Institut für Automatisierungstechnik

Master Thesis

A data-driven approach to predict the load profile at an electric vehicle charging station

Hrushikesh Mantri

December 3, 2024

Project Number: MT2425

Tutor:
Babak Ravanbach (M.Sc., M.B.A.)
Msc. Markus Miller

Examiner:
Prof. Dr.-Ing. Johanna Myrzik
Dr. Amin Mallek

Universität Bremen

Master Arbeit/Thesis

Institut für Automatisierungstechnik

Anwendung datengesteuerter Ansätze zur Vorhersage des Lastprofils an einer Ladestation für Elektrofahrzeuge.

Hintergrund Diese Masterarbeit untersucht die Vorhersage des Energieverbrauchs an Ladestationen durch datengestützte Methoden. Diese Forschung zielt darauf ab die Nutzung von Ladestationen und georäumlichen Daten, um den Energiebedarf vorherzusagen. Die Studie umfasst die Sammlung und Vorverarbeitung von Daten über das Ladeverhalten von Elektrofahrzeugen und andere Einflussfaktoren, gefolgt von Feature-Engineering und Datenanalyse, um die wichtigsten Determinanten der lastprofile Profile.

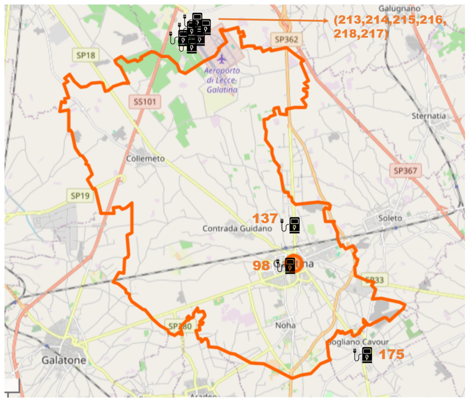

Abbildung 1: Charging stations locations

Aufgaben Entwurf und Implementierung eines Modells der künstlichen Intelligenz zur Vorhersage des Lastprofils an einer Ladestation für Elektrofahrzeuge, basierend auf den folgenden Anforderungen:

- Berücksichtigung sowohl zeitlicher als auch räumlicher Abhängigkeiten..
- Robustheit gegenüber fehlenden oder verrauschten Daten.
- Vergleich der Ergebnisse mit den Echtzeitdaten der Ladestationen.

A data-driven approach to predict load profile at a electric vehicle charging station

Background This master's thesis explores prediction of load profile at charging stations,through data-driven methodologies. This research aims to use charging station usage and geo-spatial data, to predict load profile. The study involves collecting and preprocessing data on EV charging patterns and other influencing factors, followed by feature engineering and data analysis to identify key determinants of load profiles.

Tasks Design and implementation of an artificial intelligence model to predict the load profile at an electric vehicle charging station based on the following requirements:

- Handle both Temporal and Spatial Dependencies.
- Robustness to Missing or Noisy Data.
- Compare the results with the real time data of charging stations.

Babak Ravanbach, MT2425, Babak.Ravanbach@dlr.de

Universität Bremen

Declaration of honour

I hereby confirm on my honor that I personally prepared the present academic work and carried out myself the activities directly involved with it. I also confirm that I have used no resources other than those declared. All formulations and concepts adopted literally or in their essential content from printed, unprinted or Internet sources have been cited according to the rules for academic work and identified by means of footnotes or other precise indications of source.

The support provided during the work, including significant assistance from my supervisor has been indicated in full.

The academic work has not been submitted to any other examination authority. The work is submitted in printed and electronic form. I confirm that the content of the digital version is completely identical to that of the printed version.

Date: 03/12/2024 Signature: ______________________

Declaration of publication

- ☒ I hereby agree, that my thesis will be available for third party review in purpose of academic research.
- ☒ I hereby agree, that my thesis will be available after 30 years (§7 Abs. 2 BremArchivG) in the university archive for third party review in purpose of academic research.
- ☐ I hereby do *not* agree, that my thesis will be available for third party review in purpose of academic research.

Datum: 03/12/2024 Unterschrift: ______________________

Abstract

The rapid adoption of electric vehicles (EV's) presents growing challenges for EV charging station operators, particularly in managing operational costs and optimizing energy pricing. This master thesis address how to predict power consumption at EV charging stations using data driven approaches.

Using data from charging stations in Italy, this study evaluates predictive models, such as Long short term memory (LSTM), Graph convolution Long short term memory (GCLSTM), Simple Recurrent network (RNN), and a persistence model as a benchmark. These models leverage historical power usage and geo spatial data to predict power consumption patterns. Model performances are evaluated on the dataset, highlighting the model best suited for the predictions.

Results indicate that LSTM model, using the original dataset, achieves an average MAE of 0.3456 over six weeks of test dataset, effectively capturing temporal dependencies in the data. In comparison, the GCLSTM model achieves an MAE of 0.4593 trying to capture inter station relations. Overall LSTM best capturing if temporal dependencies are predominant and GCLSTM helps to achieve inter station relations.

Contents

List of Abbreviations

Abbreviation	Definition
EV	electric vehicle
SVM	support vector machines
RNN	Recurrent Neural Network
LSTM	Long Short-Term Memory
GCLSTM	Graph Convolutional Long Short-Term Memory
MSE	Mean Squared Error
RMSE	Root Mean Squared Error
MAE	Mean Absolute Error
IEA	International Energy Agency
STL	Seasonal tend loess decomposition
UTC	Coordinated universal time
EPIOT	Electromobility prediction interactive tool
TSCV	Time series cross validation
ARIMA	Autoregressive integrated moving average
SARIMA	Seasonal auroregressive moving average
kW	kilowatt
kWh	kilowatt hour
MAPE	mean absolute error

Chapter 1

Introduction

1.1 Introduction to Electric Vehicles and Charging Infrastructure

The global transportation sector is undergoing a significant transformation as electric vehicles (EVs) emerge as a sustainable alternative to traditional internal combustion engine vehicles. This transition, largely driven by environmental concerns and the push for cleaner energy sources, is supported by governments worldwide through subsidies, tax incentives, and strict emissions standards. As a result, EV adoption is rising rapidly, with over 10 million EVs on the road as of 2020 and expected to continue growing [27].

While the expansion of EVs contributes to lowering the transportation sector's carbon footprint, it also brings new challenges for EV charging operators, particularly around managing operating costs and energy prices. With higher demand, especially during peak charging periods, operators face increased costs for energy procurement and the risk of grid strain [1iteLund2015, which could impact their profitability.

This is where load profile prediction becomes crucial. By predicting demand patterns, operators can optimize energy purchasing strategies, avoiding costly last-minute procurement during peak periods and taking advantage of lower rates during off-peak hours. Effective prediction also enables better peak load profile management, helping operators balance supply and demand to prevent system overloads, stabilize operational costs, and offer more competitive pricing to customers. In this way, load profile prediction not only supports efficient energy distribution but also helps operators maintain cost-effective operations in a rapidly evolving energy landscape.

1.2 Goals & Research question

The goal of this thesis is to develop data-driven models capable of forecasting the short-term EV charging load . The models aim to predict the power consumption of hourly resolution data, ensuring a reliable understanding of charging patterns.

This task will be performed using the charging session data from stations operated by emotions in Italian cities, which account for approximately 133 charging stations.

This thesis will therefore address the following research questions:

- What parameters do the forecasted consumption of EV charging station networks depends on this dataset?
- Which forecasting model provides better day-ahead predictions of consumption?

1.3 Structure of the thesis

Chapter 1 talks about the Introduction of this thesis In Chapter 2, the different types of mathematical notations are explained, along with the mathematical and theoretical parts of all the methods and algorithms used in this thesis. In Chapter 3, the dataset and the different transformations and preprocessing steps that are used on the charging session data before modeling are described. you'll find information on how to use the different prediction models, as well as instructions for training and testing. Testing results and comparing the models based on how well they did on the testing set and the mistakes that were left over after the tests. In Chapter 5, we talk about the results of these analyses, along with the flaws in the prediction models. Finally, Chapter 6 wraps up this thesis and suggests a few ways to finish this job.

Chapter 2

Theory

This chapter lays the theoretical foundation for the methodologies employed in this thesis, which focuses on predicting load profiles at electric vehicle (EV) charging stations. Understanding the dynamic nature of load profiles consumption requires a detailed exploration of time series data characteristics, data preprocessing techniques, and advanced data driven models. The discussion begins with fundamental concepts of time series data and its patterns, which are essential for modeling temporal dependencies. Subsequently, baseline models such as the persistence model are introduced, followed by an in-depth examination of recurrent neural networks (RNNs) and their variants, including Long Short-Term Memory (LSTM) networks. Furthermore, the integration of spatial dependencies through Graph Neural Networks (GNNs) is explored, culminating in the Graph Convolutional Long Short-Term Memory (GCLSTM) model. These theoretical insights provide the necessary context for understanding the model selection, implementation, and evaluation processes detailed in later chapters.

