SEP 799 – Project Report

Forecasting of Electric Vehicle Charging Loads on Distribution Systems

McMaster University

ENGINEERING

W Booth School of Engineering Practice and Technology

Community Partner: Burlington Hydro Inc.

Faculty Lead: Dr. Marjan Alavi

Group Members:

Junting Ye (400552280)

Yinghua Ma (400552029)

1 Background

The project's primary goal is to forecast EV charging loads to support Burlington Hydro Inc. in managing electrical load distribution efficiently. The report delves into methodologies that utilize data analysis, machine learning models, and real-time data acquisition to predict EV charging demand accurately. By understanding EV adoption patterns, charging behaviors, and the spatial distribution of residential EV chargers, the project aims to optimize resource allocation and enhance grid stability. The initiative also focuses on environmental stewardship by reducing greenhouse gas emissions and supporting the transition to a low-carbon economy.

1.1 Project Introduction

The rapid adoption of electric vehicles (EVs) is reshaping the landscape of electricity demand, particularly at the end-user distribution level. Residential EV charging, which often occurs during peak hours, presents unique challenges for local distribution systems like Burlington Hydro Inc.'s. These challenges primarily involve managing fluctuations in demand, ensuring load balancing, and maintaining the reliability and resilience of the grid infrastructure.

Burlington Hydro Inc. (BHI) serves approximately 68,500 customers through a network of distribution lines and substations. With the increase in EV registrations, BHI faces challenges in forecasting load demands accurately, which complicates resource allocation and grid management. This project involves refining BHI's forecasting methodologies using advanced analytics and machine learning techniques. The project synthesizes data from various sources, including smart meters and demographic trends, to construct a comprehensive view of demand drivers. This approach ensures a resilient power grid capable of adapting to urban mobility and energy consumption shifts.

1.2 Literature Review

The increasing adoption of electric vehicles (EVs) has prompted significant research into load forecasting methodologies over the past decade, particularly focusing on residential charging patterns and their impact on power distribution networks. Research of Ahmad et al. (2020) provided a comprehensive review of electric load forecasting in vehicle-to-grid systems, highlighting the evolution from traditional statistical approaches to sophisticated machine learning solutions. Their analysis demonstrated the rapid growing importance of integrating various machine learning approaches to improve forecasting accuracy. Time series analysis has remained fundamental to load forecasting, as evidenced by Amini et al. (2016), who developed an ARIMA-based decoupled time series forecasting method for EV charging demand, specifically designed to address the stochastic nature of power system operations.

Deep learning approaches, particularly Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks, have demonstrated superior performance in capturing the temporal dependencies inherent in EV charging patterns. Chen et al. (2021) addressed a critical challenge in the field by developing an LSTM-based forecasting model capable of handling missing values in charging station data, demonstrating the robustness of deep learning approaches in real-world applications.

Hybrid approaches combining multiple methodologies have emerged as particularly promising solutions. Wang et al. (2022) proposed a hybrid deep learning framework that considers multiple influencing factors for EV charging load forecasting. Their framework demonstrated superior performance by integrating various data sources and accounting for multiple variables that affect charging patterns. This comprehensive approach showed significant improvements in forecasting accuracy compared to single-method approaches.

2 Technology

In this section, we will introduce all the technologies applied within the community initiative, "Forecasting of Electric Vehicle Charging Loads on Distribution Systems" in co-operation with Burlington Hydro Inc. We will show and explain all the models and methods which are mostly materialized with the programming language Python.

2.1 EMA and Wavelet Decomposition

The Exponential Moving Average (EMA) is a weighted moving average technique that places greater importance on more recent data points, making it effective for identifying trends in power or energy consumption. For instance, analyzing hourly power usage data can reveal spikes that indicate potential electric vehicle (EV) charging events. The EMA is calculated using the formula:

$$EMAt = \alpha \times Power\ Usaget + (1 - \alpha) \times EMAt - 1$$

where α is the smoothing factor, determining the rate at which older data points decrease in significance. By setting a deviation threshold, such as 20% above the EMA, unusual consumption patterns can be flagged in real-time. This method provides a dynamic way to monitor power usage and detect anomalies, as it continuously adapts to the latest data (Davis, 2016) (Hyndman, 2013).

Wavelet decomposition, on the other hand, breaks down a signal into its constituent components at various levels of resolution using wavelets. Wavelets are small waves that are scaled and shifted to capture both frequency and location information. One common type of wavelet used is the Daubechies wavelet ('db1'). The process involves decomposing the power usage signal into wavelet coefficients using a discrete wavelet transform (DWT), extracting approximation coefficients to represent the baseline trend, and reconstructing the signal using only these approximation coefficients. The mathematical representation of wavelet decomposition is:

$$f(t) \approx \sum j, kcj, k\psi j, k(t)$$

where $\psi j, k(t)$ are the wavelet basis functions, and cj, k are the wavelet coefficients. This method effectively isolates the underlying patterns from noise, allowing for the identification of significant deviations indicative of EV charging events (Mallat, 1999).

Both EMA and wavelet decomposition enhance the detection of EV charging behaviors by providing a robust framework for analyzing power usage data. The smooth wavelet baseline highlights significant power usage deviations, aiding in EV charging event detection. Both methods are beneficial in this case, and a combine application of both will provide a better result.

2.2 Clustering Algorithms

Clustering algorithms are a class of machine learning techniques designed to cluster/group up different sets of objects, in such a way that objects in the same cluster are more like each other than to those in other clusters. These algorithms can function with unlabeled data, which makes them particularly useful for distinguishing clusters corresponding to charging days and non-charging days in our dataset. The primary goal of clustering is to reveal the intrinsic structure of the data, which can be challenging due to subtle variations in hourly power consumption data. Given the complexity of setting a threshold to determine specific charging hours, clustering algorithms provide a holistic approach to separate charging days from non-charging days without the need for explicit labels.

Clustering techniques such as K-means, hierarchical clustering, and DBSCAN (Density-Based Spatial Clustering of Applications with Noise) are commonly used. K-means clustering, for example, partitions the data into K clusters by minimizing the variance within each cluster as shown in the **Fig 2.1** in below. Specifically for this 2-dimensianal 2-clusters task, the clustering steps are: 1) creating 2 centroids randomly; generating 2 clusters by assigning all the data points to their nearest centroids; 3) calculating new centroids for clusters; 4) reassigning the data points using the new centroids; 5) deciding whether to iterate from step 3 or to stop the loop if there are no new centroids or the gap in between is acceptably small. Hierarchical clustering builds a tree of clusters, whereas DBSCAN groups together points that are closely packed together, marking points that lie alone in low-density regions as outliers. These techniques enable us to identify patterns and trends that might not be immediately apparent through other forms of analysis.

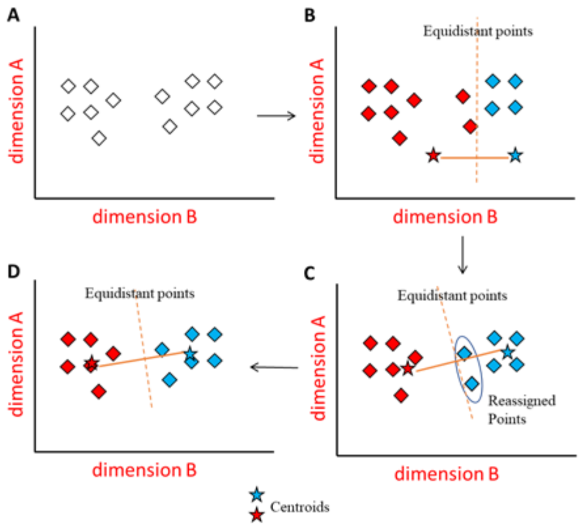


Fig 2.1 K-Means Steps (source: blopig.com/blog/2020/07/k-means-clustering-made-simple/)

To evaluate the performance of clustering algorithms comprehensively, several key evaluation metrics were employed. These metrics included: 1) Silhouette Score, which measures how similar an object is to its own cluster compared to other clusters; 2) Davies-Bouldin Index, which quantifies the average similarity ratio of each cluster with respect to its most similar one, indicating the algorithm's ability to form distinct clusters; and 3) Calinski-Harabasz Index, which assesses the ratio of the sum of between-cluster dispersion and within-cluster dispersion, thus providing an indication of the overall quality of the clustering. Mathematically,

$$Silhouette\ Score = \frac{1}{N}\sum_{i=1}^{N}\frac{b(i)-a(i)}{\max\big(a(i),b(i)\big)},$$

where N is the total number of all the data points, $a(i)$ is the average distance of the data point i to all the other points in its own cluster, representing the intra-cluster cohesion; and

$b(i)$ is the average distance of point i to all the points in the nearest neighboring cluster, representing the inter-cluster separation. Also,

$$Davies - Bouldin\ Index\ = \frac{1}{k}\sum_{i=1}^{k} \max_{j \neq i} \frac{s_i + s_j}{d_{ij}},$$

where k is the total number of all the clusters; s_i and s_j are the average intra-cluster distances of two clusters respectively denoted with i and j; while d_{ij} is the distance between the centroids of these two clusters. At last,

$$Calinski - Harabasz\ Index\ = \frac{\sum_{i=1}^{k} n_i \|c_i - c\|^2 /(k-1)}{\sum_{i=1}^{k} \sum_{x \in C_i} \|x - c_i\|^2 /(N-k)},$$

where N and k are the numbers of data points and clusters; n_i is the number of the data points within a cluster denoted with i, and c_i is the centroid of this cluster; c is the centroid of the entire dataset.

The greater Silhouette Score (values in [-1, 1]) and Calinski-Harabasz Index (positive values) indicate better clustering effects. The Silhouette Score measures how similar an object is to its own cluster compared to other clusters, with higher values signifying better-defined clusters. The Calinski-Harabasz Index evaluates the ratio of the sum of the between-clusters dispersion and of the within-cluster dispersion, with higher values suggesting that the clusters are dense and well-separated. Conversely, the Davies-Bouldin Index (positive values) is a metric where smaller values indicate better clustering. This index assesses the average similarity ratio of each cluster with the one that is most similar to the cluster itself, with lower values implying less similarity between clusters and therefore better clustering performance. By utilizing these evaluation metrics, the effectiveness and accuracy of each clustering algorithm were thoroughly assessed, allowing for a detailed comparison, and understanding of their performance on the dataset.

