text{en}}^{l+1} &= \text{Feed-Forward}\left(\mathbf{P}_{\text{en}}^{l}\right) \\
\mathbf{G}_{\text{en}}^{l+1} &= \text{Feed-Forward}\left(\mathbf{G}_{\text{en}}^{l}\right)
\end{aligned} \tag{1.6}
$$

Forecasting Loss:

$$
Loss = \sum_{i=1}^{S} \left\| \mathbf{x}_i - \hat{\mathbf{x}}_i \right\|_2^2, \; where\, \hat{\mathbf{x}} = \text{Projection}\left(\left[\mathbf{P}_{\text{en}}^{L}, \mathbf{G}_{\text{en}}^{L}\right]\right) \tag{1.7}
$$

However, to tailor TimeXer specifically for load forecasting tasks, we introduced a key enhancement targeting the representation of temporal features. Temporal features, such as month, hour, or holiday indicators, are inherently discrete and often encoded as numerical labels (e.g., 1, 2, 3, 4). This naive encoding can introduce unintended ordinal relationships that interfere with the learning process of neural networks. The original TimeXer implementation did not address this limitation. To resolve this, we incorporated a learnable temporal embedding layer, which maps discrete temporal features into a continuous, high-dimensional space. This approach mitigates numerical bias, enhances the network's ability to model temporal features effectively, and improves overall performance in load forecasting tasks.

5. Experiment Results

I have uploaded all codes to Github, the repository link is below:

https://github.com/RuidongDavidLin/Load-Forecasting-Open-Source

Please give me a star, if you think my code is helpful for you. This is very important for me! Thank you!!