2.1 Time Series data fundamentals

Time series data consists of sequential observations collected over time, where the temporal order of the data points is significant. Unlike conventional datasets, time series data exhibits distinct characteristics such as trends, seasonality, and irregular variations, which make it critical for capturing dynamic patterns in sequential processes. In the context of electric vehicle (EV) charging stations, time series data is indispensable for understanding and predicting load patterns influenced by factors like time of day, day of the week, and seasonal changes. Trends represent the long-term shifts in load, often linked to growing EV adoption, while seasonality captures recurring patterns, such as daily or weekly fluctuations in charging behavior. Irregular components, or residuals, encompass unexpected variations or anomalies in load consumption. Recognizing these elements is crucial for developing accurate predictive models that can inform resource allocation and load management strategies at EV charging stations. This understanding forms the foundation for preprocessing techniques and the design of advanced data driven models discussed in subsequent sections.

2.2 Scaling and Data Preprocessing

Scaling and preprocessing are critical steps in preparing time series data for effective analysis and modeling. Scaling involves transforming the range of input features to standardize the data, which ensures that all variables are comparable and helps improve the performance of data driven models, particularly those sensitive to the magnitude of values. Common scaling techniques, such as Min-Max Scaling or Standard Scaling, normalize the data to a specific range or distribution, enabling models to converge more efficiently during training.

In the context of this study, preprocessing also includes handling missing values, standardizing timestamps, and aggregating load consumption data to a uniform resolution. Missing data, a common issue in real-world datasets, is addressed by either imputation or removing incomplete records, depending on the extent of the missing values. Time-based operations, such as resampling and alignment to a consistent hourly interval, are applied to ensure temporal consistency. Additionally, spatial attributes like station coordinates and proximity to the city center are incorporated to capture location-based variations in demand. These preprocessing steps are essential for maintaining the integrity of the dataset and preparing it for subsequent feature engineering and model training, as discussed in later sections. By standardizing and enriching the data, this process enhances the model's ability to capture both temporal and spatial dependencies in EV charging demand.

2.3 persistence model

The persistence model, also known as a naive prediction model, is a simple yet often effective approach to time series prediction. It assumes that future values of a time series will match the most recent observed value, operating on the principle that what happened in the past is the best indicator of the immediate future. Mathematically, this model can be expressed as $\hat{y}_{t+1} = y_t$, where $\hat{y}_{t+1}$ is the predicted value for the next time step and y_t is the actual value at the current time step.

In the context of EV charging stations, the persistence model provides a straightforward approach to predict short-term load demand. By assuming that the power consumption in the next hour will be same as the current hour, it captures the immediate patterns in usage. While it does not account for longer-term trends, seasonality, or external influences, the model serves as an essential benchmark for evaluating the performance of more advanced techniques. Comparing the results of complex models to this baseline highlights their added value and ensures that their predictions offer a significant improvement over simple, intuitive methods. As such, the persistence model establishes a foundation for understanding the necessity and efficacy of employing advanced prediction models in this thesis.

2.4 Recurrent Neural Networks

Recurrent Neural Networks (RNNs) are a class of neural network architectures designed to process sequential data by leveraging their ability to retain information from previous time steps. Unlike traditional feedforward neural networks, RNNs incorporate a feedback loop within their structure, allowing the network to maintain a "memory" of prior inputs. This characteristic makes RNNs particularly well-suited for time series prediction tasks, where capturing temporal dependencies is critical.

The fundamental idea behind RNNs is the hidden state, which acts as a dynamic memory that gets updated at each time step based on the current input and the previous hidden state. Mathematically, the hidden state h_t at time t is computed as:

$$h_t = f(W_h x_t + U_h h_{t-1} + b_h)$$

where x_t is the input, h_{t-1} is the previous hidden state, W_h and U_h are weight matrices, b_h is the bias vector, and f is the activation function, often tanh. The output y_t is then derived from the hidden state using a transformation.

In the context of EV charging load prediction, RNNs are capable of modeling the temporal patterns in power consumption, such as daily peaks and troughs. However, standard RNNs often struggle with long-term dependencies due to issues like vanishing gradients during training. This limitation motivates the use of advanced variants, such as Long Short-Term Memory (LSTM) networks, which address these challenges effectively. By building upon the foundational principles of RNNs, these enhancements enable more accurate and robust prediction, as discussed in subsequent sections.

2.5 Long Short-Term Memory (LSTM) Networks

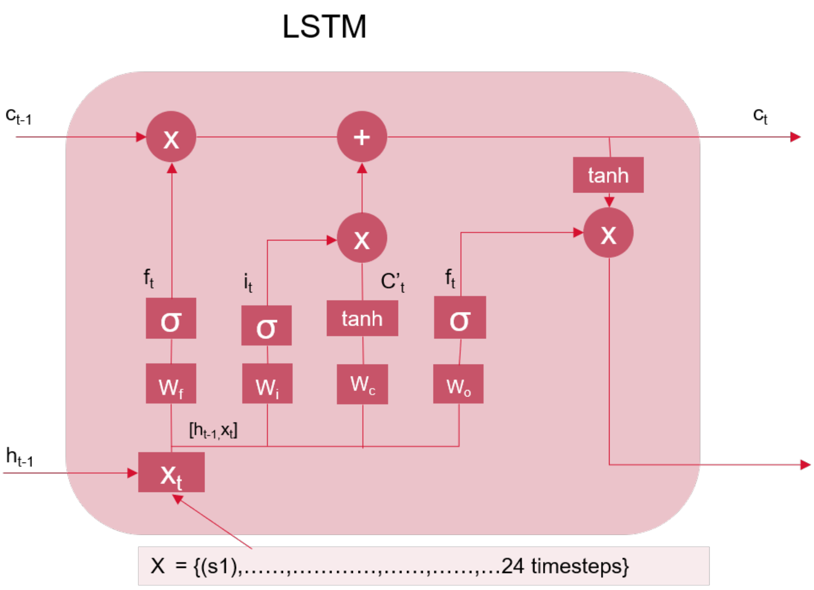

Figure 2.1: An illustration of a Long-Short Term Memory (LSTM).

Long Short-Term Memory (LSTM) networks are an advanced type of Recurrent Neural Network (RNN) specifically designed to overcome the vanishing gradient problem that limits the ability of standard RNNs to learn long-term dependencies. By introducing a memory cell and a system of gates, LSTMs can selectively retain or discard information over extended sequences, making them highly effective for time series prediction tasks, such as predicting power consumption patterns at EV charging stations. The architecture of an LSTM cell consists of three primary gates—forget, input, and output—along with a cell state that acts as the memory of the network.

The forget gate determines which parts of the previous cell state should be retained or forgotten. This is calculated as:

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

where f_t is the forget gate vector, x_t is the current input, h_{t-1} is the hidden state from the previous time step, W_f and U_f are weight matrices, b_f is the bias, and σ is the sigmoid activation function. The input gate decides which new information to store in the cell state. It is computed using:

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

and the candidate cell state, which proposes new information to add, is calculated as:

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

The updated cell state is a combination of the retained information from the forget gate and the new candidate values modulated by the input gate. This is expressed as:

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

where c_t represents the updated cell state and $\odot$ denotes element-wise multiplication. Finally, the output gate determines the part of the cell state to pass to the hidden state, which is used as the output of the current time step. This is computed as:

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

and the hidden state is given by:

$$h_t = o_t \odot \tanh(c_t)$$

These mechanisms allow LSTMs to effectively capture both short-term and long-term dependencies in sequential data, which is critical for modeling the temporal dynamics of EV charging patterns. By leveraging these capabilities, LSTMs can predict power consumption more accurately than traditional methods, particularly in scenarios where temporal patterns are complex and non-linear.

2.6 Chebyshev Graph Convolutional Networks (Chebyshev GCN)

Chebyshev Graph Convolutional Networks (Chebyshev GCNs) are a spectral variant of Graph Convolutional Networks (GCNs) that leverage Chebyshev polynomials to approximate graph convolution operations. They address the computational inefficiencies of traditional spectral methods, such as the direct computation of eigenvalues and eigenvectors of the graph Laplacian, which is computationally prohibitive for large graphs. Chebyshev GCNs provide a scalable and efficient approach to learning from graph-structured data by approximating spectral filters in the graph domain.