2.3 Time Series Analysis

The methods of Time Series Analysis are usually used for data indexed with timely or sequential orders. They are widely applied in forecasting stock prices, sales figures, temperature, etc. The residential data of power consumption is well segmented and recorded in hours, days, and months, which provides a suitable basis for time series analysis. Time Series Analysis treats the timely data with four main factors, respectively trend (T), seasonality (S), cyclicality (C), and randomness (White Noise). Trend is the long-term change pattern in the data. Seasonality is the periodic (or seasonal) fluctuation capturing regular data changes. Cyclicality is the longer-term cyclical change compared to seasonality. Randomness is the unexpected and unpredicted random variation. There are two important concepts in this analysis, autocorrelation and partial autocorrelation. Similar to the concept of correlation, autocorrelation describes the correlative relation between a k step lagged time series and the time series itself, denoted as ρ_k, and

$$\rho_k = \frac{Cov(X_t, X_{t-k})}{\sqrt{Var(X_t) \cdot Var(X_{t-k})}},$$

where $\{X_t\}$ is the time series. The partial autocorrelation ϕ_{kk} is the coefficient of the time series' regression on its k step lags shown in the equation:

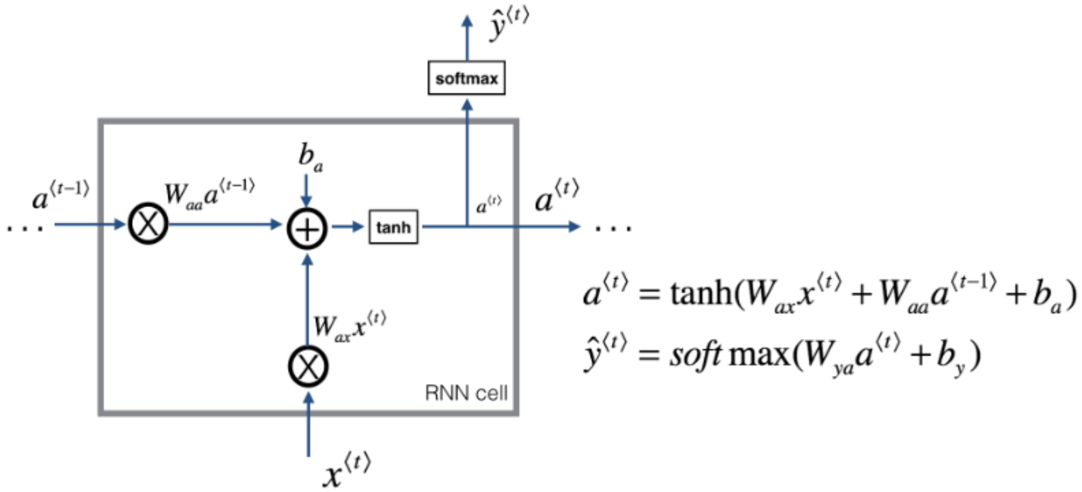
$$X_t = \phi_{k1}X_{t-1} + \phi_{k2}X_{t-2} + \cdots + \phi_{kk}X_{t-k} + \epsilon_t,$$

where ϵ_t is the white noise error fitting a 0-mean normal distribution.

2.4 Deep Learning Models

Recurrent Neural Networks (RNNs) represent a fundamental architecture in deep learning as shown in the **Fig 2.2**, specifically designed to handle sequential data. Unlike feedforward neural networks, RNNs maintain an internal state (memory) that allows them to process sequences of inputs by incorporating information from previous timesteps.

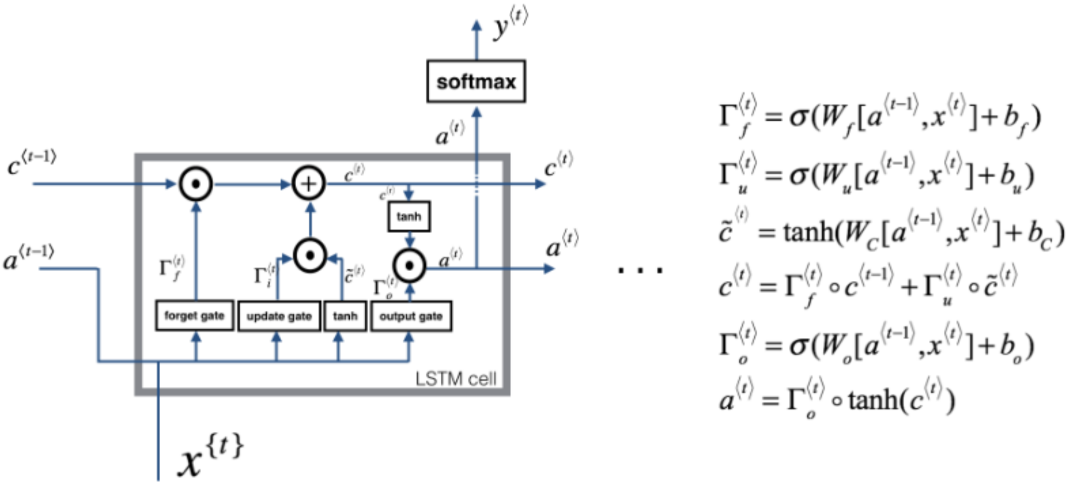
However, traditional RNNs suffer from the vanishing gradient problem, leading to the development of Long Short-Term Memory (LSTM) networks as a more robust alternative. LSTM networks address the vanishing gradient problem through a more complex architecture incorporating memory cells and gating mechanisms by adding Forget Gate, Input Gate, Output Gate as shown in the **Fig 2.3** in below.



$\hat{y}^{\langle t \rangle}$
softmax
b_a
$a^{\langle t-1 \rangle}$
$W_{aa}a^{\langle t-1 \rangle}$
tanh
$a^{\langle t \rangle}$
$a^{\langle t \rangle}$
$W_{ax}x^{\langle t \rangle}$
RNN cell
$x^{\langle t \rangle}$

$$a^{\langle t \rangle} = \tanh(W_{ax}x^{\langle t \rangle} + W_{aa}a^{\langle t-1 \rangle} + b_a)$$
$$\hat{y}^{\langle t \rangle} = soft\max(W_{ya}a^{\langle t \rangle} + b_y)$$

Fig 2.2 RNN Cell Structure



$$\Gamma_f^{\langle t \rangle} = \sigma(W_f[a^{\langle t-1 \rangle}, x^{\langle t \rangle}] + b_f)$$
$$\Gamma_u^{\langle t \rangle} = \sigma(W_u[a^{\langle t-1 \rangle}, x^{\langle t \rangle}] + b_u)$$
$$\tilde{c}^{\langle t \rangle} = \tanh(W_C[a^{\langle t-1 \rangle}, x^{\langle t \rangle}] + b_C)$$
$$c^{\langle t \rangle} = \Gamma_f^{\langle t \rangle} \circ c^{\langle t-1 \rangle} + \Gamma_u^{\langle t \rangle} \circ \tilde{c}^{\langle t \rangle}$$
$$\Gamma_o^{\langle t \rangle} = \sigma(W_o[a^{\langle t-1 \rangle}, x^{\langle t \rangle}] + b_o)$$
$$a^{\langle t \rangle} = \Gamma_o^{\langle t \rangle} \circ \tanh(c^{\langle t \rangle})$$

Fig 2.3 LSTM Cell Structure

We chose to implement both RNN and LSTM architectures for this prediction task for several key reasons:

- Temporal Pattern Recognition: RNNs excel at capturing sequential patterns in time series data, making them particularly suitable for analyzing historical charging load data. Their ability to maintain an internal state allows them to identify recurring patterns in daily and weekly charging behaviors.
- Long-term Dependencies: LSTM networks, with their sophisticated gating mechanisms, are especially effective at capturing long-term dependencies in charging patterns, such as weekly cycles or seasonal variations. This is crucial for accurate long-term load forecasting.
- Variable Input Length: Both architectures can handle variable-length input sequences, allowing our model to adapt to different prediction timeframes and data availability scenarios.

3 Charging Event Detecting

In this section, we will introduce 3 technologies for the application of charging event detecting. For this application, the primary issue we encountered is a biased dataset. As illustrated in the **Fig 3.1**, more than 90% of the labeled data is categorized as "1." Consequently, the model predominantly learns the pattern of "1" and tends to produce predictions favoring "1."

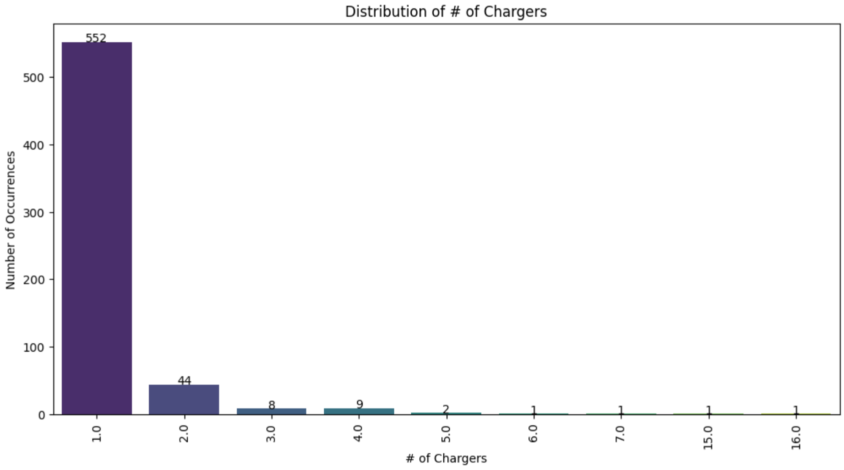


Fig 3.1 Distribution of EV Charger Number

3.1 Exponential Moving Average (EMA) and Wavelet Decomposition

To address the data bias issues, we applied EMA and Wavelet decomposition to provide a base line on 0 (no EV chargers) and 1 (with one EV charger), by filtering out the date when the labelled data households are not charging their EVs. The extracting logic is intuitive and straight: if the selected day is available, it extracts the hourly power usage.

Using wavelet transforms, it decomposes the data to obtain a smoothed baseline and calculates an Exponential Moving Average (EMA) for trend detection. Spikes are identified when the power usage exceeds the EMA by a threshold of 0.5, and the neighboring hours and changing significantly. And plateaus are detected when usage remains above the wavelet baseline by a threshold of 1, for a specified duration.

3.2 Clustering

3.2.1 Initial Clustering Result

Because of the subtle changes of the hourly power consumption data series, we had unprecedented difficulties in analyzing the hourly data (such as setting a threshold to determine if the user is charging the EV in one specific hour). We decided to apply clustering algorithms to bypass this obstacle. The specific technical route is applying clustering to divide the possible charging days and non-charging days holistically. We segmented the dataset into 12 months to address the seasonal variations in power consumption. This approach aimed to mitigate the influence of relatively higher power consumption in the summer months, which could skew clustering results. The segmentation allowed for more accurate analysis of consumption patterns within each specific period, avoiding the distortion that could arise from combining data across different seasons. Clustering was performed separately for each month's data to ensure the results were relevant to the specific time of year.

KMeans, Hierarchical, DBSCAN, MeanShift, GMM, Spectral, and OPTICS clustering algorithms were applied to the dataset. Each of these algorithms was selected due to its unique characteristics and ability to handle different types of data distributions and clustering challenges. All the clustering algorithms are assessed and evaluated using 3 metrics, and the results are shown in the following **Table 3.1**.

