Analyzing Power Demand Variations in Electric Vehicle Charging Stations Across Cities: Identifying Key Regional Influences

Avinash Kola – akola@unh.newhaven.edu

Prem Niranjan Undavali – piunda1@unh.newhaven.edu

Hemanth Reddy Reddy Battula – hredd2@unh.newhaven.edu

TEAM UNSTOPPABLE

Data Mining

shivanjali khare

ABSTRACT

The purpose of this research is to determine the regional factors that affect the changes in power demand at electric vehicle (EV) charging stations in various cities. The study intends to find trends and insights that can guide the strategic planning and optimization of EV charging infrastructure by examining data from several areas. locations, Power consumed, number of stalls, and the extent of EV adoption are important regional considerations considered. The results will give city planners, and operators of EV charging stations important information to improve the effectiveness and dependability of EV charging networks and make sure they satisfy the various demands of that city populations.

INTRODUCTION

The transportation systems around the world are changing rapidly because of the electric cars' quick adoption. Infrastructure for dependable and effective EV charging has become more and more important as the number of EVs keeps rising. City planners and infrastructure suppliers have difficulties since EV charging station power demands might range greatly between cities. Optimizing EV charging station placement and management requires an understanding of these variances and the underlying regional characteristics.

We have many benefits by moving to EV's which includes reducing greenhouse gas emissions, reduced usage of fossil fuels. However, the increased usage of EVs puts additional strain on the electrical system. To meet the demands of EV owners, especially during peak charging hours, EV charging stations require a steady and increased power supply. Many factors, including location, power output from charging stations, number of stalls, and elevation in city, can be utilized to explain the variation in power consumption between cities.

RESEARCH QUESTION

This study aims to answer the following research question: How does the power demand of electric vehicle (EV) charging stations vary across different cities and what regional factors influence these variations?

RELATED WORK

Chen, Y., & Zhang, Y. (2018). "Electric vehicle charging behavior analysis and prediction based on big data." IEEE Transactions on Smart Grid, 9(5), 4822-4831.

In this study they used big data analytics to examine how EVs are charged. Considering variables like the time of day, day of the week, and weather, the authors create a prediction model to estimate charging demand. Their research emphasizes how crucial data-driven methods are for comprehending and forecasting EV charging trends.

Kang, C., & Kim, J. (2019). "Impact of urban form on electric vehicle charging demand: A case study of Seoul, South Korea." ***Journal of Cleaner Production*****, 235, 1175-1184.**

This research focuses on the relationship between Seoul's EV charging demand and urban form, including land use and population density. The authors proved important metropolitan features that affect charging behavior by combining survey and geographic information system (GIS) data. According to their research, the demand for EV charging is significantly influenced by urban density and mixed-use development.

Liu, Y., & Wang, J. (2020). "A spatiotemporal analysis of electric vehicle charging demand in urban areas." ***Energy Policy*****, 144, 111663.**

The spatial distribution and temporal patterns of charging sessions are the main topics of this research spatiotemporal analysis of EV charging demand in urban settings. By knowing the hotspots and charging peak times using data from several charging stations, the researchers got to know the temporal and spatial dynamics of EV charging behavior.

A Deep Learning Approach for Prediction of Electrical Vehicle Charging Stations Power Demand in Regulated Electricity Markets: The Case of Morocco

This research focuses on the future power consumption for EV charging stations in Rabat, Morocco, using LSTM networks and a deep learning technique. Including station ID, charging start and end times, location. The data collection comprises 2000 records from two EV charging stations. The LSTM model achieves low error rates and an MAE below 10%, according to the study's evaluation of model performance using Mean Absolute Error (MAE) and Root Mean Square Error (RMSE).

Electric Vehicle Charging Station Demand Prediction Model

To determine which model is most effective at forecasting EV charging demand, this research evaluates a number of models. Data on charging sessions and charge IDs were gathered from 14 EV charging stations run by Perth & Kinross Council between September and December of 2017 and 2019. In the study to compare the LSTM, random forest, decision tree, and linear regression models. With an R-squared value of 0.87 and an RMSE of 5.6, the random forest model performed the best, demonstrating remarkable accuracy.

PROPOSED METHOD

The method that is proposed determines the regional factors influencing these discrepancies by analyzing and forecasting the power demand of electric vehicles (EV) charging stations across various cities. To maximize model performance, the study makes use of a large dataset and best data mining approaches.