Spectral Graph Convolutions Spectral graph convolution operates in the frequency domain and is defined using the graph Laplacian L, which encodes the structure of the graph. The Laplacian is expressed as:

$$L = D - A$$

where A is the adjacency matrix and D is the degree matrix. Alternatively, the normalized Laplacian is used:

$$\tilde{L} = I - D^{-1/2} A D^{-1/2}$$

The spectral convolution of a graph signal x with a filter g_θ is defined as:

$$g_\theta * x = U g_\theta(\Lambda) U^T x$$

where U and Λ are the eigenvectors and eigenvalues of $\tilde{L}$, respectively. However, computing the eigendecomposition of $\tilde{L}$ is computationally expensive, especially for large graphs.

Chebyshev Polynomial Approximation To overcome the computational challenges, Chebyshev GCNs approximate the spectral filter $g_\theta(\Lambda)$ using Chebyshev polynomials. Chebyshev polynomials $T_k(x)$ are recursively defined as:

$$T_0(x) = 1, \quad T_1(x) = x, \quad T_{k+1}(x) = 2xT_k(x) - T_{k-1}(x)$$

The filter g_θ is approximated as a K-order polynomial of the Laplacian:

$$g_\theta(L) \approx \sum_{k=0}^{K} \theta_k T_k(\tilde{L})$$

where θ_k are the learnable parameters of the model and $T_k(\tilde{L})$ represents the k-th order Chebyshev polynomial of the normalized Laplacian $\tilde{L}$.

Graph Convolution Operation The convolution operation for a node feature x using the Chebyshev approximation is given by:

$$g_\theta * x \approx \sum_{k=0}^{K} \theta_k T_k(\tilde{L})x$$

This formulation enables efficient computation by propagating information up to K-hop neighborhoods around each node without requiring explicit eigendecomposition. The value of K controls the extent of the neighborhood considered in the aggregation process, with higher values capturing more global information.

Applications in EV Charging Prediction In the context of EV charging networks, Chebyshev GCNs can effectively model spatial dependencies by aggregating features from nearby charging stations based on their geographical or operational relationships. By leveraging the Chebyshev polynomial approximation, the model achieves a balance between computational efficiency and the ability to capture localized patterns, making it well-suited for large-scale graph-based data such as power consumption in EV charging networks.

2.7 Graph Convolutional Long Short-Term Memory (GCLSTM) Networks

The Graph Convolutional Long Short-Term Memory (GCLSTM) network is a hybrid model that combines the strengths of Graph Convolutional Networks (GCNs) and Long Short-Term Memory (LSTM) networks. It is specifically designed to handle spatio-temporal data, where both spatial and temporal dependencies are critical. GCLSTMs integrate graph convolution operations into LSTM-like structures, enabling the model to capture spatial correlations from graph-structured data while simultaneously learning temporal patterns from sequential data.

The core idea behind GCLSTMs is to replace the traditional fully connected layers in the gates of LSTMs with graph convolution operations. This allows the model to aggregate information from neighboring nodes within the graph at each time step, enriching the temporal modeling with spatial context. For a node i at time t, the forget gate determines the extent to which the previous cell state should be retained or forgotten, and is defined as:

$$f_t^{(i)} = \sigma\left(\mathrm{GCN}(X_t, A)W_f + U_f H_{t-1} + b_f\right)$$

where $\mathrm{GCN}(X_t, A)$ represents the graph convolution operation applied to the input X_t with adjacency matrix A, H_{t-1} is the hidden state from the previous time step, W_f and U_f are weight matrices, and b_f is the bias vector. Similarly, the input gate determines the information to be added to the cell state, and is computed as:

$$i_t^{(i)} = \sigma\left(\mathrm{GCN}(X_t, A)W_i + U_i H_{t-1} + b_i\right)$$

The candidate cell state, which represents potential updates to the memory, is given by:

$$\tilde{C}_t^{(i)} = \tanh\left(\mathrm{GCN}(X_t, A)W_c + U_c H_{t-1} + b_c\right)$$

The updated cell state combines the retained information from the forget gate and the new candidate values modulated by the input gate, expressed as:

$$C_t^{(i)} = f_t^{(i)} \odot C_{t-1}^{(i)} + i_t^{(i)} \odot \tilde{C}_t^{(i)}$$

where $\odot$ denotes element-wise multiplication. The output gate determines the portion of the updated cell state to pass to the hidden state, which is computed as:

$$o_t^{(i)} = \sigma\left(\mathrm{GCN}(X_t, A)W_o + U_o H_{t-1} + b_o\right)$$

Finally, the hidden state, which is propagated to the next time step, is calculated as:

$$H_t^{(i)} = o_t^{(i)} \odot \tanh(C_t^{(i)})$$

In these equations, W, U, and b are learnable weights and biases, and σ is the sigmoid activation function. By incorporating graph convolution operations into the LSTM framework, GCLSTMs effectively capture both spatial and temporal dependencies in the data.

In the context of EV charging networks, GCLSTMs are particularly used for predicting power consumption. By modeling the spatial dependencies between charging stations through graph convolutions and capturing temporal dependencies using LSTM mechanisms, GCLSTMs might predict load profiles with greater accuracy. The adjacency matrix A, derived from spatial distances and correlations, encodes the relationships between charging stations, allowing the model to incorporate geographical and operational context into its predictions. This capability makes GCLSTMs a powerful tool for spatio-temporal prediction tasks in dynamic and interconnected systems like EV charging networks.

2.8 Evaluation Metrics

Evaluation metrics are crucial for assessing the performance of prediction models, providing quantitative measures of their accuracy and reliability. In this thesis, several standard metrics are employed to evaluate the predictive accuracy of models, including Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), Mean Absolute Percentage Error (MAPE), and R^2-score. These metrics enable a comprehensive comparison of models and their ability to capture patterns in power consumption at EV charging stations.

The **Mean Absolute Error (MAE)** quantifies the average magnitude of errors in predictions, irrespective of their direction. It is defined as:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n} |y_i - \hat{y}_i|$$

where y_i represents the actual value, $\hat{y}_i$ is the predicted value, and n is the total number of observations. MAE provides an intuitive interpretation of prediction accuracy in the same units as the target variable.

The **Root Mean Squared Error (RMSE)** penalizes larger errors more significantly by squaring the residuals, making it sensitive to outliers. It is expressed as:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}$$

While RMSE shares the same units as the target variable, its sensitivity to larger deviations makes it a valuable metric for applications where large errors are undesirable.

The **Mean Absolute Percentage Error (MAPE)** is a scale-independent metric that expresses the prediction error as a percentage of the actual value. It is defined as:

$$\text{MAPE} = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \times 100$$

MAPE is particularly useful for comparing model performance across datasets with varying scales, but it can be sensitive to very small actual values, which may lead to large percentage errors.

The R^2**-score**, or coefficient of determination, measures the proportion of variance in the target variable that is predictable from the input features. It is given by:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

where $\bar{y}$ is the mean of the actual values. The R^2-score ranges from 0 to 1, with higher values indicating better model performance.

These metrics collectively provide insights into the accuracy, robustness, and reliability of the prediction models. While MAE and RMSE focus on the magnitude of errors, MAPE offers a percentage-based evaluation, making it easier to interpret relative errors. The R^2-score highlights the model's ability to explain variance in the data. Together, these metrics form a comprehensive framework for evaluating the suitability of models for power consumption prediction tasks.

Chapter 3

Methodology

The below flowchart presents a step-by-step methodology for predicting load profiles at electric vehicle (EV) charging stations. It begins with Data Cleaning, which involves removing missing values, handling inconsistencies, and preparing the data for further analysis. Following this, Timestamp Resampling is performed, aligning data to a consistent time interval (e.g., 1-hour intervals) to maintain uniformity in temporal resolution. The next step, Feature Engineering, involves creating additional informative features such as time series decomposition elements (trend, seasonality, and residuals), and spatial features (e.g., distance to the city center). After engineering features, Data Splitting divides the dataset into training, validation, and testing subsets to facilitate robust model evaluation.

The methodology then progresses to Model Setup, where various predictive models (Persistence, Simple RNN, LSTM, GCLSTM) are chosen based on the problem's requirements. Model setup involves setting parameters, model training and validation follows, where the models learn patterns from the training data while being fine-tuned based on validation performance. The final step, Model Testing, assesses the model's performance on the test set, evaluating metrics like MAE, RMSE, mape, and score. This structured approach ensures a comprehensive analysis, from data preparation to model validation, for effective load profile prediction at EV stations.