Table 3.1 Evaluation of Clustering Algorithms

Model Name	**Silhouette Score**	**Davies-Bouldin Index**	**Calinski-Harabasz Index**
KMeans	0.5149	0.8143	464.5123
Hierarchical	0.5149	).8143	464.5123
DBSCAN	Only one cluster found		
MeanShift	0.3948	1.0216	64.9731
GMM	0.5101	0.8220	451.8174
Spectral	0.5149	0.8143	464.5123
OPTICS	0.3930	1.2473	293.1408

Among all these clustering algorithms, KMeans and Hierarchical stand out based on these metrics. The KMeans algorithm achieved a Silhouette Score of 0.5149, a Davies-Bouldin Index of 0.8143, and a Calinski-Harabasz Index of 464.5123, indicating well-defined and distinct clusters. Similarly, the Hierarchical clustering algorithm showed identical performance metrics to KMeans, suggesting that it also effectively identified clear and distinct clusters within the data. On the other hand, the DBSCAN algorithm only found one cluster, which indicates that it was not effective for this dataset. MeanShift and OPTICS clustering algorithms showed relatively lower performance, with Silhouette Scores of 0.3948 and 0.3930, respectively, and higher Davies-Bouldin Index values, suggesting less distinct and more overlapping clusters. The GMM algorithm, while performing slightly better than MeanShift and OPTICS with a Silhouette Score of 0.5101 and a Davies-Bouldin Index of 0.8220, still did not outperform KMeans and Hierarchical clustering. Spectral clustering mirrored the performance of KMeans and Hierarchical, with a Silhouette Score of 0.5149, a Davies-Bouldin Index of 0.8143, and a Calinski-Harabasz Index of 464.5123, reinforcing the effectiveness of these methods in producing high-quality clustering results.

To visualize the clustering results, we applied KMeans algorithm to the EV registered user's January dataset with 551 unique locations and 17080 records as an example and combine the consumption of 24 hours (dimensions) into 3 axes (x: 0:00-8:00, y: 8:00-16:00, and z: 16:00-24:00) to show the clustered data points as shown in the **Fig 3.2**.

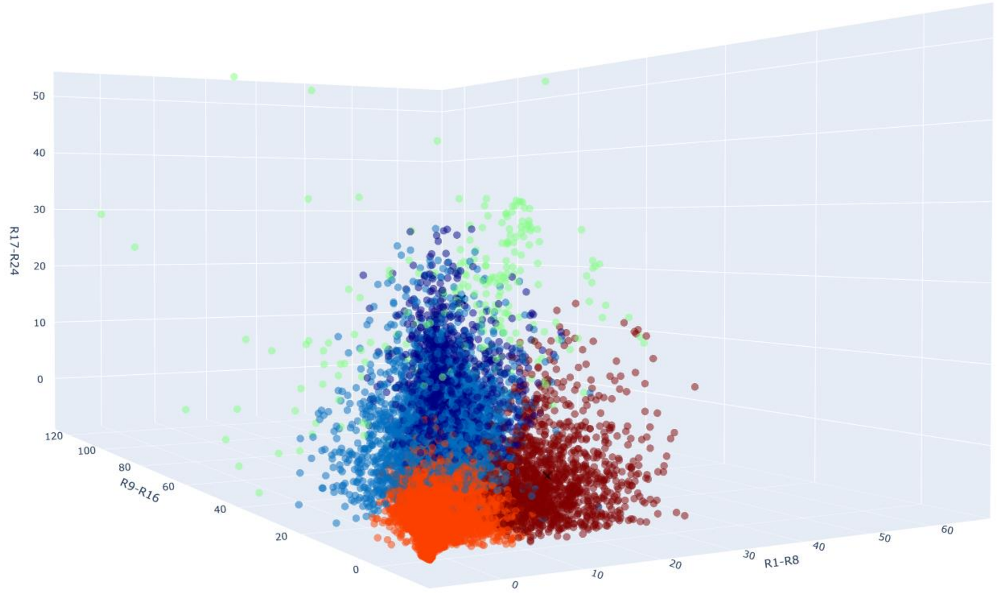
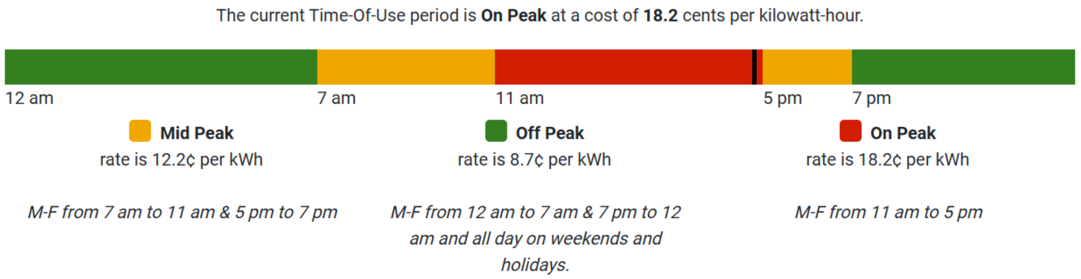


Fig 3.2 Result of Initial Clustering

Noting that in the figure shown above, we adjusted the number of clusters to achieve better separation and obtained some "by-product" results and insights, which will be elaborated in the latter part. In this case, there are 5 different clusters including: 1) higher consumption in the first 8 hours taking the percentage of 8.29% in all the records; 2) higher consumption in the middle 8 hours taking 19.18%; 3) high consumption in the last 8 hours taking 11.64%; 4) high consumption in the whole day taking 1.32%; and 5) low consumption in the whole day taking the rest 59.57%.

3.2.2 New Method for Visualization

Further, we applied a new combination method for the 24 dimensions, which is to divide one day into Off-peak, Mid-peak, and On-peak hours aligning with different time-of-use prices shown in the **Fig 3.3** to achieve a better clustering result and visualization effect.


The current Time-Of-Use period is **On Peak** at a cost of **18.2** cents per kilowatt-hour.

🟧 **Mid Peak**
rate is 12.2¢ per kWh

M-F from 7 am to 11 am & 5 pm to 7 pm

🟩 **Off Peak**
rate is 8.7¢ per kWh

M-F from 12 am to 7 am & 7 pm to 12 am and all day on weekends and holidays.

🟥 **On Peak**
rate is 18.2¢ per kWh

M-F from 11 am to 5 pm

Fig 3.3 Hour Separation with Different Time-of-Use Prices

Following the previous example, with this new combination method, January's records can be clustered into 4 groups including: 1) "Cost Sensitive" users with high consumption in the off-peak hours in this specific day; 2) "Insensitive" users with average consumption in the off-, mid-, and on-peak hours; 3) "Thrifty" users with electricity saving pattern with approximately 1 kwh in all hours; and 4) "Consuming" users with highly consuming pattern with over 100kwh each day. The **Fig 3.4** in below shows the portion of these 4 clusters in January.

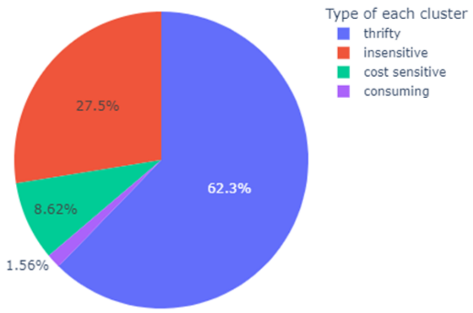


Fig 3.4 Portion of Clusters for January

For each month's dataset, we set the clustering parameter to 3 or 4 clusters. This number was chosen based on the observed variability in consumption patterns within each period. The visualization of data points clustering for January is shown as the **Fig 3.5** in below, and the visualizations for the clustering results for the other months in year 2023 are shown in the appendix.

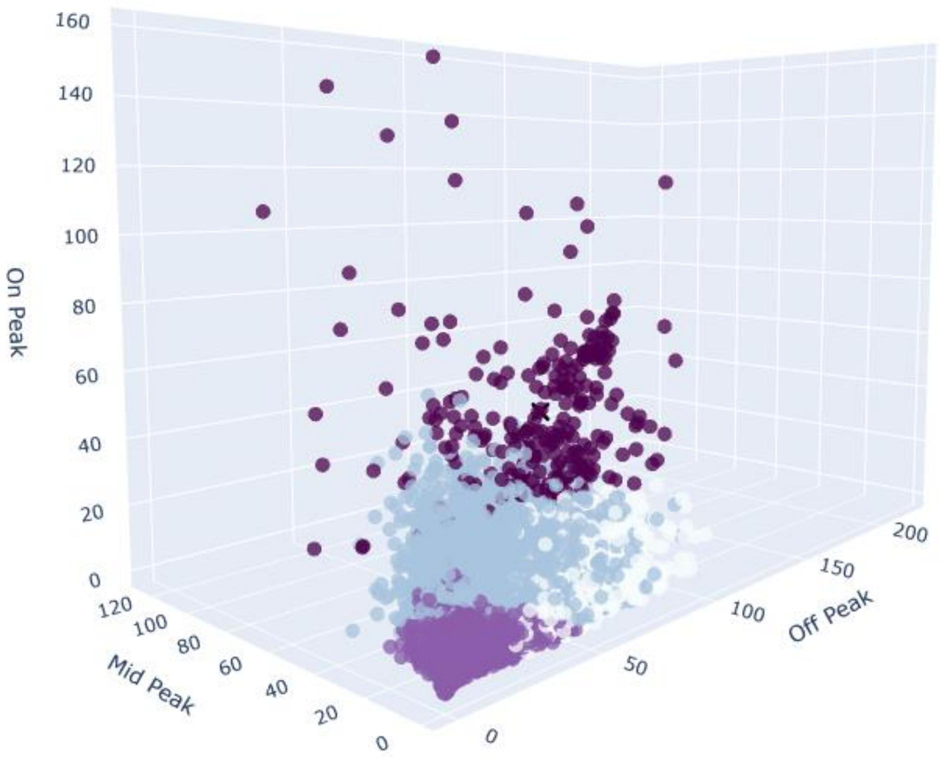
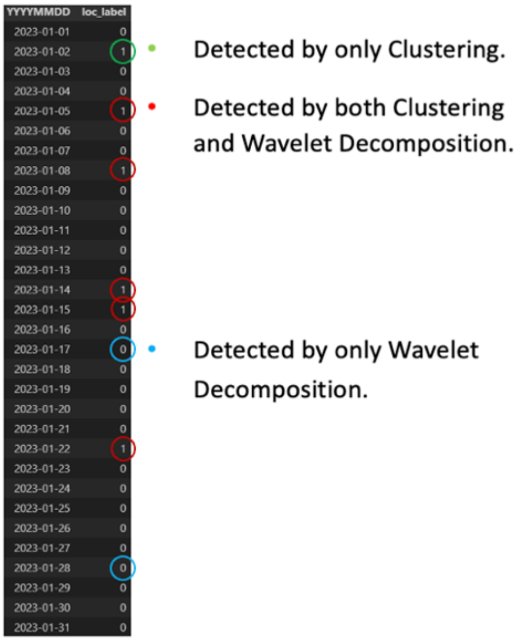


Fig 3.5 Result of Clustering with New Combination

3.2.3 Two Step Clustering and Cross Validation

We figured out that the clustering algorithms helped to identify one user’s overall power consumption pattern of one particular day instead off the charging behavior pattern, thus after the initial clustering, we continued to perform a second-step clustering with the hypothesis that charging days always exhibit higher consumption than non-charging days. To substantiate this hypothesis, the second-step clustering aimed to refine and verify the initial results. This step was crucial as it provided a deeper layer of analysis, distinguishing between days with high consumption typically associated with EV charging events and those without. This involved an intricate process of cross validating the clustering results with outcomes derived from Wavelet Decomposition, a technique used to break down and analyze the power usage signal into its constituent frequencies. Additionally, a Threshold Policy was applied, which set specific benchmarks for power usage to differentiate charging events from regular consumption patterns. By integrating these methods, we enhanced the confidence and accuracy in detecting actual charging days as shown as the **Fig 3.6** in below.