Data Collection and Preprocessing

Data Source: The dataset used in this study is taken from Kaggle https://www.kaggle.com/datasets/omarsobhy14/supercharge-locations. It includes details such as street, city, zip code, country, GPS coordinates, kilowatts (KW), and elevation. The dataset consists of more than 5000 sessions from 100 drivers across 25 EV stations, which has various data types including strings, timestamps, categorical variables, and integers.

Data Preprocessing:

Cleaning: Handled missing values and removed duplicates.

Transformation: Converted categorical variables into numerical format using techniques like one-hot encoding.

Normalization: Scaling numerical features to ensure uniformity in the data using standard scalar.

Visualization of Data:

	Supercharger	Street Address	City	State	Zip	Country	Stalls	kW	GPS	Elev(m)	Open Date
0	Buellton, CA	555 McMurray Rd	Buellton	CA	93427	USA	10	150.0	34.61456, -120.188387	114	7/13/2013
1	Corning, CA	950 Hwy 99	Corning	CA	96021	USA	6	150.0	39.926454, -122.198393	87	10/18/2013
2	Barstow, CA	2812 Lenwood Rd	Barstow	CA	92311	USA	16	150.0	34.849129, -117.085446	725	11/19/2012
3	Tifton, GA	1310 U.S. 82	Tifton	GA	31794	USA	8	150.0	31.448847, -83.53221	92	7/10/2014
4	Roseville, CA	1151 Galleria Blvd	Roseville	CA	95678	USA	7	150.0	38.771208, -121.266149	66	4/29/2014

Fig 1: Top 5 records of our data.

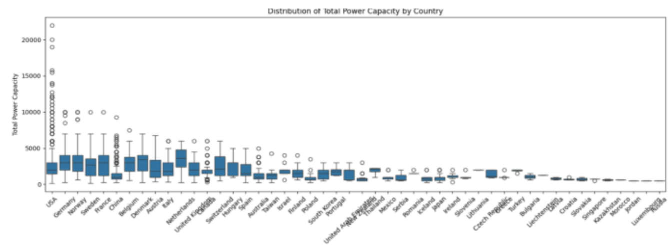


Fig 2: Box plot to identify outliers and understand the data.

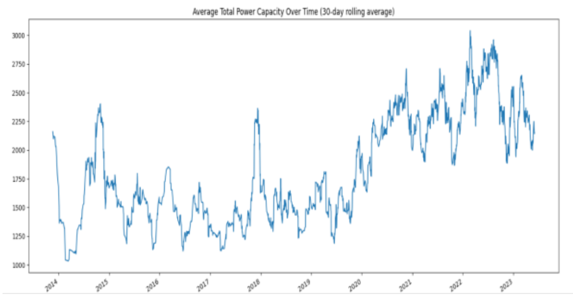


Fig 3: Showing an increase in use of EV over the years.

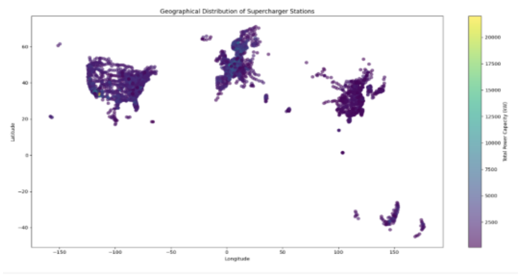


Fig 4: Geographical distribution of charging stations.