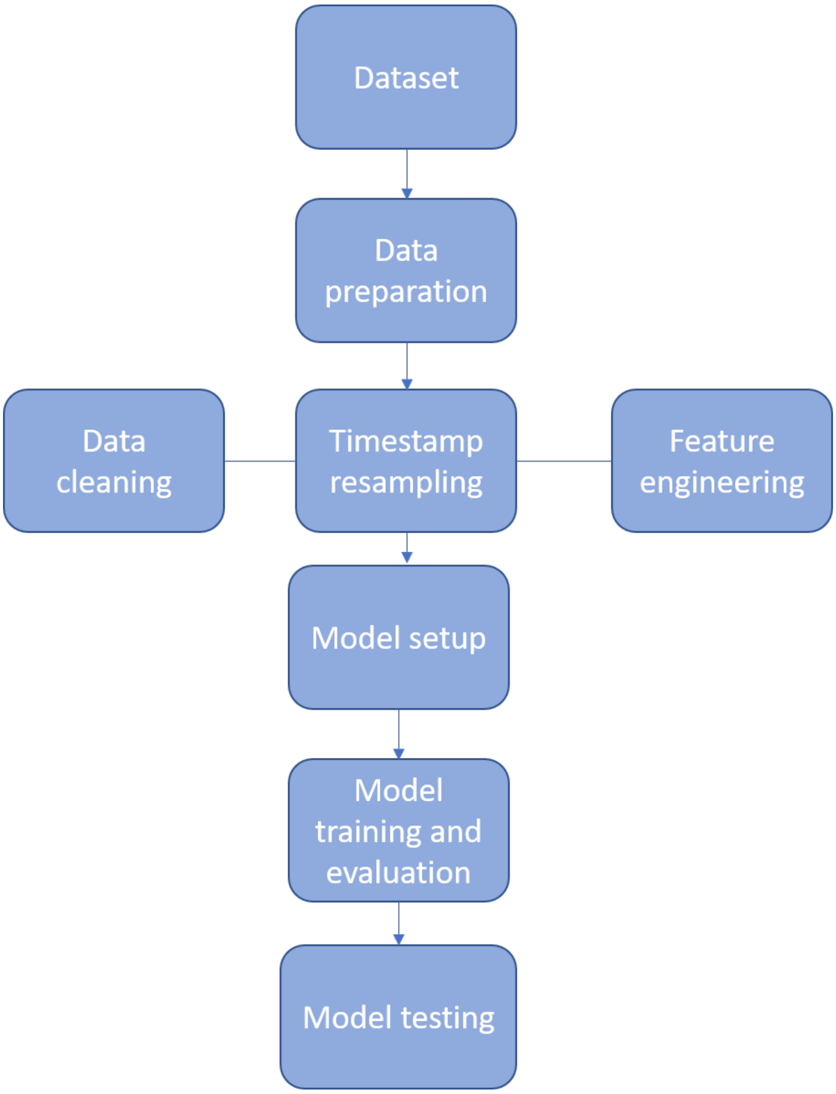

Figure 3.1: Methodology Flow Chart for Predicting EV Load Profiles

3.1 Description of the dataset

The dataset for this thesis is provided by Emotions SRL a start-up company based in Tuscany, Italy, which is part of an EU project Drive2X (https://drive2x.eu/) developing tools and technology related to mass EV adoption in the future. Electromobility Prediction Interactive Open-Source Tool (EPIOT) is a tool under development by the DLR Institute of Networked energy Systems, where my thesis is related to researching different methods to predict load profiles at charging stations using the datasets of the partners in the project with a goal to predict the next 24 hours ahead with historical 24 hours of data.

Out of all the 133 EV station datasets provided by Emotions SRL from Italian cities, the City of Galantina is selected due to the availability of more charging events at charging stations compared to other cities. As the raw dataset is basically an event-based dataset rather than a continuous time series, the no of charging events in Galantina compared to other cities has a large difference. Because of this reason, the city of Galantina is selected.

This study is based on real charging data of 63840 slow charging events with a maximum capacity of 22 kW, measured at 9 different charging stations in Galantina City, each specified by distinct latitude and longitude coordinates. These stations cover a diverse range of areas, providing a comprehensive view of spatial distribution and power usage patterns. The data spans a significant time period, from July 26, 2023, to April 17, 2024,, offering nearly a year of hourly records. This extensive temporal coverage allows for the analysis of variations and evolving trends in electric vehicle (EV) charging demand. The combination of geographical and temporal data provides a foundation for modeling load profiles, understanding fluctuations in power consumption, and drawing insights critical for optimizing the management and scaling of the EV charging network.
Figure 3.2 shows the locations of each station with their respective numbers. As shown on the map, stations 98 and 98 are present in the middle of the city, which will offer access to city residents. Station 175 will have different patterns because it is available near highway SP33, which will help travelers. Stations 213 to 217 are airport charging stations, which will have high usage during weekends. The city center stations 98 and 98 will have high usage on weekdays. Station 175 will have scheduled fleet activities as it is near the highway. On the right side of the map, the charging station 98 snapshot is added, as is the type of charging. Same as station 98, all the charging stations are of the same power rating, which is slow charging type.

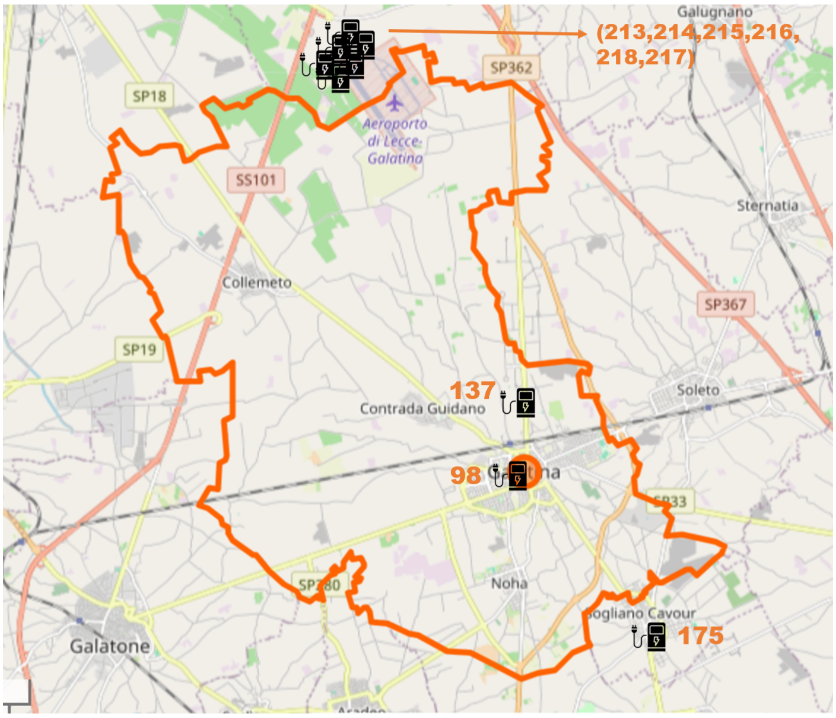

Figure 3.2: charging station locations

3.2 Data cleaning

Before utilizing the dataset for prediction and analysis, several data cleaning and filtering operations were performed to ensure the quality and integrity of the information. These steps are critical to eliminate inconsistencies, handle missing data, and prepare the dataset for accurate modeling. Below is the table explaining all attributes in the raw dataset.

Table 3.1: Description of raw Dataset Attributes for EV Charging Stations

Attribute	Description
`timestamp`	The exact date and time when the record was logged. Used for time series analysis and modeling.
`tower_latitude`	Geographic latitude of the charging station, used for spatial analysis.
`tower_longitude`	Geographic longitude of the charging station, used for spatial analysis.

Attribute	Description
`tower_id`	Unique identifier for the charging station, used to distinguish between different towers.
`kWh`	Amount of energy consumed during the charging session.

To prepare the dataset for analysis, several cleaning and filtering steps were conducted to ensure reliability and consistency. First, missing values were addressed: essential columns, such as timestamp, kWh, and tower_id, were checked for null entries. columns with a high percentage of missing data were removed. so, focusing only on attributes like tower_id, W, timestamp, and geographical coordinates. Next, timestamp formatting was standardized. The timestamp column was converted to a proper date-time format, enabling time-based operations such as resampling, aggregation, and filtering. Data consistency was also verified by removing duplicate records to eliminate redundant data points that could distort the analysis. Additionally, geographical coordinates (latitude and longitude) for each station were validated to check for any inaccuracies or mismatched locations. These cleaning and formatting steps transformed the raw charging records into a reliable, consistent dataset, establishing a solid foundation for load predictions.