The integration of these advanced techniques ensured that the identified charging patterns were robust and reliable, addressing potential anomalies and inconsistencies in the data, and providing a comprehensive understanding of the EV charging behaviors. This multi-faceted approach reinforced the validity of our findings and demonstrated the effectiveness of using multiple analytical layers to achieve precise and dependable results.



YYYYMMDD	loc_label
2023-01-01	0
2023-01-02	1
2023-01-03	0
2023-01-04	0
2023-01-05	1
2023-01-06	0
2023-01-07	0
2023-01-08	1
2023-01-09	0
2023-01-10	0
2023-01-11	0
2023-01-12	0
2023-01-13	0
2023-01-14	1
2023-01-15	1
2023-01-16	0
2023-01-17	0
2023-01-18	0
2023-01-19	0
2023-01-20	0
2023-01-21	0
2023-01-22	1
2023-01-23	0
2023-01-24	0
2023-01-25	0
2023-01-26	0
2023-01-27	0
2023-01-28	0
2023-01-29	0
2023-01-30	0
2023-01-31	0

- Detected by only Clustering.
- Detected by both Clustering and Wavelet Decomposition.
- Detected by only Wavelet Decomposition.

Fig 3.6 Cross Validation of Results

3.3 Semi-Supervised Learning

After the labelled dataset has been processed from 3.1, it can be merged with the unlabelled data for the training of Semi-supervised learning.

Semi-supervised learning bridges supervised and unsupervised learning by utilizing both labeled and unlabeled data. It's particularly valuable when labeled data is much smaller than the unlabeled data. In this case, the data of known having EV chargers is less than 10% of the total dataset, semi-supervised learning will be suitable. By leveraging both labeled and unlabeled data simultaneously, SSL aims to achieve better model performance than would be possible using labeled data alone as shown in the **Fig 3.7**.

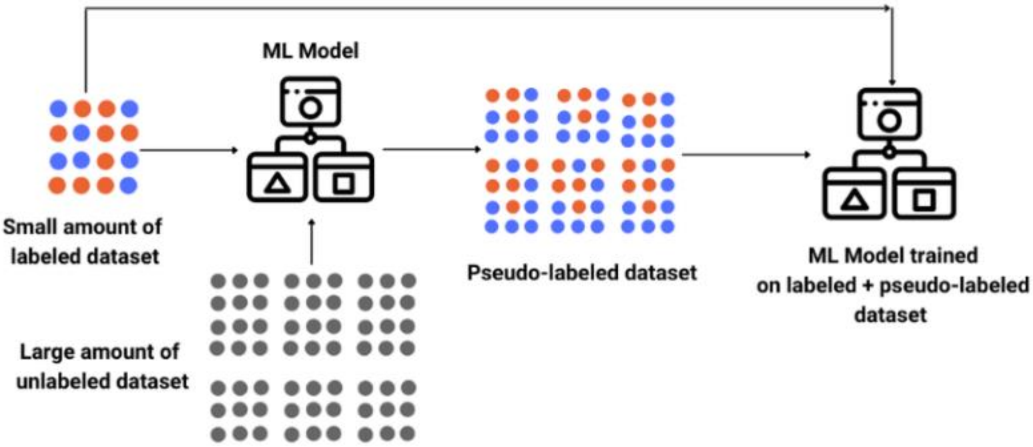

ML Model
Small amount of labeled dataset
Large amount of unlabeled dataset
Pseudo-labeled dataset
ML Model trained on labeled + pseudo-labeled dataset

Fig 3.7 Semi-Supervised Learning Illustration

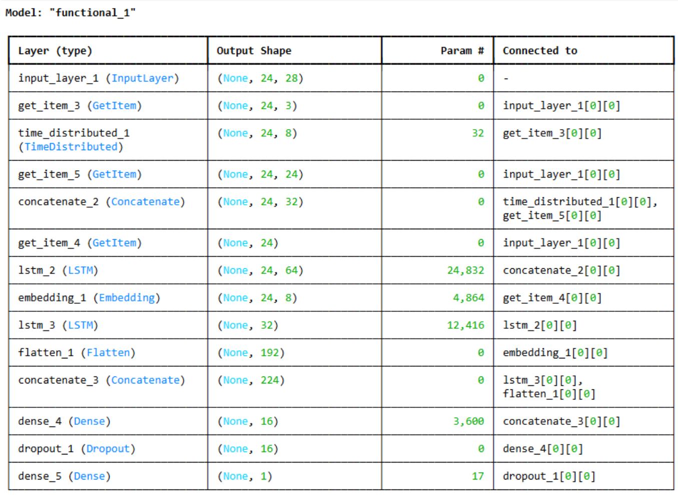
3.3.1 Methodology Implementation and Algorithm Design

The implementation focuses on the self-training paradigm, a fundamental approach in semi-supervised learning. This iterative method begins with a small set of labeled data and progressively expands its training set by incorporating high-confidence predictions on unlabeled data. The algorithm's implementation consists of several key components, structured within an iterative framework.

The core algorithm operates through a predefined number of iterations of 5, with each iteration comprising four main phases: model training, prediction, confident sample selection, and dataset updating. The process is initialized with a confidence threshold of 0.8, which serves as the criterion for selecting reliable predictions.

3.3.2 Model Training and Optimization

The model we use is a deep learning model, with structure as shown in **Fig 3.8**. The training phase incorporates several optimization strategies. Early stopping is implemented to prevent overfitting, monitoring the validation loss with a patience parameter of 5 epochs. This mechanism ensures model robustness by preserving the best-performing weights while preventing unnecessary computational overhead from overtraining.



Model: "functional_1"

Layer (type)	Output Shape	Param #	Connected to
input_layer_1 (InputLayer)	(None, 24, 28)	0	-
get_item_3 (GetItem)	(None, 24, 3)	0	input_layer_1[0][0]
time_distributed_1 (TimeDistributed)	(None, 24, 8)	32	get_item_3[0][0]
get_item_5 (GetItem)	(None, 24, 24)	0	input_layer_1[0][0]
concatenate_2 (Concatenate)	(None, 24, 32)	0	time_distributed_1[0][0], get_item_5[0][0]
get_item_4 (GetItem)	(None, 24)	0	input_layer_1[0][0]
lstm_2 (LSTM)	(None, 24, 64)	24,832	concatenate_2[0][0]
embedding_1 (Embedding)	(None, 24, 8)	4,864	get_item_4[0][0]
lstm_3 (LSTM)	(None, 32)	12,416	lstm_2[0][0]
flatten_1 (Flatten)	(None, 192)	0	embedding_1[0][0]
concatenate_3 (Concatenate)	(None, 224)	0	lstm_3[0][0], flatten_1[0][0]
dense_4 (Dense)	(None, 16)	3,600	concatenate_3[0][0]
dropout_1 (Dropout)	(None, 16)	0	dense_4[0][0]
dense_5 (Dense)	(None, 1)	17	dropout_1[0][0]

Fig 3.8 LSTM Model Structure

3.3.3 Semi-Supervised Learning Result

The semi-supervised learning model for EV charger detection yielded several key findings across different analytical dimensions. The results are presented through multiple evaluation metrics and visualization. Our semi-supervised learning model for EV charger detection demonstrated robust prediction capabilities across diverse scenarios. The overall performance metrics (MSE: 0.3735, RMSE: 0.6112, MAE: 0.4535) indicate that the model achieves reasonable accuracy in predicting charger presence. Referring to the following **Fig 3.9**, it indicates that the model's prediction is neither overfitting nor under-performing.

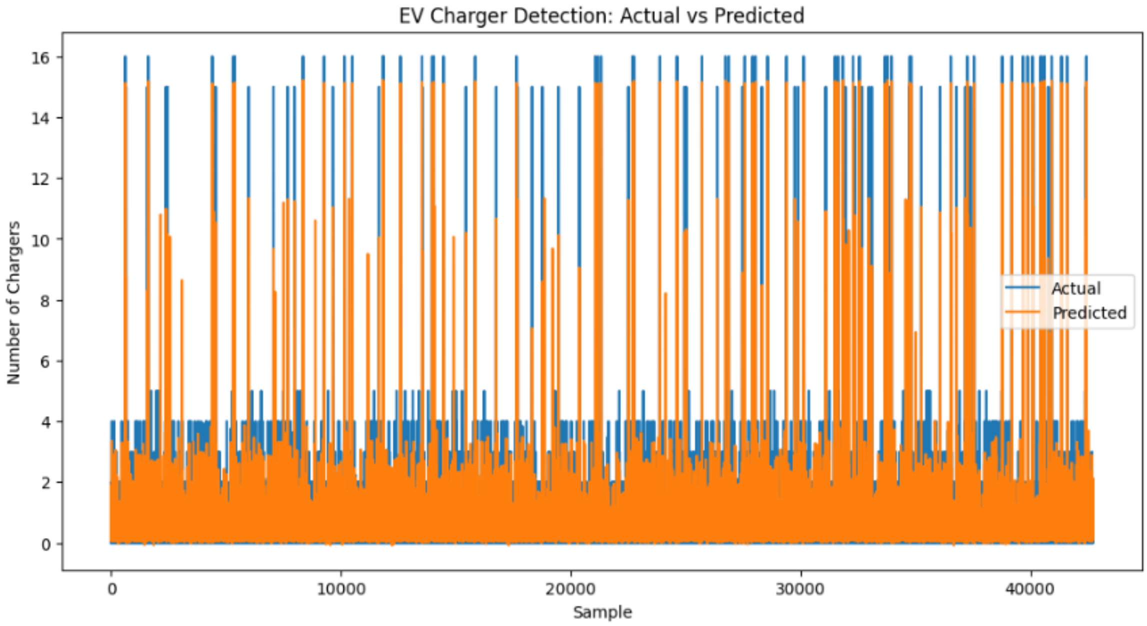
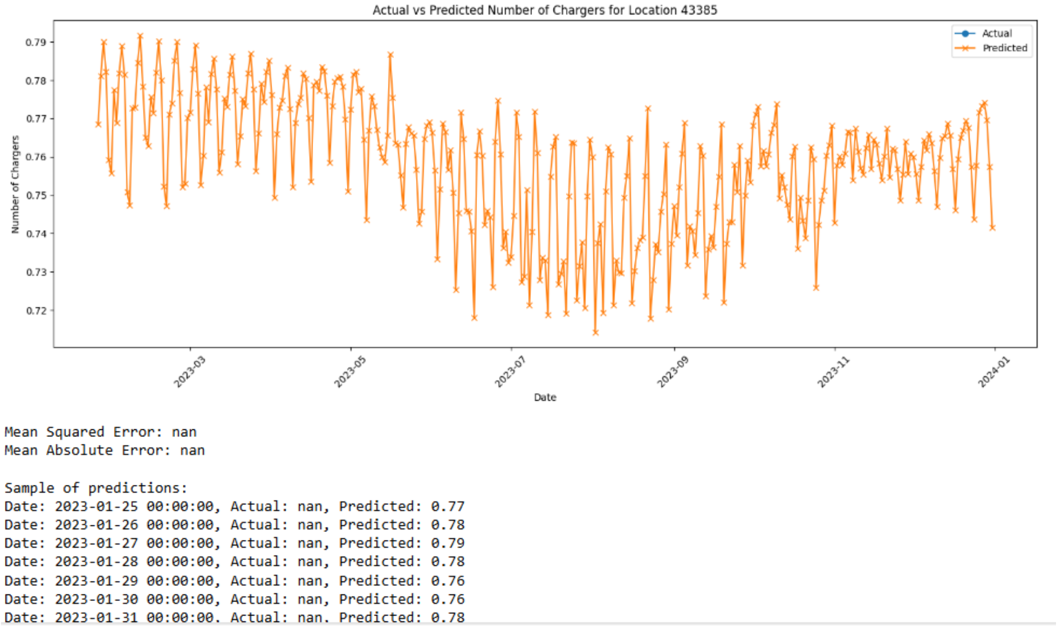