Data cleaning visualization

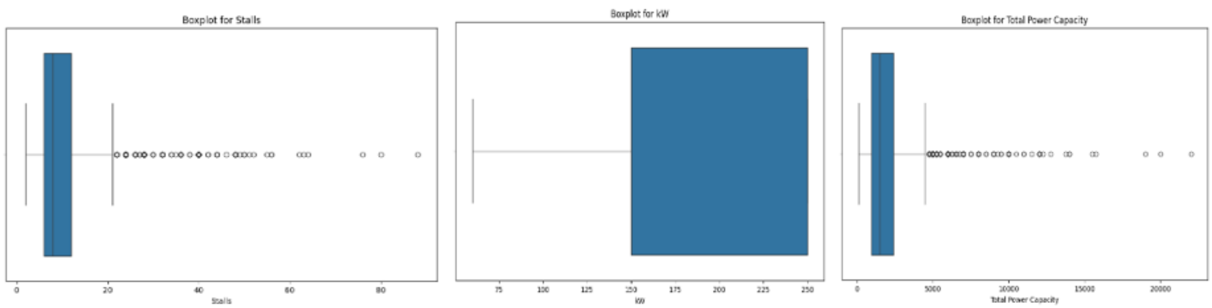


Fig 5: Box plot showing the presence of outliers in the data

```python
from sklearn.preprocessing import StandardScaler
from sklearn.preprocessing import StandardScaler, OneHotEncoder

# Select relevant features
features = ['Stalls', 'kW', 'Elev(m)', 'Latitude', 'Longitude', 'Total Power Capacity','Country']
model_data = cleaned_data[features].dropna()

# Calculate IQR for outlier detection
Q1 = model_data['Total Power Capacity'].quantile(0.25)
Q3 = model_data['Total Power Capacity'].quantile(0.75)
IQR = Q3 - Q1


# Define outlier boundaries
lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR

# Remove outliers
filtered_data = model_data[(model_data['Total Power Capacity'] >= lower_bound) &
                           (model_data['Total Power Capacity'] <= upper_bound)]

# Split features and target
X = filtered_data.drop('Total Power Capacity', axis=1)
y = filtered_data['Total Power Capacity']

X.reset_index(drop=True, inplace=True)
y.reset_index(drop=True, inplace=True)

encoder = OneHotEncoder(sparse_output=False, handle_unknown='ignore') # sparse=False for compatibility with StandardScaler

# Fit and transform the encoder on the 'Country' column
encoded_country = encoder.fit_transform(X[['Country']])

# Create a DataFrame from the encoded features
encoded_country_df = pd.DataFrame(encoded_country, columns=encoder.get_feature_names_out(['Country']))

# Drop the original 'Country' column and concatenate the encoded features
X = X.drop('Country', axis=1)
X = pd.concat([X, encoded_country_df], axis=1)

# Scale features
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
```

Fig 6: Code to remove outliers using the quartile method.

```python
plt.figure(figsize=(10, 6))
sns.histplot(model_data['Stalls'].dropna(), bins=20, kde=True)
plt.title('Distribution of Stalls')
plt.xlabel('Number of Stalls')
plt.ylabel('Frequency')
plt.show()


plt.figure(figsize=(12, 8))
plt.scatter(model_data['Longitude'], model_data['Latitude'], c=model_data['Total Power Capacity'], cmap='viridis', alpha=0.6)
plt.colorbar(label='Total Power Capacity (kW)')
plt.title('Geographical Distribution of Supercharger Stations')
plt.xlabel('Longitude')
plt.ylabel('Latitude')
plt.show()
```

Fig 7: Code to plot the distribution of stalls.

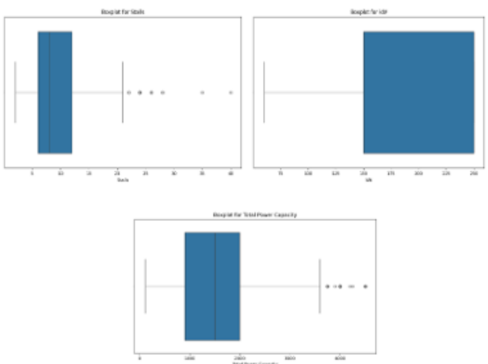


Fig 8: Box plots showing reduced outliers.

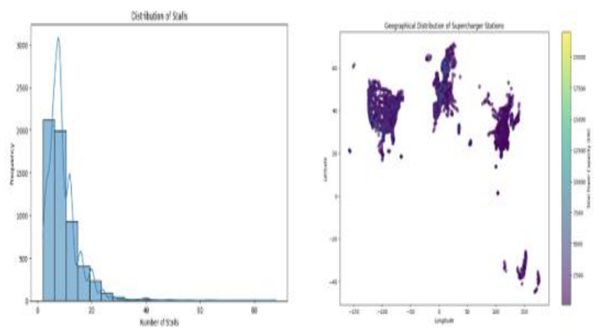


Fig 9: Distribution of stalls after removing outliers.

Data Mining Techniques

In this research we used three primary data mining techniques to predict power demand:

1. **Random Forest**:
   - **Parameters and Hyperparameters**:
     - n_estimators: Number of trees in the forest.
     - max_depth: Maximum depth of the tree.
     - min_samples_split: Minimum number of samples in an internal node.
     - min_samples_leaf: Minimum number of samples at a leaf node.