3.3 Generation of continuous load profiles

The data contains both the plug-in- timestamps and the original amount of energy, $energy_{or}$,charged for each charging event.Consequently, initially, the original charging time, t_{or} , is derived by measuring the deviation between both timestamps. Furthermore, the charging load in Kilowatt , of each charging process is calculated according to below formulae:

$$load = \frac{energy_{\text{or}}}{t_{\text{or}}}$$

The dataset underwent aggregation and feature engineering to create a reliable, continuous load profile suitable for prediction models. The raw dataset consists of random time stamps, which vary from 1-minute to 5-minute intervals with a range of time stamps available One-hour resolution is the best resolution suitable for a number of charging events available. So, the Aggregation of charging records to hourly intervals helped smooth out anomalies, emphasizing general demand trends and enhancing model performance. power consumption (kW) values were summed for each hour across active charging sessions, creating a uniform time series of load profile data. Handling inactive periods by assigning zero kW during non-charging hours ensured continuity. Data resampling standardized timestamps to an hourly frequency, ensuring every hour in the study period was represented, while time zone consistency (UTC) prevented discrepancies from time zone differences. Feature engineering generated additional input variables to boost model accuracy. This included time decomposition and Spatial attributes such as station coordinates, distance to the city center, and inter-station distances captured location-specific demand trends. time decomposition into trend, seasonality, and residuals provided insight into regular patterns and periodic behaviors. So, Predictions were made on residuals, later combined with trend and seasonal components to produce final power consumption predictions. These transformations structured the raw data into a continuous, comprehensive load profile, meeting statistical requirements for time series modeling and enabling robust demand predictions.

3.4 Power usage trends based on distance to city center

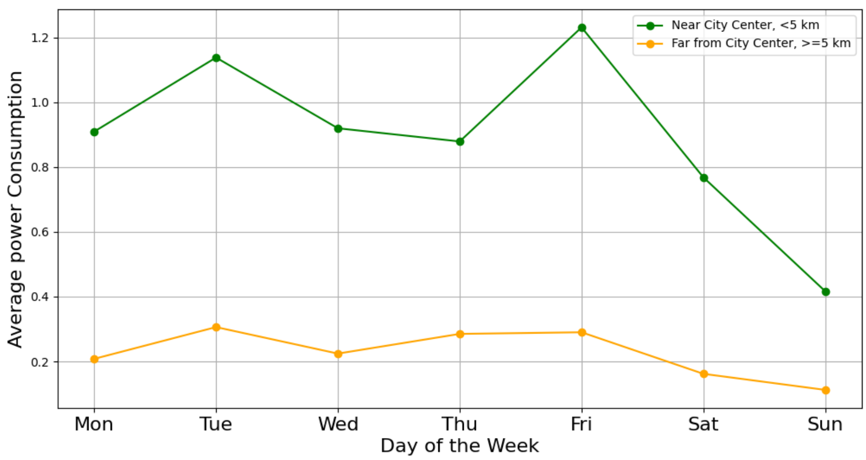

Figure 3.3: Average weekly power Consumption Patterns based on city center

The plot shows how the proximity of charging stations to the city center influences weekly power consumption patterns. Stations closer to the city center show higher power usage during weekdays, aligning with commuting patterns and regular daily activity, while usage tends to drop on weekends. In contrast, stations farther from the city center exhibit a reverse trend, with increased power consumption during weekends, likely driven by travel and leisure activities. These patterns suggest that charging station utilization is closely tied to their location, with weekday peaks reflecting work-related travel and weekend peaks corresponding to long-distance or recreational trips. Understanding these trends is crucial for optimizing station operations, scheduling maintenance, and implementing dynamic pricing strategies to meet varying demands efficiently.

3.5 Analysis of temporal patterns in dataset

In this section, we analyze power consumption patterns using STL decomposition (Seasonal-Trend Decomposition using Loess), which breaks the time series into three components: trend ($T(t)$), seasonality ($S(t)$), and residuals ($R(t)$). The decomposition for additive models is mathematically expressed as:

$$L(t) = T(t) + S(t) + R(t)$$

For this dataset with consideration of multiple seasonalities, the seasonal component is divided into daily ($S_{\text{daily}}(t)$) and weekly ($S_{\text{weekly}}(t)$) components, resulting in:

$$L(t) = T(t) + S_{\text{daily}}(t) + S_{\text{weekly}}(t) + R(t)$$

The trend ($T(t)$) is estimated using Loess smoothing on the detrended series, computed as $L(t) - S(t)$. The seasonal component ($S(t)$) is determined by grouping the detrended series ($L(t) - T(t)$) by periodicity (e.g., daily or weekly) and applying smoothing. For datasets with multiple seasonalities, the total seasonality is given by:

$$S(t) = S_{\text{daily}}(t) + S_{\text{weekly}}(t)$$

Finally, the residuals ($R(t)$) are calculated as the difference between the original series and the sum of the trend and seasonality:

$$R(t) = L(t) - [T(t) + S(t)]$$

STL iteratively refines these components, making it effective for handling noisy, non-stationary data. For datasets with a weekly cycle, STL captures weekly seasonality independently of the trend. This decomposition reveals long-term trends, periodic behaviors, and random fluctuations in power consumption, offering valuable insights into usage patterns [1iteFonteijn2019.

3.5.1 Time decomposition analysis

For analysis, station 98 is considered to understand the temporal patterns as an example.
The time decomposition of the time series data is presented as actual, trend, daily, weekly, and residual. Each of these components provides insights into different aspects of the data (Figure 3.1):

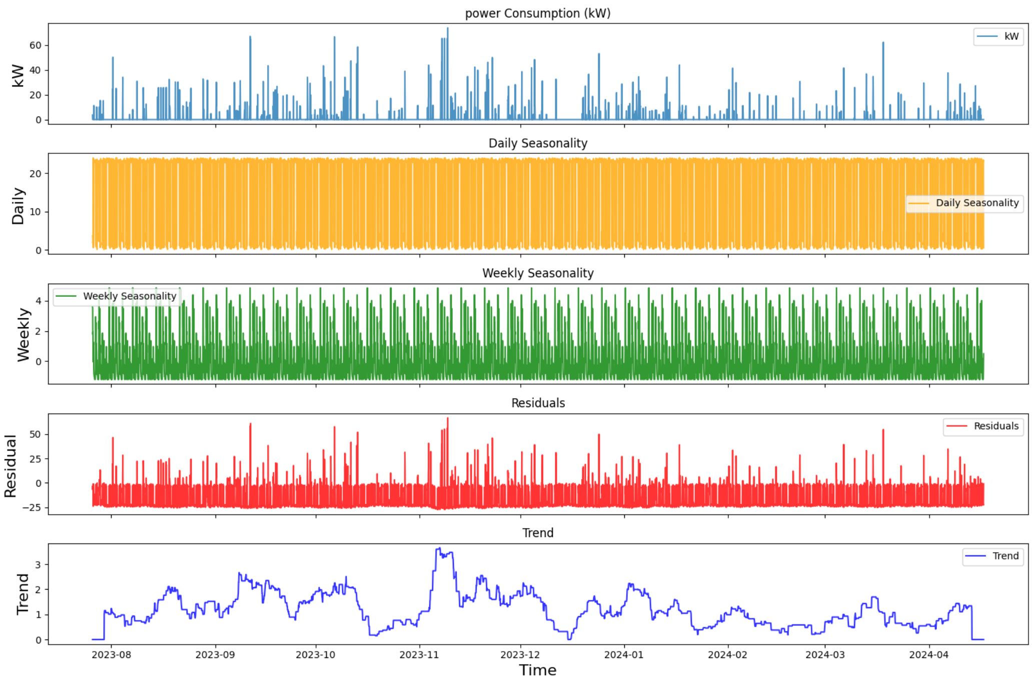

Figure 3.4: Time decomposition of station 98 in the dataset

(Figure 3.4) The decomposition highlights clear daily and weekly cycles in power usage, which can be leveraged for effective scheduling and demand prediction. The increasing trend during certain periods suggests growing station utilization or seasonal influence. Residuals indicate that while seasonality and trend capture much of the variability, occasional unpredictable spikes remain, underscoring the need for robust forecasting methods to address irregular usage patterns.