Fig 3.9 Semi-Supervised Learning Illustration

Our analysis also reveals distinctive prediction patterns across different locations. The below two **Fig 3.10** contrasted cases of Location 43385 and Location 295, these cases demonstrate the model's ability to differentiate between areas with high and low probabilities of EV charging infrastructure presence. Location 43385 exhibits remarkably consistent high-probability predictions for EV charger presence, despite the absence of labeled data. The predictions maintain a stable range between 0.72 and 0.79, suggesting strong confidence in the presence of charging infrastructure.



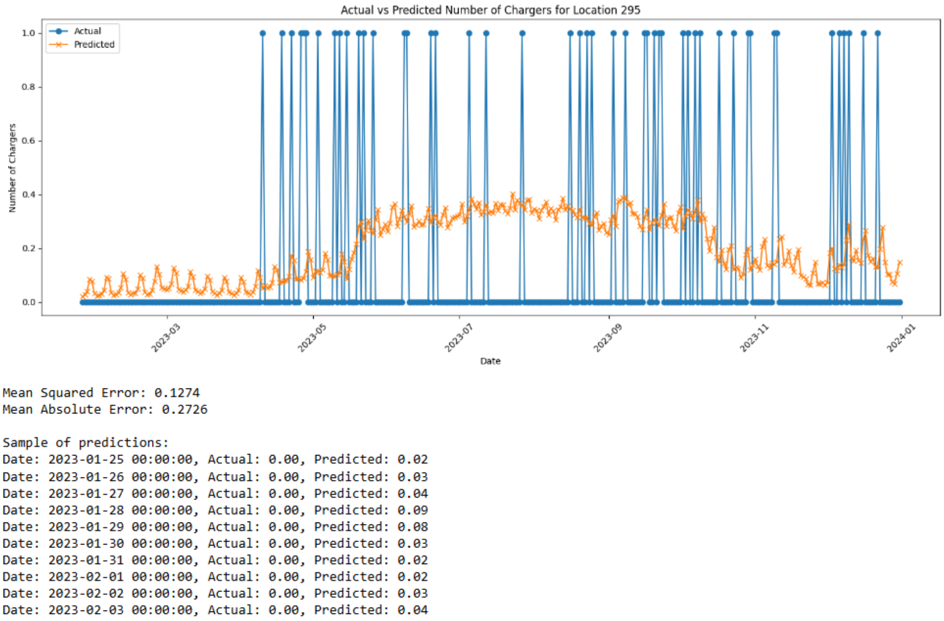
```
Mean Squared Error: nan
Mean Absolute Error: nan

Sample of predictions:
Date: 2023-01-25 00:00:00, Actual: nan, Predicted: 0.77
Date: 2023-01-26 00:00:00, Actual: nan, Predicted: 0.78
Date: 2023-01-27 00:00:00, Actual: nan, Predicted: 0.79
Date: 2023-01-28 00:00:00, Actual: nan, Predicted: 0.78
Date: 2023-01-29 00:00:00, Actual: nan, Predicted: 0.76
Date: 2023-01-30 00:00:00, Actual: nan, Predicted: 0.76
Date: 2023-01-31 00:00:00, Actual: nan, Predicted: 0.78
```

Fig 3.10 Actual vs. Detected Charging Event (for Example Location 43385)

The stability and high confidence of these predictions suggest that this location possesses characteristics strongly associated with EV charging infrastructure. These might include factors such as proximity to commercial areas, high traffic volumes, or demographic patterns typical of EV adoption. The cyclic patterns in the predictions further support this interpretation, as they align with expected usage patterns in areas with established charging infrastructure.

In contrast, Location 295 presents a markedly different pattern, with the model generating consistently low probability predictions ranging from 0.02 to 0.40 as shown in the **Fig 3.11**. These predictions are validated by actual data showing zero chargers present, resulting in a Mean Squared Error of 0.1274 and a Mean Absolute Error of 0.2726. The model's conservative predictions, staying predominantly below 0.4, demonstrate its ability to accurately identify locations unlikely to have charging infrastructure.

The prediction pattern for Location 295 shows slight variations over time but maintains a consistent low-probability range. This stability in low predictions, combined with the confirmation from actual data, validates the model's ability to identify areas without EV charging infrastructure. The low error rates further support the model's accuracy in such scenarios.



```
Mean Squared Error: 0.1274
Mean Absolute Error: 0.2726

Sample of predictions:
Date: 2023-01-25 00:00:00, Actual: 0.00, Predicted: 0.02
Date: 2023-01-26 00:00:00, Actual: 0.00, Predicted: 0.03
Date: 2023-01-27 00:00:00, Actual: 0.00, Predicted: 0.04
Date: 2023-01-28 00:00:00, Actual: 0.00, Predicted: 0.09
Date: 2023-01-29 00:00:00, Actual: 0.00, Predicted: 0.08
Date: 2023-01-30 00:00:00, Actual: 0.00, Predicted: 0.03
Date: 2023-01-31 00:00:00, Actual: 0.00, Predicted: 0.02
Date: 2023-02-01 00:00:00, Actual: 0.00, Predicted: 0.02
Date: 2023-02-02 00:00:00, Actual: 0.00, Predicted: 0.03
Date: 2023-02-03 00:00:00, Actual: 0.00, Predicted: 0.04
```

Fig 3.11 Actual vs. Detected Charging Event (for Example Location 295)

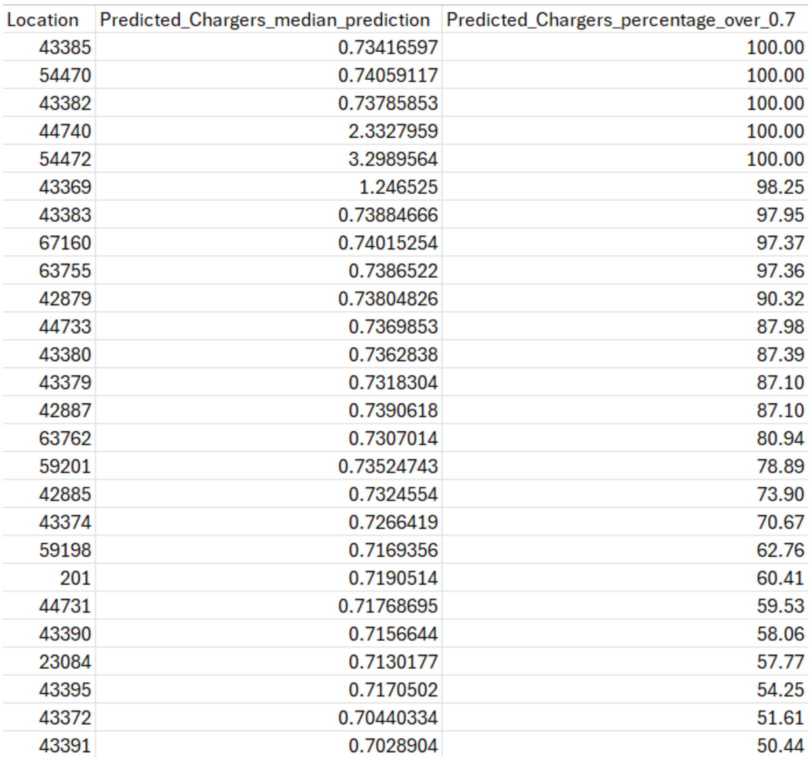
A note for the actual data is that, with the EMA and Wavelet Decomposition analysis, it indicates that although the location is marked as having an EV charger, they do not use it very often. Therefore the “label 0” outnumbers “label 1” in this specific location.

To summarize the result, the model gave following statistic:
Total number of locations: 142
Mean percentage over 0.7 across all locations: 20.92%
Median percentage over 0.7 across all locations: 1.17%
Locations with >35% predictions above 0.7: 39
Locations with <35% predictions above 0.7: 103

Our analysis employs a significant threshold criterion for determining the likelihood of EV charger presence across different locations. Specifically, locations where more than 35% of predictions exceed 0.7 are classified as likely to have EV charging infrastructure. Analysis across all 142 locations reveals a clear distinction in prediction patterns. Of these locations, 39 (27.5%) exceed the threshold criterion of having more than 35% of their predictions above 0.7, while 103 locations (72.5%) fall below this threshold. This distribution suggests a concentrated pattern of EV charging infrastructure, aligned with expected urban and commercial development patterns.

3.3.4 Result Prototype

As the prototype result, the model will generate an excel sheet that contain a consolidate result for all location, as shown in **Fig 3.12.** The table displays several columns including *Location*, *Predicted Chargers mean prediction*, and *Predicted Chargers percentage over 0.7*. The *Location* indicates the household by the location number, *Predicted Chargers mean prediction* indicates the average number of EV chargers were predicted in this household, and *Predicted Chargers percentage over 0.7* indicates the percentage of the date in a year where found high chance (>0.7) of EV charging event. For example, the first row means that in Location 43385, the household have 0.73 EV chargers, and 100% of the date support this data.



Location	Predicted_Chargers_median_prediction	Predicted_Chargers_percentage_over_0.7
43385	0.73416597	100.00
54470	0.74059117	100.00
43382	0.73785853	100.00
44740	2.3327959	100.00
54472	3.2989564	100.00
43369	1.246525	98.25
43383	0.73884666	97.95
67160	0.74015254	97.37
63755	0.7386522	97.36
42879	0.73804826	90.32
44733	0.7369853	87.98
43380	0.7362838	87.39
43379	0.7318304	87.10
42887	0.7390618	87.10
63762	0.7307014	80.94
59201	0.73524743	78.89
42885	0.7324554	73.90
43374	0.7266419	70.67
59198	0.7169356	62.76
201	0.7190514	60.41
44731	0.71768695	59.53
43390	0.7156644	58.06
23084	0.7130177	57.77
43395	0.7170502	54.25
43372	0.70440334	51.61
43391	0.7028904	50.44

Fig 3.12 Partial Result for all location

4 Application: Charging Loads Forecasting

In this section, we will introduce time series analysis and deep learning methods to forecast the EV charging loads. The time series analysis provides a good comparable benchmark for the deep learning method in this application.