2. **Gradient Boosting Machines (GBM)**:
    - **Parameters and Hyperparameters**:
        - n_estimators: Number of boosting stages.
        - learning_rate: Learning rate of each tree.
        - max_depth: Maximum depth of the individual regression estimators.
        - min_samples_split: Minimum number of samples in an internal node.
3. **Support Vector Machines (SVM)**:
    - **Parameters and Hyperparameters**:
        - C: It is a Regularization parameter that can control the trade-off between getting a low training error and a low testing error.
        - gamma: Kernel coefficient for rbf, poly, and sigmoid.
        - kernel: It specifies the kernel type to be used in the algorithm.

Optimization Techniques

To improve the performance of the data mining techniques, we used the following technique:

Grid Search Cross Validation (GridSearchCV):

- **Purpose**: It is used to find the optimal hyperparameters for each model by evaluating all possible combinations from given parameters.
- **Implementation**: We used it to test specific parameter values and to evaluate the performance for each combination using cross-validation.

Hyperparameter Tuning

1. **Random Forest**:
    - **Parameter Grid**:
        - n_estimators: [100, 200, 300]
        - max_depth: [10, 20, 30, None]
        - min_samples_split: [2, 5, 10]
        - min_samples_leaf: [1, 2, 4]

2. **Gradient Boosting**:
    - **Parameter Grid**:
        - n_estimators: [100, 200, 300]
        - learning_rate: [0.01, 0.1, 0.3]
        - max_depth: [3, 4, 5]
        - min_samples_split: [2, 4]
3. **Support Vector Machine (SVM)**:
    - **Parameter Grid**:
        - C: [0.1, 1, 10]
        - gamma: ['scale', 'auto', 0.1, 0.01]
        - kernel: ['rbf', 'linear']

```
# Import additional libraries for optimization
from sklearn.model_selection import GridSearchCV, RandomizedSearchCV
from sklearn.metrics import mean_squared_error, r2_score, mean_absolute_error
import numpy as np

# 1. Default Random Forest
rf_default = RandomForestRegressor(max_depth=2,random_state=12)
rf_default.fit(X_train, y_train)
rf_default_pred = rf_default.predict(X_test)

# 2. Default Gradient Boosting
gb_default = GradientBoostingRegressor(max_depth=1,random_state=12)
gb_default.fit(X_train, y_train)
gb_default_pred = gb_default.predict(X_test)

# 3. Default SVM
svm_default = SVR()
svm_default.fit(X_train, y_train)
svm_default_pred = svm_default.predict(X_test)

# Function to calculate metrics
def calculate_metrics(y_true, y_pred):
  return {
      'R2': r2_score(y_true, y_pred),
      'MSE': mean_squared_error(y_true, y_pred),
      'RMSE': np.sqrt(mean_squared_error(y_true, y_pred)),
      'MAE': mean_absolute_error(y_true, y_pred)
  }

# Calculate default metrics
default_metrics = {
  'Random Forest': calculate_metrics(y_test, rf_default_pred),
  'Gradient Boosting': calculate_metrics(y_test, gb_default_pred),
  'SVM': calculate_metrics(y_test, svm_default_pred)
}

# Display default metrics
default_df = pd.DataFrame(default_metrics).round(4)
print("\nDefault Model Metrics:")
print(default_df)
```

Fig 10: Code for techniques with default parameters.

```
# 1. Random Forest Optimization
rf_param_grid = {
  'n_estimators': [100, 200, 300],
  'max_depth': [10, 20, 30, None],
  'min_samples_split': [2, 5, 10],
  'min_samples_leaf': [1, 2, 4]
}

rf_grid = GridSearchCV(RandomForestRegressor(random_state=42),
                       rf_param_grid, cv=5, scoring='r2')
rf_grid.fit(X_train, y_train)
rf_optimized = rf_grid.best_estimator_
rf_optimized_pred = rf_optimized.predict(X_test)
```

Fig 11: Optimizing random forest using GridSearchCV

```
# 2. Gradient Boosting Optimization
gb_param_grid = {
  'n_estimators': [100, 200, 300],
  'learning_rate': [0.01, 0.1, 0.3],
  'max_depth': [3, 4, 5],
  'min_samples_split': [2, 4]
}

gb_grid = GridSearchCV(GradientBoostingRegressor(random_state=42),
                       gb_param_grid, cv=5, scoring='r2')
gb_grid.fit(X_train, y_train)
gb_optimized = gb_grid.best_estimator_
gb_optimized_pred = gb_optimized.predict(X_test)
```

Fig 12: Optimizing Gradient Boosting

```
# 3. SVM Optimization
svm_param_grid = {
  'C': [0.1, 1, 10],
  'gamma': ['scale', 'auto', 0.1, 0.01],
  'kernel': ['rbf', 'linear']
}

svm_grid = GridSearchCV(SVR(), svm_param_grid, cv=5, scoring='r2')
svm_grid.fit(X_train, y_train)
svm_optimized = svm_grid.best_estimator_
svm_optimized_pred = svm_optimized.predict(X_test)
```

Fig 13: Optimizing SVM using GridSearchCV

THE EXPERMINTAL RESULTS

By using the best data mining and optimization approaches, we were able to significantly improve its ability to get the accurate power demand of electric vehicle (EV) charging stations across various cities.