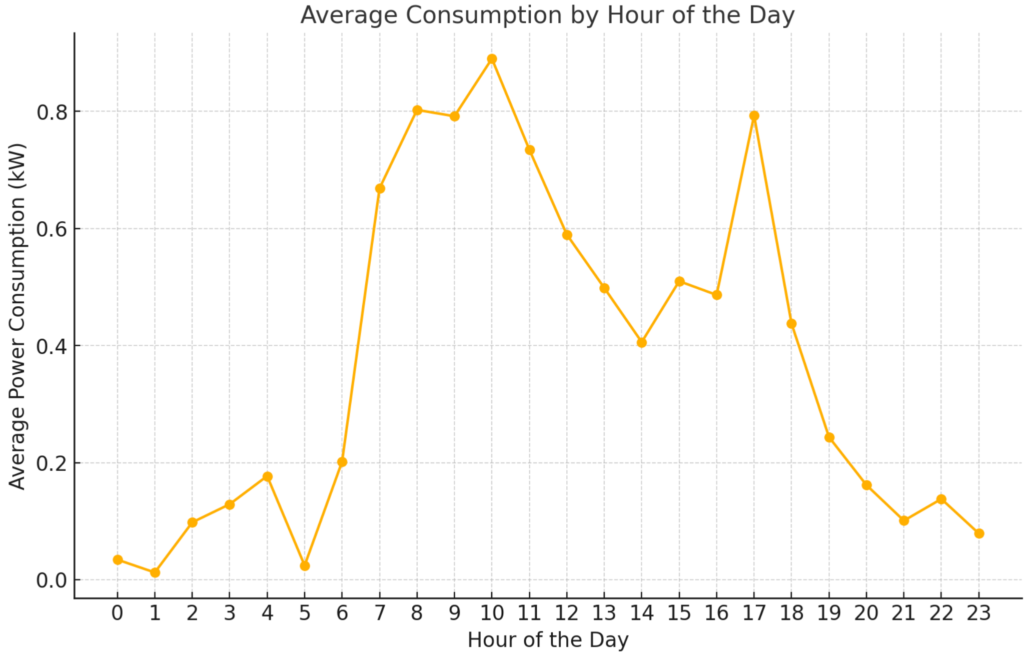

Figure 3.5: Average power consumption patterns on different hours of the day

Figure 3.5 shows how the daily cycle patterns, tell peak and off-peak hours. The daily seasonality in power consumption at station 98 follows a clear cyclic pattern influenced by human activity and transportation needs. power usage starts to increase in the early morning hours, reflecting charging demand as individuals prepare for their day. This trend peaks between late morning and midday, likely due to commuters and fleet vehicles utilizing the station. In the early afternoon, a decline in power consumption is observed, potentially as vehicles remain idle during work hours or after the morning rush. A secondary peak occurs in the late afternoon and early evening as individuals return home or prepare for evening travel. Finally, power usage significantly drops during the night, indicating minimal activity as most EVs are idle or fully charged. This pattern highlights the strong correlation between power consumption at EV stations and daily behavioral trends.

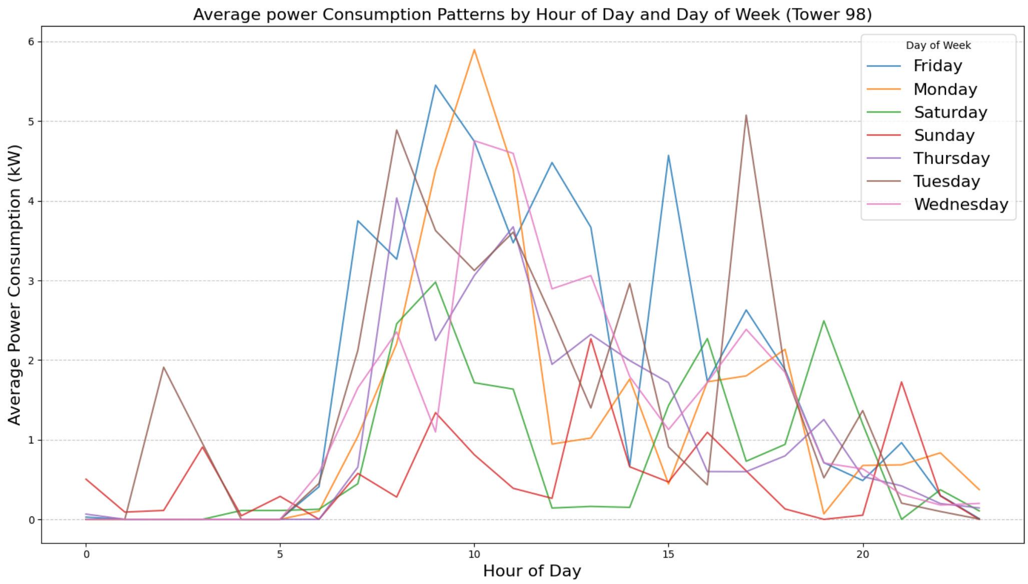

Figure 3.6: Average power consumption patterns on different days

Figure 3.6 shows the hourly average power consumption (kW) for each day of the week for station 98 in the dataset, revealing how power usage varies by both the hour and the day. Each line represents a different day of the week, highlighting distinct daily patterns in power demand at the EV charging stations. For example, weekdays such as Monday through Friday exhibit consistent power consumption patterns, with noticeable peaks during typical commuting hours, such as early morning and late afternoon, likely reflecting the behavior of EV users charging before and after work. In contrast, the weekends—Saturday and Sunday—show different profiles, with potentially later peaks and more gradual fluctuations, which could be attributed to more flexible travel and charging habits. Overall, the plot provides valuable insights into the temporal patterns of power usage, which can help optimize charging station operations and load profile management by aligning resources with expected demand throughout the week.

3.6 Model selection

In my thesis, model selection was driven by the objective of predicting the load profile at electric vehicle (EV) charging stations while accommodating the unique challenges of time series data. I opted for a combination of baseline and advanced models: persistence, Simple RNN, LSTM, and GCLSTM. The persistence model served as a straightforward benchmark, capturing recent trends for comparison. The Simple RNN model was chosen to explore recurrent neural network capabilities in handling temporal dependencies, while the LSTM model, known for its effectiveness with sequential data, aimed to capture longer-term dependencies and variations in EV charging demand. The GCLSTM model was selected to incorporate both temporal and spatial dependencies, which are crucial for this application due to the geographic spread of charging stations. This diverse set of models allowed for a comprehensive evaluation of different approaches, providing insights into their respective strengths and potential for improving load profile prediction in the EV sector.

Neural networks were selected over other model architectures due to the inherent complexities of EV charging demand data, which include non-linear patterns, temporal dependencies, and spatial variations across different stations. Traditional statistical models, like ARIMA or SARIMA, often excel at linear or seasonally structured time series but struggle to capture the non-linear interactions and dependencies within high-frequency data from multiple locations. In contrast, neural network architectures—particularly recurrent models like LSTM and GCLSTM—are adept at learning from sequential data, making them well-suited to recognize long-term dependencies and fluctuating patterns in EV charging profiles.

Moreover, neural networks offer flexibility in handling diverse input features, such as time series decomposition features and spatial coordinates, allowing a multi-dimensional approach to prediction that static models typically cannot achieve. This adaptability and capacity to learn complex relationships make neural networks a superior choice, particularly for data-driven applications with intricate and dynamic behavior like EV charging demand prediction.

3.7 Input features for the Models

After data preparation, the number of charging events occurring across the whole dataset is around 30% to 40% across all stations, and the rest of the values of the station's power consumption are zero. As there will be sharp peaks after zero consumption, it will be difficult for the models to have good accuracy to predict. So, instead of directly predicting power consumption as mentioned in the time decomposition analysis in section 3.5.1. we split the power consumption into a trend, daily seasonality, weekly seasonality, and residuals. Residual power consumption is used for the models and predicted, and later it is summed with the rest of the decomposed components. This approach is referred to in Fonteijn, R. and Castelijns [16]

$$P(t) = T(t) + S_d(t) + S_w(t) + R(t)$$

$P(t)$: Total power consumption at time t, $T(t)$: trend component, $S_d(t)$: Daily seasonality, $S_w(t)$: Weekly seasonality, $R(t)$: Residual component.

$$\hat{R}(t) \rightarrow \text{Predictive Model}$$

The residual $R(t)$ is used as the input to the predictive model to generate the predicted residual $\hat{R}(t)$.

$$\hat{P}(t) = T(t) + S_d(t) + S_w(t) + \hat{R}(t)$$

$\hat{P}(t)$: predicted total power consumption, $\hat{R}(t)$: Predicted residual.