4.1 Time Series Analysis

The power consumption data's average level provides a basic understanding for us to the overall operation condition. On a daily basis and a hourly basis, the time series data is respectively shown in the following **Fig 4.1** and **Fig 4.2**.

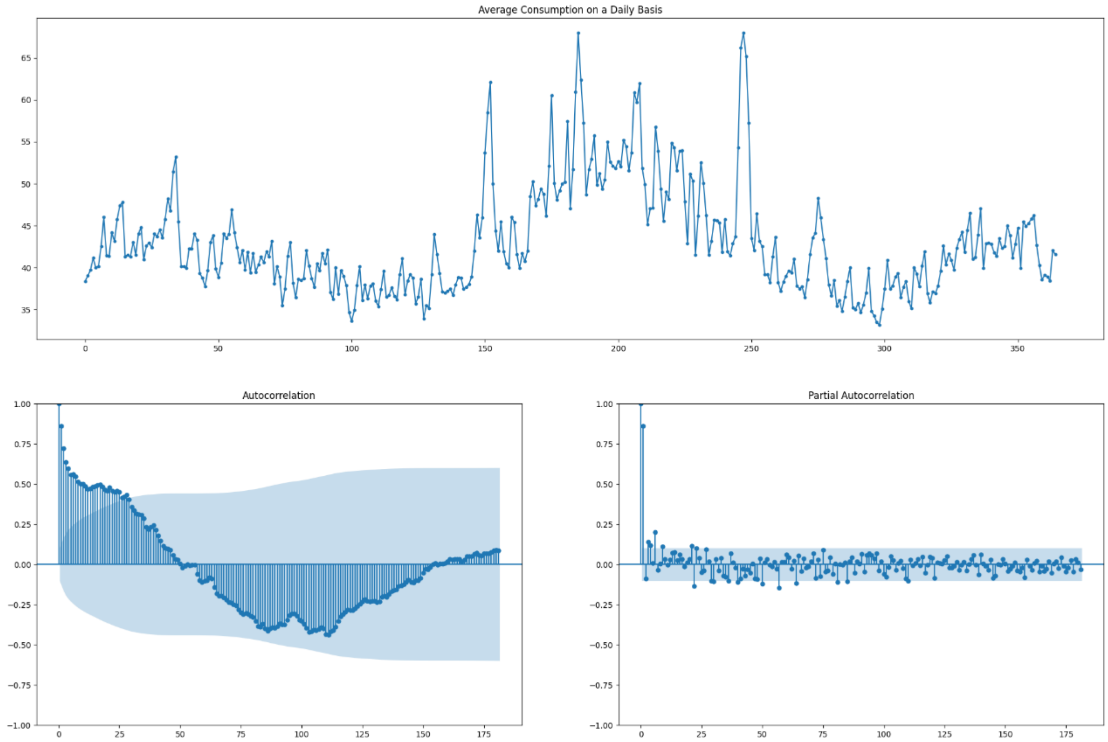


Fig 4.1 Daily Average Power Consumption

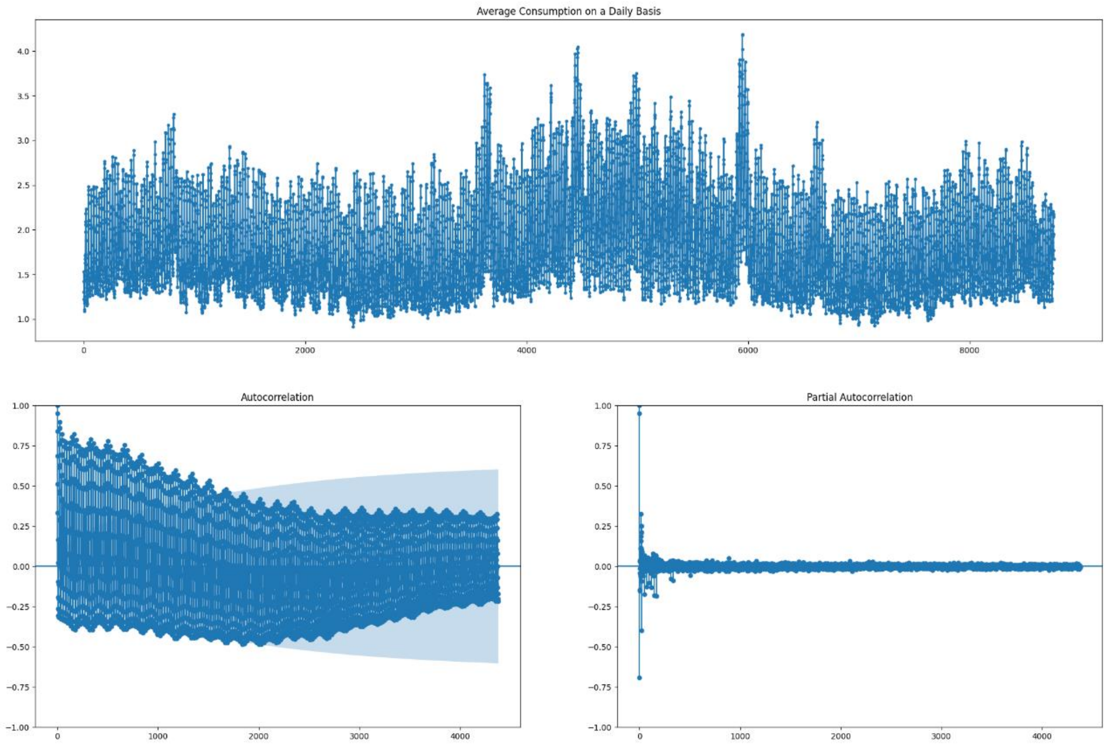
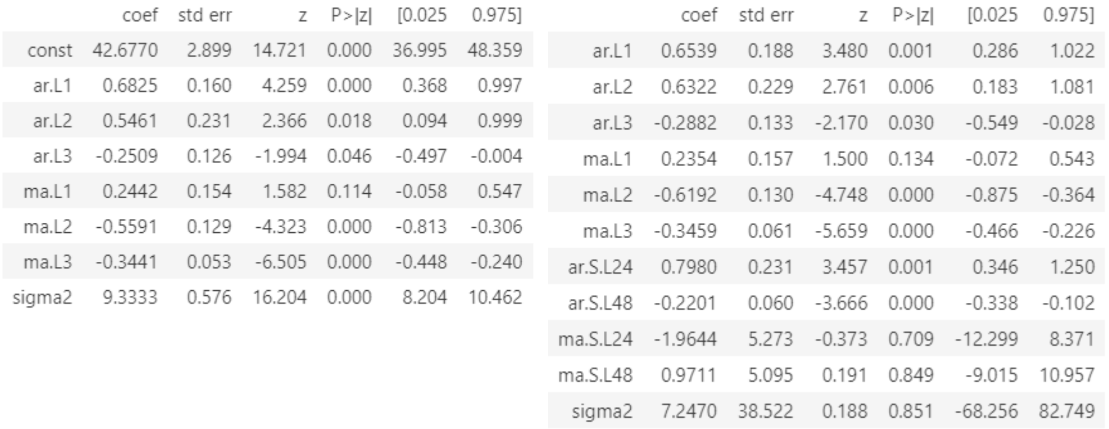


Fig 4.2 Hourly Average Power Consumption

The Seasonal Autoregressive Integrated Moving Average model (SARIMA) is used in analyzing power usage data in our experiment. It can be considered as a specially combined regression model. As its name suggests, SARIMA contains autoregression, differencing, and moving averages. The model firstly applies one-step and s-step differential (s denotes the length of seasonal periods) to the original data. Secondly, the model regresses the data onto the historical data and white noises. We manually selected and collected all the power consumption records with registered electric vehicles. Both on a daily and hourly basis, we calculated the average power usage and denoted them as d_t and h_t. It is worth mentioning that we used independently and identically distributed white noises $\{\varepsilon_t\}$ to fit the randomness in data.

The fitted results for daily and hourly series are shown in the following **Tables 4.1** and **4.2**.

Table 4.1 (Left) and **4.2 (Right)** SARIMA Model Fitting Results



	coef	std err	z	P>|z|	[0.025	0.975]
const	42.6770	2.899	14.721	0.000	36.995	48.359
ar.L1	0.6825	0.160	4.259	0.000	0.368	0.997
ar.L2	0.5461	0.231	2.366	0.018	0.094	0.999
ar.L3	-0.2509	0.126	-1.994	0.046	-0.497	-0.004
ma.L1	0.2442	0.154	1.582	0.114	-0.058	0.547
ma.L2	-0.5591	0.129	-4.323	0.000	-0.813	-0.306
ma.L3	-0.3441	0.053	-6.505	0.000	-0.448	-0.240
sigma2	9.3333	0.576	16.204	0.000	8.204	10.462

	coef	std err	z	P>|z|	[0.025	0.975]
ar.L1	0.6539	0.188	3.480	0.001	0.286	1.022
ar.L2	0.6322	0.229	2.761	0.006	0.183	1.081
ar.L3	-0.2882	0.133	-2.170	0.030	-0.549	-0.028
ma.L1	0.2354	0.157	1.500	0.134	-0.072	0.543
ma.L2	-0.6192	0.130	-4.748	0.000	-0.875	-0.364
ma.L3	-0.3459	0.061	-5.659	0.000	-0.466	-0.226
ar.S.L24	0.7980	0.231	3.457	0.001	0.346	1.250
ar.S.L48	-0.2201	0.060	-3.666	0.000	-0.338	-0.102
ma.S.L24	-1.9644	5.273	-0.373	0.709	-12.299	8.371
ma.S.L48	0.9711	5.095	0.191	0.849	-9.015	10.957
sigma2	7.2470	38.522	0.188	0.851	-68.256	82.749

Though all parameters are statistically significant, the regression is highly dependent on the randomness part. The SARIMA model is limited in the explanatory and predictive power. As shown in the **Fig 4.3** in below, we can see the fitted orange line for daily average power consumption is close to the blue original one. However, there is a much bigger gap between the green forecasted results and the blue true values.