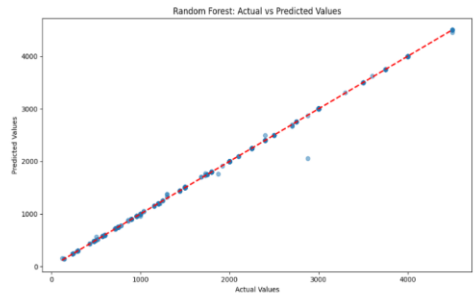
1. **Model Performance**:
    - **Random Forest**:
        - **R^2 Score**: Improved from 0.811 to 0.99.
        - **Mean Squared Error (MSE)**: Reduced from 682 to 657, showing a decrease in prediction error.
        - **Root Mean Squared Error (RMSE)**: Decreased, indicating better accuracy in predictions.
        - **Mean Absolute Error (MAE)**: Reduced, further confirming improved prediction accuracy.
    - **Gradient Boosting**:
        - **R^2 Score**: Improved from 0.95 to 0.99.
        - **Mean Squared Error (MSE)**: Reduced from 349 to 156, indicating significant error reduction.
        - **Root Mean Squared Error (RMSE)**: Decreased, showing improved accuracy.
        - **Mean Absolute Error (MAE)**: Reduced, confirming better prediction performance.
    - **Support Vector Machine (SVM)**:
        - **R^2 Score**: Improved from 0.15 to 0.90, showing a substantial increase in predictive power.

- **Mean Squared Error (MSE)**: Reduced significantly, indicating better accuracy.
- **Root Mean Squared Error (RMSE)**: Decreased, showing improved accuracy.

Default Model Metrics:

	Random Forest	Gradient Boosting	SVM
R2	0.8111	0.9584	0.1520
MSE	182442.6392	40155.4643	818896.3579
RMSE	427.1330	200.3883	904.9289
MAE	376.5513	127.6184	689.9524

Fig 14: Results with default parameters




```
Best Parameters:
{'max_depth': 20, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 300}

Best Score: 0.9995
```

Fig 15: Best parameters after performing optimization.

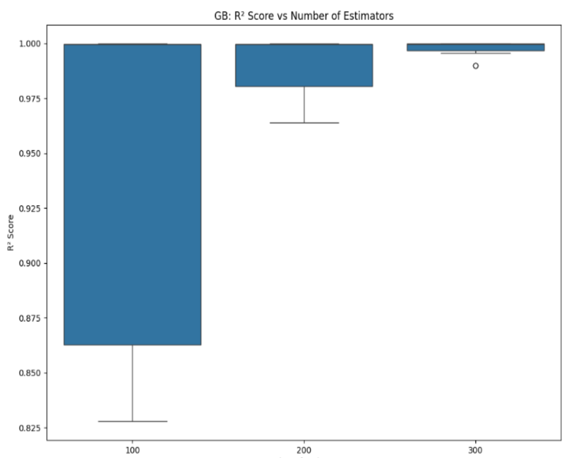


Fig 16: Best parameters for Gradient Boosting

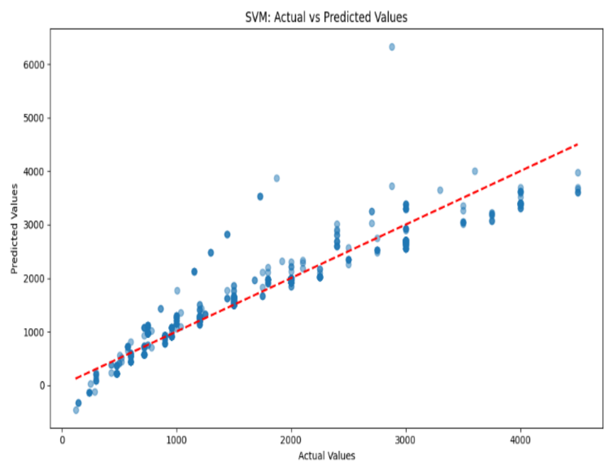


Fig 17: Best parameters for SVM after optimization

Optimized Model Metrics:

	Random Forest	Gradient Boosting	SVM
R2	0.9993	0.9998	0.9027
MSE	657.8305	156.7533	93993.9994
RMSE	25.6482	12.5201	306.5844
MAE	1.6297	2.3004	189.6670

Fig 18: Results after hyper parameter tuning

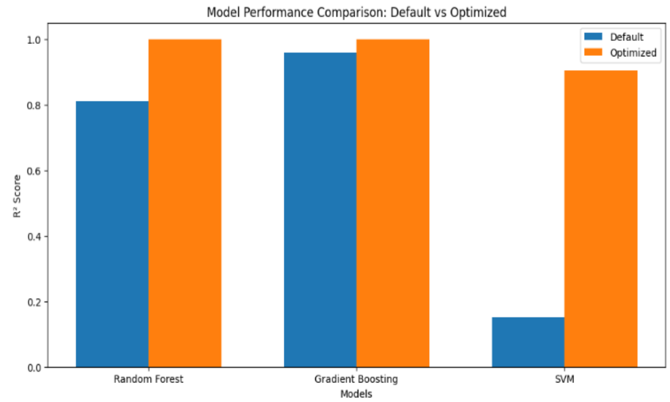


Fig 19: Performance comparison b/w default and optimized

DISSCUSSION

The results of this research have shown us some valuable insights into the power demand variations at electric vehicle (EV) charging stations across different cities and the regional factors that influence these variations.

Strengths of the study:

In this research we have used a large and diverse dataset, which included more than 5000 sessions from 100 drivers across 25 EV stations. This dataset provided a foundation for analysis and model training.

We have used machine learning techniques such as Random Forest, Gradient Boosting Machines, and Support Vector Machines, along with optimization techniques, and models were well-tuned and capable of providing accurate predictions.

The analysis of regional factors provided great insights into the factors that are influencing power demand variations, which can be used for city’s strategic planning and infrastructure development.

Limitations of the Research:

The dataset was from a specific EV Provider and may not be same for all EV's. Future studies could benefit from more real time data and from different EV networks representative dataset.

The study considered many regional factors but did not consider some potential external factors, such as economic conditions, policy changes, and technological advancements. Future research could add these variables to improve model accuracy.

The dataset does not include seasonal data and weather conditions, which can significantly affect EV charging behavior. Future studies can add these variables to better understand and predict power demand variations.

CONCLUSION AND FUTURE WORK

Conclusion:

The proposed method has predicted the power demand of EV charging stations using advanced data mining techniques and optimization methods. The Random Forest, Gradient Boosting, and SVM models showed significant improvements in performance metrics, with the Random Forest model getting the highest R^2 score of 0.99. These results show the effectiveness of the approach in addressing research questions and providing valuable insights for optimizing EV charging infrastructure.

Future Work:

So, we can improve the data set which can include real-time data that is generated directly from the EV providers and can update on day to basis. So we can improve our model on the real-time data to improve the model predictions.

We can include additional factors into the dataset such as economic conditions, policy changes, and technological advancements that could improve the accuracy and robustness of the models.

We can conduct a more detailed analysis of user behavior, such as charging patterns and preferences, could provide additional insights into optimizing charging infrastructure.

We can use seasonal data and weather conditions into the models that could improve the accuracy of power demand predictions.

REFERENCES

-https://ieeexplore.ieee.org/document/8310451

-https://www.sciencedirect.com/science/article/pii/S095965261932003X

-https://www.sciencedirect.com/science/article/pii/S030142152030200X

-https://www.sciencedirect.com/science/article/pii/S096014812102837X

https://www.researchgate.net/publication/344234567_A_Deep_Learning_Approach_for_Prediction_of_Electrical_Vehicle_Charging_Stations_Power_Demand_in_Regulated_Electricity_Markets_The_Case_of_Morocco

-https://www.elsevier.com/books/data-mining/han/978-0-12-381479-1

Link to github repo: Link to our git hub repository

Link to Google Co lab: Link of Google Co lab

we have connected with the writing center for proofreading our work:

Organizer Moushume Chowdhury	**Location** CLR - Remote/Zoom
Topic CLR - Writing Center	**Date of Appointment** 12/09/2024
Comments No comment provided.	**Time** 3:00pm - 4:00pm ET
Cancel/Reschedule Appointment: Cancel Appointment	