The input features for the predictive models were carefully structured to capture relevant temporal and spatial dependencies. For the Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) models, the input features included residual power consumption and distance to the city center, represented as:

$$\mathbf{X}_{RNN/LSTM}(t) = [R(t), D]$$

where $R(t)$ denotes the residual power consumption at time t and D represents the distance to the city center.
For the Graph Convolutional Long Short-Term Memory (GCLSTM) model, additional spatial features, such as geographical coordinates, were incorporated to capture spatial dependencies across charging stations. The input feature structure for the GCLSTM model is given by:

$$\mathbf{X}_{GCLSTM}(t) = [R(t), D, \mathbf{G}]$$

where $\mathbf{G}$ denotes the geographical coordinates of the stations.
The predictive models focused on forecasting the residual power consumption, represented as:

$$\hat{R}(t) \rightarrow \text{Predictive Model}$$

where $\hat{R}(t)$ is the predicted residual at time t. The total power consumption was then reconstructed by summing the predicted residual with the trend ($T(t)$) and seasonal components ($S_d(t)$ for daily seasonality and $S_w(t)$ for weekly seasonality), as follows:

$$\hat{P}(t) = T(t) + S_d(t) + S_w(t) + \hat{R}(t)$$

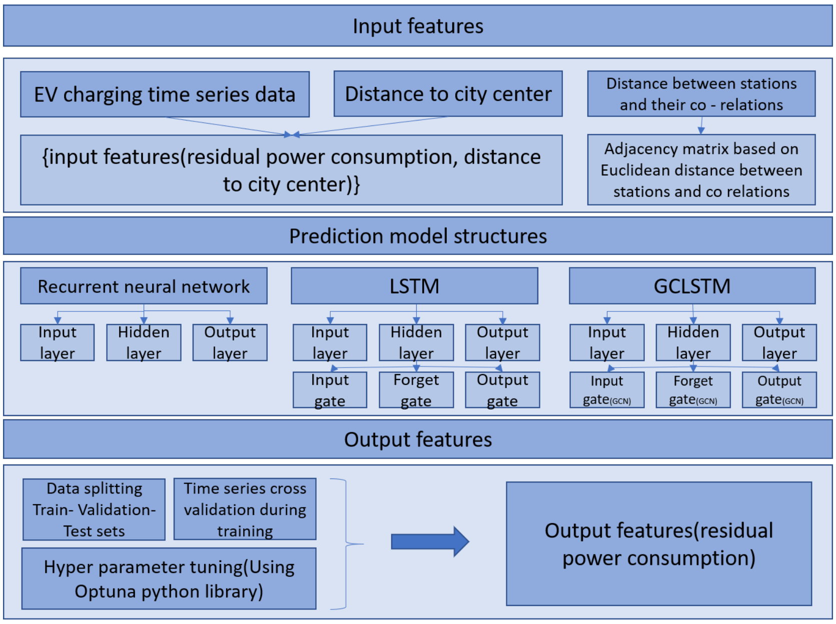

Figure 3.7: Illustration of the structure of input features, Model structures, and output features

3.8 Sequence creation

Sequence creation is essential for models like RNN, LSTM, and GCLSTM because these architectures are designed to learn temporal dependencies in sequential data. By feeding the model time-ordered sequences, it retains context from previous time steps, enabling it to capture short-term patterns (e.g., daily trends) and long-term dependencies (e.g., weekly seasonality). The recurrent structure of these models relies on historical input to influence predictions, making sequence creation vital for preserving the temporal patterns in data. Additionally, techniques like sliding window ensure the model receives continuous and overlapping sequences, enhancing its ability to learn fine-grained patterns. In GCLSTMs, sequence creation also integrates spatial information, allowing the model to analyze how temporal patterns evolve across interconnected nodes. Without sequences, these models cannot fully leverage their memory and state management mechanisms to predict time-dependent behaviors effectively.
The sequence creation step transforms continuous time-series data of EV station power consumption into structured input-target pairs, enabling the models to learn temporal patterns effectively. Using a sliding window approach, sequences are created by defining a sequence length (number of past

time steps), prediction length (number of future steps to forecast). For each station, a sequence of historical values forms the input, while the following data points (based on prediction length) serve as the target. This process iterates across the entire dataset, producing multiple overlapping input-target pairs. These sequences, organized as [sequence length, number of features] for inputs and [prediction length, number of features] for targets, allow the model to analyze patterns in past data to predict future consumption, thereby preparing the data for effective training and prediction.

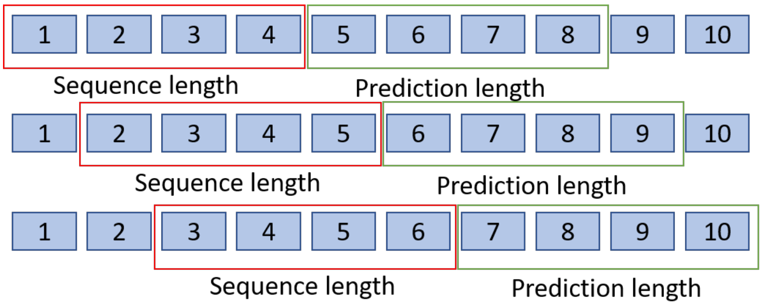

Figure 3.8: Example of how sequences are created for the dataset

For RNN and LSTM, the input pairs are created as [sequence length, number of features (Residual power (R(t)) & distance to city center] and target pairs are [prediction length, number of features (Residual power (R(t))]. For GCLSTM the input pairs are created as [sequence length, number of stations, number of features (Residual power (R(t)) & distance to city center] and target pairs are [prediction length, number of stations, number of features (Residual power (R(t))] for GCLSTM we need to predict all stations at once because we need to apply graph convolution parallel to capture temporal patterns.

3.9 Training, validation, and testing

The dataset is divided into three subsets: the training set spans from July 26, 2023, to January 9, 2024, the validation set from January 9, 2024, to February 27, and the test set from February 27 to April 17, 2024. Each subset consists of sequences, where each sequence is a 24-hour input window of prediction features paired with a 24-hour target consumption value. The sequences are created from the original time series data and assigned to the respective subsets based on their starting timestamps, ensuring no overlap between subsets and preserving temporal independence. Time-series cross-validation is applied to the training sequences to iteratively split them into smaller training and validation sets, improving the model's robustness and generalization. The datasets are structured as PyTorch time series tensors with dimensions as mentioned in Section 3.7 and enriched with spatial and temporal decomposition features to enable the model to capture complex patterns in EV charging demand effectively.

During the training and validation stage, time series cross-validation (TSCV) is applied to ensure robust performance while maintaining temporal integrity. During the training stage, TSCV splits data into sequential training and validation sets through a forward chaining approach. The model is trained on past data and validated on future data on each fold, mimicking real-world scenarios. This method ensures the model learns patterns according to real-world scenarios. During hyperparameter optimization, TSCV evaluates various configurations, such as learning rate or LSTM units, across multiple folds. This enables the selection of hyperparameters that generalize well, avoiding overfitting to specific time periods.

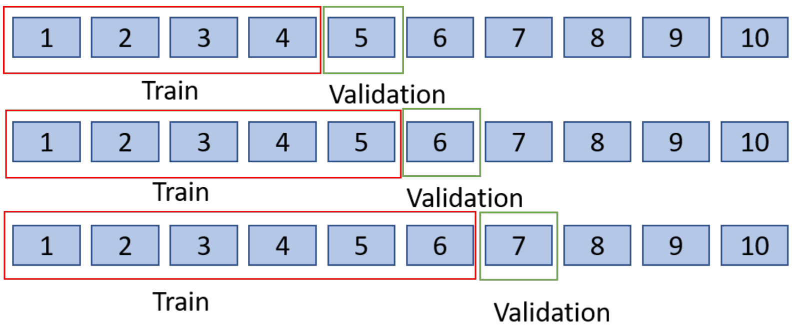

Figure 3.9: Example of how Time series cross-validation works

3.10 Hyper parameter optimization

Hyperparameter optimization is a crucial step in model development because the performance of a model heavily depends on chosen hyperparameters. These parameters, such as learning rates and number of hidden layers, are not learned during training but must be set before training begins. Incorrectly chosen hyperparameters can lead to overfitting or underfitting.Automating the optimization process ensures a systematic and efficient search for the best combination of parameters, which are important.

The Optuna Python library is used for hyperparameter optimization. Optuna employs Bayesian optimization through techniques like tree-structured parzen estimators, which enable it to balance exploration (trying uncertain values) and exploitation (refining promising areas of the search space). This approach is more efficient than grid or random search. as it dynamic pruning of underperforming trails significantly reduces computational overhead. making it ideal for resource-intensive tasks.