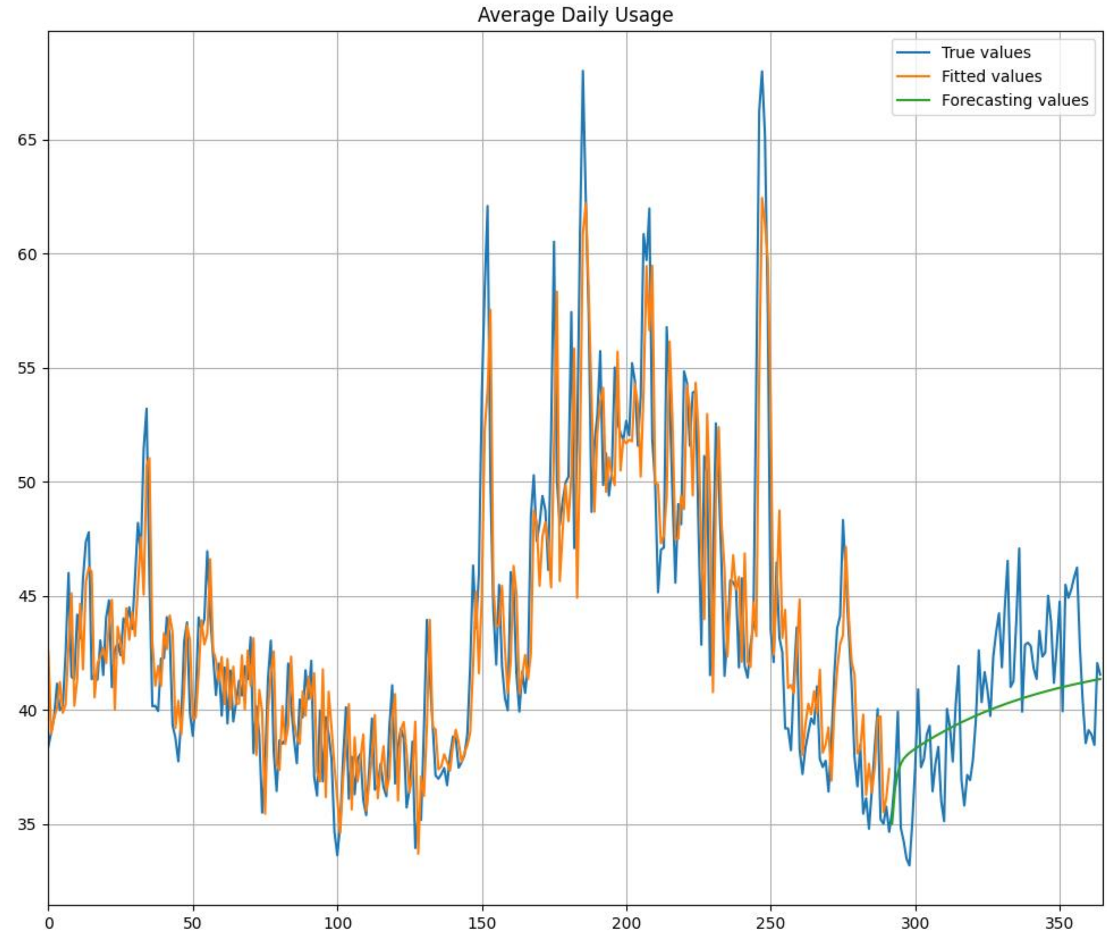


Fig 4.3 Fitted Result of SARIMA Model

To conclude, Time Series Analysis is feasible but not pretty suitable for residential records. Because the electricity usage for charging EV is relatively stable in one family. No family is likely to buy new EVs then sell some in our current weekly-scaled data. On the other hand, Time Series Analysis could be more effective in forecasting for public chargers as the paper shows. (K. C. Akshay, 2024). On the plus side, Time Series Analysis provided us a new perspective to understand the original data better. Further on, more complex models with bigger fixed ranks can be explored afterwards to discover patterns in data changes. Also, results from SARIMA provided us a comparable benchmark for the Deep Learning model in this forecasting task.

4.2 Deep Learning Method

To achieve more accurate forecasting of EV charging loads, we leveraged a Long Short-Term Memory model, which is a type of RNN as introduced in the section 2. This model was proven suitable for time series forecasting in many different scenarios and industries due to its ability to retain long-term dependencies.

4.2.1 Data Collection and Preprocessing

Our data encompassed average hourly power consumption across all the registered EV chargers within the area of Burlington for the entire year of 2023. This dataset provided a long and continuous sequence with 8,760 data points, representing the hourly power consumption over 365 days. Given the goal of hourly load forecasting, we employed a rolling window approach to construct the training data. This method of segmentation involved taking sequences of 24-hour windows as inputs, with each window predicting the following 25th hour's power consumption. The transformation produced a final training set with dimensions of 8736 × 24, allowing the LSTM to learn temporal patterns based on the daily trends.

4.2.2 Model Design and Validation

The LSTM model was constructed with two stacked LSTM layers, each comprising 50 units. These layers were sequentially organized to allow the network to progressively learn more complex temporal dependencies. Following each LSTM layer, we implemented dropout layers with a dropout rate of 0.2 to avoid the problem of overfitting. The dropout mechanism helped to mitigate overfitting by randomly deactivating a fraction of the LSTM neurons during training, thus promoting a more generalized learning. The final architecture concluded with two fully connected feed forward (dense) layers. These layers served to adjust the output shape from the LSTM layers, providing the final prediction in the required format: a single value representing the 25th hour's power consumption. This structure enabled efficient mapping from the input sequence to the predicted outcome. Overall, the combination of LSTM layers, dropout, and fully connected layers produced a model architecture tailored to capture both short-term fluctuations and longer-term dependencies in EV consumption patterns.

The model training process involved a total of 20 epochs as shown in the **Fig 4.4**. With each epoch, both training and validation losses showed consistent decreasing trend, indicating effective learning and reduction of prediction error, further showing our architecture is free from the problem of overfitting.

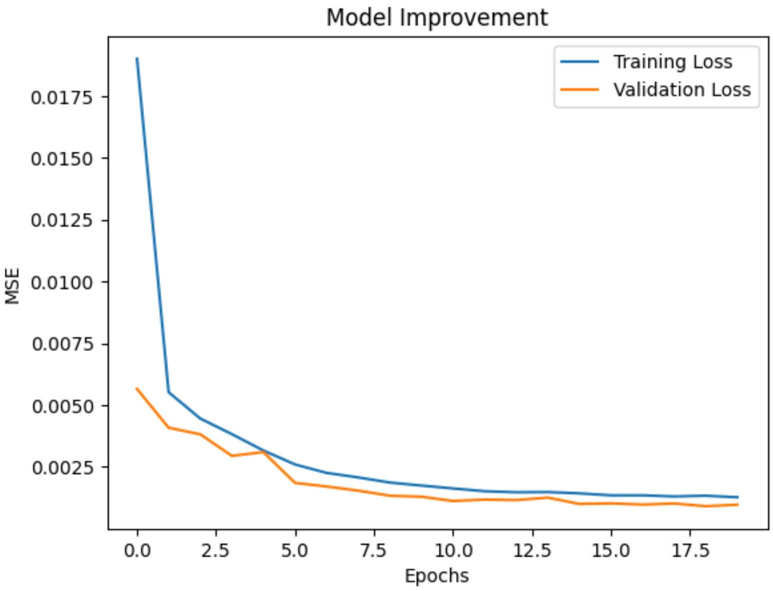


Fig 4.4 Model Improvement on Training and Validation Sets

The following **Fig 4.5** shows the training process based on all the available data points without partitioning into separate training and validation sets. The model prioritized generalization from the entire power consumption dataset, maximizing the utility of historical information for the future forecasts.

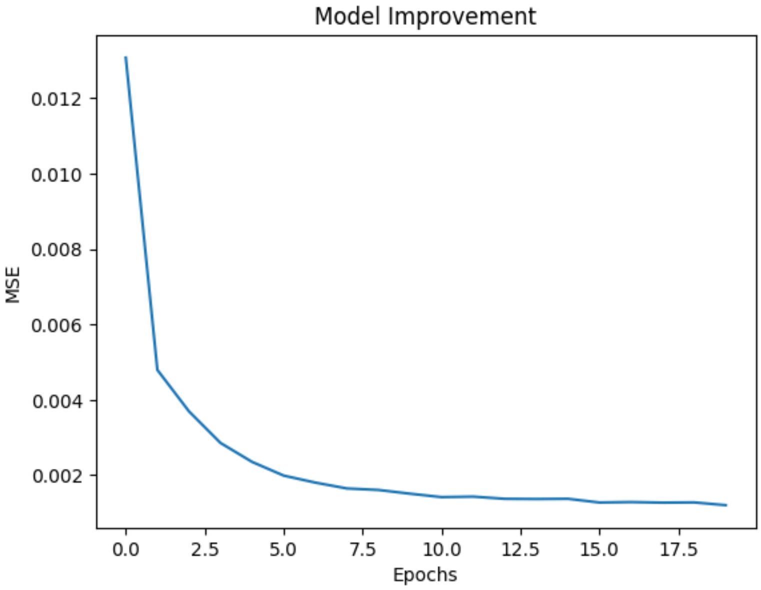


Fig 4.5 Model Improvement on All Available Information

In terms of computational efficiency, each epoch's training time was approximately one minute, a manageable demand on resources that allows for rapid iteration and tuning. This efficiency is significant for real-world deployment, where forecasting models may need frequent updates as new data is acquired. Additionally, the model's simplicity relative to more complex architectures (such as Transformer-based models) provides an optimal balance between performance and efficiency, making it a viable candidate for real-time applications in load forecasting.

4.2.3 Forecasting Results

The LSTM model demonstrated reliable performance in forecasting hourly EV charging loads, effectively capturing the temporal dependencies of EV charging behavior as shown in the **Fig 4.6**.

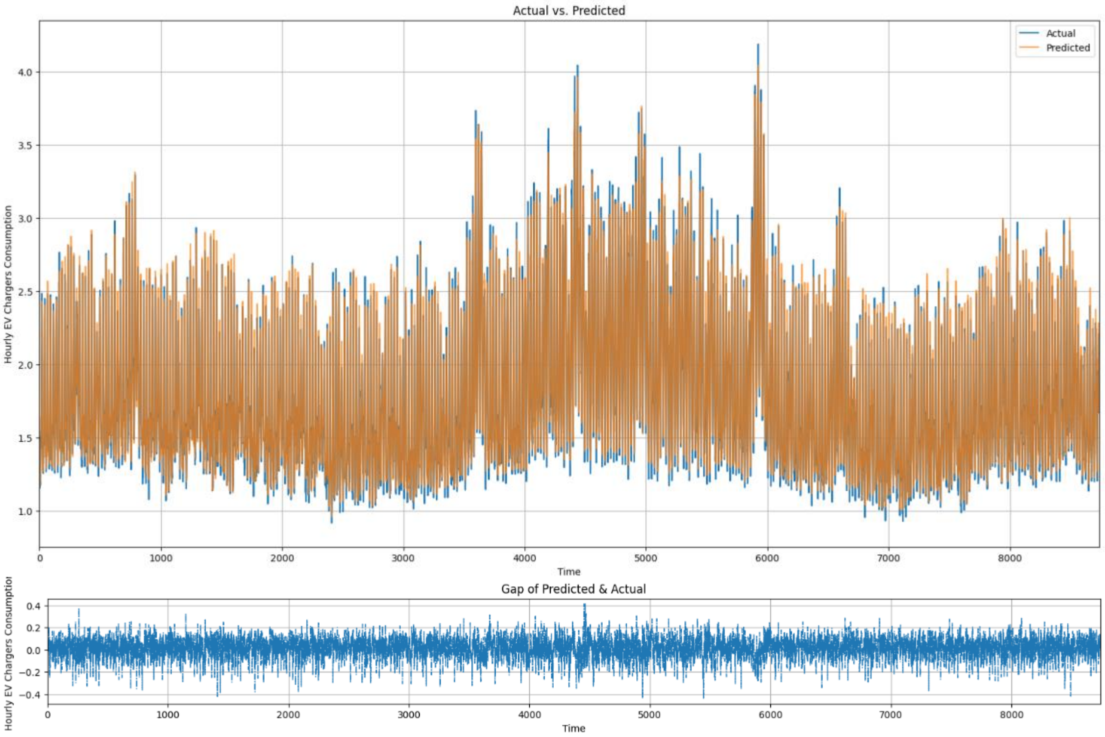


Fig 4.6 Result of Deep Learning Model

The orange line of predicted values is close to the blue line of actual values of power consumption. The gap in between is also controlled within an acceptable range compared to the actual values. Based on the data for the year of 2023, we can also forecast the hourly power consumption for the year of 2024 in the same rolling pattern approach.

To conclude, the deep learning method shows a powerful forecasting capacity with the mean square error controlled at the level of 10^{-6} compared to the mean square error of around 0.25 obtained from the time series analysis model.

5 Conclusion

5.1 Key Findings

Our research and analysis provided several important insights for both local distribution system and decision makers in the industry of energy and utility. As the proliferation of EVs introduces novel demand patterns to the grid, forecasting models such as ours become instrumental in ensuring operational resilience and supply robustness. Through accurate forecasting, utility providers can proactively mitigate overload risks, thereby reducing the likelihood of service interruptions, and enhancing their customers' experience and satisfaction. The clustering approach, particularly the use of K-Means clustering, was instrumental in differentiating between charging and non-charging days. By two-step clustering the consumption data according to unique daily power consumption profiles, we effectively identified patterns in user charging behaviors that could inform demand-side management strategies.

Additionally, the use of wavelet decomposition and exponential moving average (EMA) provided valuable preprocessing methods, filtering noise and isolating significant consumption spikes that indicate EV charging events. These methods ensured that our model received refined inputs, enhancing its ability to make accurate predictions. Our two-step clustering process further validated the categorization of charging days with higher accuracy with the results from wavelet decomposition and clustering outputs, which can be used to make granular assessments of EV-related demand at different times of the day.

Overall, this research confirms the utility of combining clustering, time-series analysis, and deep learning methods for load forecasting in distribution systems impacted by EV usage. This approach provides a scalable framework for power distribution companies to manage the EV charging loads effectively, helping prevent overloading issues in specific transformers and supporting sustainable infrastructure planning as EV adoption continues to grow.

5.2 "By-product" Insights

During the clustering analysis, we observed several valuable "by-product" insights that could be instrumental in refining future demand-side management strategies. First, by segmenting power consumption profiles into off-peak, mid-peak, and on-peak hours, we identified specific consumer behavior patterns that align with price sensitivity. Some clusters revealed distinct consumption peaks during off-peak hours, suggesting that certain EV users are likely influenced by time-of-use pricing structures. Recognizing these patterns allows utility providers to create targeted incentives that encourage off-peak usage, balancing loads across the grid more effectively.

The clustering results also highlighted behavioral distinctions across user types. We identified groups of "Cost Sensitive" users who charge predominantly during off-peak hours, "Insensitive" users with balanced consumption throughout the day, "Thrifty" users with consistently low consumption, and "Consuming" users with consistently high usage. These profiles can inform personalized energy plans, guiding utilities in designing incentives or recommending charging schedules that fit different user preferences.

Moreover, our clustering experiments with variable numbers of clusters provided insight into how seasonal shifts affect EV charging patterns. For example, consumption patterns varied significantly between summer and winter months, likely due to differences in travel behavior and energy needs. By adapting clustering parameters seasonally, our model could offer even more precise consumption profiles, helping utilities to adjust their forecasting and load management strategies according to seasonal variations in EV demand.

5.3 Expectations & Reflections

Our research has the potential to extend beyond forecasting EV loads in residential settings, with applications in detecting charging events and forecasting charging loads for public EV charging stations and other variable-demand energy sources like renewable energy installations. As our approach combines time-series forecasting and clustering with wavelet decomposition, it provides a robust framework for handling complex, variable demand patterns across a range of applications. We expect that further refinement, particularly through real-time data integration and semi-supervised learning models, could enhance the forecasting accuracy and responsiveness of this model. Moreover, we can customize and personalize the forecasting process for a more micro condition with more computing power and time input, to produce even more accurate and sensitive managerial insights for local distribution systems.

Reflecting on the challenges faced, one of the primary limitations of this study was data availability. While the labeled data for EV charging events was limited, semi-supervised learning could be a solution to leverage the large amount of unlabeled data available. This approach could significantly improve model performance by iteratively refining predictions with new high-confidence data. One another improvement is the use of more complex deep learning architectures, such as Transformer models or attention mechanisms. These models may enhance long-term forecasting capabilities and enable multi-step forecasting, such as daily or weekly consumption predictions. However, we still need to validate the models and decide the trade-off between model's ability and training input.

Overall, this project underscores the importance of adaptive forecasting tools in managing EV charging loads and ensuring grid resilience at the transformer level. By employing a multi-faceted approach to forecasting that integrates clustering, wavelet decomposition, and deep learning, this study contributes a practical, replicable framework for addressing the evolving challenges posed by EVs on distribution systems.

Reference

[1] Biswajit Biswal, S. Deb, S. Datta, Taha Selim Ustun, and U. Cali, “Review on smart grid load forecasting for smart energy management using machine learning and deep learning techniques,” *Energy Reports*, vol. 12, pp. 3654–3670, Sep. 2024, doi: https://doi.org/10.1016/j.egyr.2024.09.056.

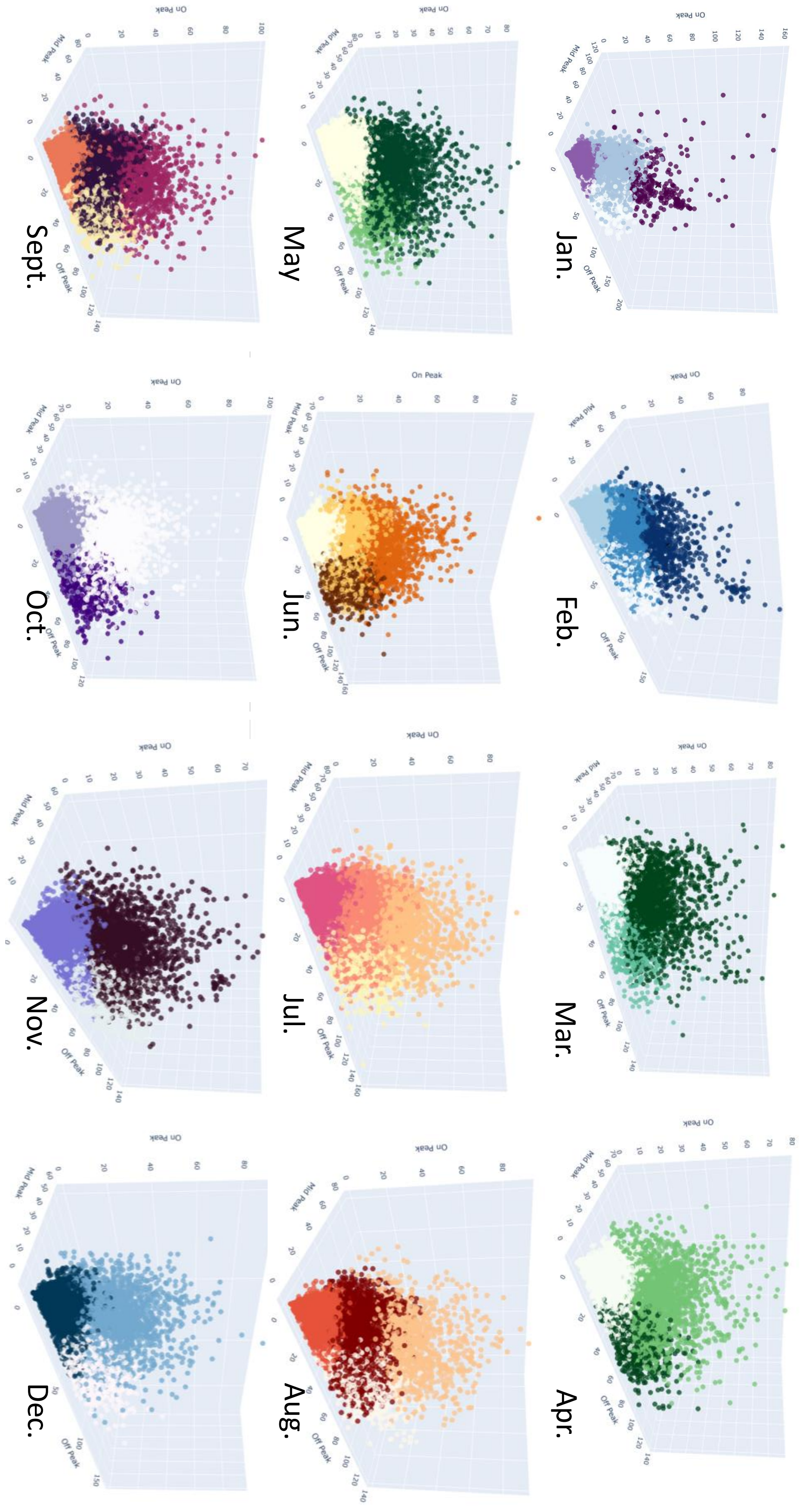
[2] F. Ren, C. Tian, G. Zhang, C. Li, and Y. Zhai, “A hybrid method for power demand prediction of electric vehicles based on SARIMA and deep learning with integration of periodic features,” *Energy*, vol. 250, p. 123738, Jul. 2022, doi: https://doi.org/10.1016/j.energy.2022.123738.

[3] M. H. Amini, A. Kargarian, and O. Karabasoglu, “ARIMA-based decoupled time series forecasting of electric vehicle charging demand for stochastic power system operation,” *Electric Power Systems Research*, vol. 140, pp. 378–390, Nov. 2016, doi: https://doi.org/10.1016/j.epsr.2016.06.003.

[4] J. Zhu, Z. Yang, Y. Guo, J. Zhang, and H. Yang, “Short-Term Load Forecasting for Electric Vehicle Charging Stations Based on Deep Learning Approaches,” *Applied Sciences*, vol. 9, no. 9, p. 1723, Apr. 2019, doi: https://doi.org/10.3390/app9091723.

[5] Z. Zhuang, X. Zheng, Z. Chen, T. Jin, and Z. Li, “Load forecast of electric vehicle charging station considering multi-source information and user decision modification,” MDPI, https://www.mdpi.com/1996-1073/15/19/7021 (accessed Nov. 7, 2024).

Appendix 1 – Monthly clustering results for year of 2023


Jan.
Feb.
Mar.
Apr.
May
Jun.
Jul.
Aug.
Sept.
Oct.
Nov.
Dec.
On Peak
Mid Peak
Off Peak