3.11 Model setup

3.11.1 Persistence Model

The rolling persistence model is a straightforward approach to time series prediction, operating similarly to the naive persistence model but with a slight variation. Instead of predicting the next value solely based on the last observation, it uses the average of the most recent observations within a specified time window, typically 24 hours, to account for minor fluctuations in the data. This model leverages the strong autocorrelation often present in time series data, making it a simple baseline for short-term load profile prediction. Despite its simplicity, the rolling persistence model serves as a useful benchmark to assess the performance of more advanced prediction models. However, the naive persistence model's limitations include the inability to account for long-term trends, seasonality, or external factors that may affect power consumption patterns.

3.11.2 Simple recurrent neural network

Simple recurrent neural network models (RNN) process sequential data through layered architecture for each station separately, beginning with an input layer that receives sequences of 24-time steps, and the output is also a sequence of 24-time steps. The input layer takes data in the shape $(n_{\text{batch}}, 24, n_{\text{features}})$, with each sequence representing 24-time steps of features such as historical power residuals and distance to the city center. The recurrent layer processes the input, maintaining a hidden state across the time steps. using activation functions like tanh which is internally available inside the model. To prevent overfitting, dropout is applied during training, randomly deactivating neurons, while L2 regularization penalizes large weights to improve generalization. Adam optimizer is used during the training. The hidden states at each time step are passed to a fully connected neural network layer, which then maps the processed data into a sequence of 24-hour predictions. The hyperparameters such as hidden size and learning rate are finalized after hyperparameter optimization using Optuna.

3.11.3 Long short-term memory

The LSTM model is a pytorch-based neural network designed for sequence modeling and time series forecasting. It consists of LSTM layers to capture temporal dependencies in the input data, followed by a fully connected layer to map the hidden state outputs to desired predictions. This model is applied to each station separately. The model takes input sequences of shape $(n_{\text{batch}}, 24, n_{\text{features}})$ and uses LSTM layers, initialized with zero hidden and cell states, to process the sequences. The output from the final time step of the LSTM is passed through the fully connected layer to generate predictions of shape $(n_{\text{batch}}, 24$. The same as in the above section to prevent overfitting L2 regularization and dropout are applied.

3.11.4 Graph convolution long short term memory

The GCLSTM is a hybrid graph convolution long short-term memory model that integrates LSTM sequence modeling capabilities with graph convolution layers to capture spatial dependencies in graph-structured data. It processes input sequences with dimensions $(n_{\text{batch}}, 24, n_{\text{features}}, n_{\text{stations}})$ and uses Chebyshev graph convolutions for the input, forget, cell and output gates to incorporate spatial relationships alongside temporal dynamics. The model initializes hidden and cell states and iteratively updates these states over the prediction length using LSTM equations enhanced by graph convolutions. The outputs include hidden states for each predicted timestep and the final cell state, enabling the model to handle spatiotemporal prediction tasks.

The GCLSTM processes input sequences through Chebyshev graph convolutions, which efficiently aggregate information from neighboring nodes in the graph while preserving the temporal sequence of data. Features are normalized using MinMax scaling to facilitate effective training, and hyperparameters such as filter size and learning rate are fine-tuned to optimize predictive accuracy. As mentioned in the above section, model performances are evaluated to provide a comprehensive assessment of its prediction capability. Visualizations comparing predicted and actual power demand further illustrate the model's effectiveness. This approach captures spatial and temporal trends in EV charging demand. This Model architecture is referred from the work done by R. Chen K. Eksombatchai P. Hamilton W. L. Leskovec, J. Ying, for dynamic link prediction[9]

Chapter 4

Results and Analysis

This section aims to provide an analysis of model performance, with a particular focus on assessing each model's capability to capture temporal dependencies and spatial features relevant to power consumption. Additionally, I employed Optuna (Hyperparameter Optimization), an automated hyperparameter optimization framework, to fine-tune model parameters, enhancing the performance of advanced models like Long Short-Term Memory (LSTM) and Graph Convolutional LSTM (GCLSTM). This optimization process ensures that the models achieve a generalization. The outputs from the models are residual power values which are summed with the trend, daily seasonal, and and weekly seasonal to generate predicted power consumption

Different metrics used for evaluating model performances are RMSE, MAE, and R^2score. Given its scale independence, the popular mean absolute percentage error (MAPE) would provide an easily interpretative error metric. However, the charging load profile time series exhibits a charging load of zero at numerous points in time. For this reason, MAPE cannot be used for overall comparison, as the calculation is based on the division of the error by the true load value at each timestep.

The neural networks are implemented using PyTorch (version 2.5.0). Plots are generated using Matplotlib (version 3.9.2) Datasets are handled using pandas (version 2.2.3) in a Python environment

4.1 Simple recurrent neural network

The Simple RNN model achieved an average MAE of 0.5494, with values of 0.5983 on Friday (2024-04-15), 0.4504 on Monday(2024-03-25), and 0.6531 on Saturday(2024-03-30). While the model maintains relatively low absolute errors. to decrease errors. Additionally, considering advanced models like LSTMs, which are better suited for capturing long-term dependencies, could reduce both MAE, leading to more accurate and reliable predictions. To improve performance more, further hyperparameter tuning, including adjustments to layers, units, and dropout, could help. Aiming for an MAE under 0.5 kW is a reasonable improvement goal with these adjustments. Now, for the upcoming neural network models, this model MAE is set as a a benchmark.

4.2 Long short-term Memory (LSTM)

The implemented LSTM model is designed to predict residual power values for EV stations, leveraging a simple yet effective architecture to capture temporal dependencies in the data. The model consists of an input layer, an LSTM layer with hidden units, and a fully connected layer that outputs predictions for stations. The LSTM processes the input sequences as mentioned in Section 3.11.3. After training and validating the model using cross-validation and hyperparameter techniques as mentioned in 3.9 and 3.10, below are the model parameters

Parameters	Training RMSE	Training MAE	Validation RMSE	Validation MAE
hidden_size: 112, num_layers: 2, lr: 0.000946, dropout rate = 0.3, L2_regularization = 0.2	0.05830952	0.0545	0.06557439	0.0698

Table 4.1: Hyperparameter Configurations and Their Metrics (computed on the scaled data)

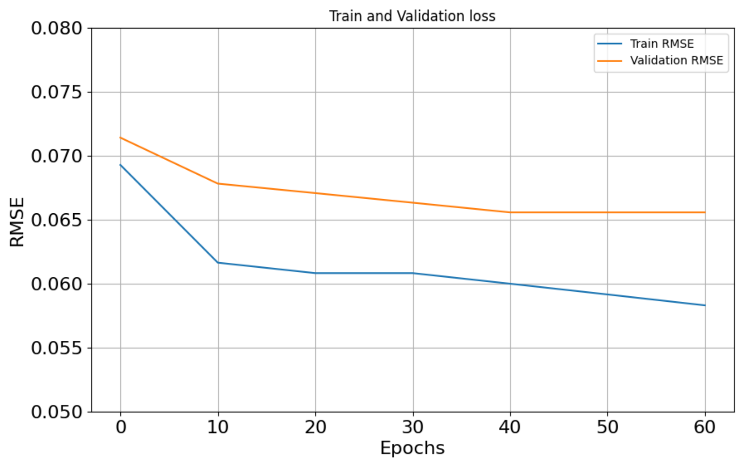

Figure 4.1: Training curves of LSTM model (blue line correspond to the training loss, orange line to the validation loss)

The "Train vs. Validation Loss" plot illustrates the training progression of a model over 60 epochs, comparing the root mean squared error (RMSE) loss for both the training and validation datasets. The training loss, represented by the blue line, steadily decreases throughout the epochs, indicating that the model is effectively learning and minimizing the error on the training set. This consistent downward trend demonstrates that the model parameters are being optimized with each iteration. On the other hand, the validation loss, shown as the orange line, decreases, reflecting an improvement in the model's ability to generalize to unseen data. However, it soon reaches a plateau and begins to exhibit minor fluctuations, suggesting that the model's generalization performance stabilizes after a certain point. The gap is observed between training and validation curves, which indicates that the networks have some difficulties generalizing the patterns that they observe on the training set. But as the error between the train and loss curves is not big, the model is working well with the unseen data.

4.3 Graph convolution long short-term memory (GCLSTM)

The implementation of the Graph Convolutional Long Short-Term Memory (GCLSTM) model is a comprehensive integration of graph convolution operations and LSTM-based temporal modeling. The model architecture is defined using several key parameters and components that ensure efficient learning from both spatial and temporal data. In this model, all stations power consumption is predicted at once instead of separately

The GCLSTM model is constructed with multiple layers that handle the input, forget, cell state, and output gates, each employing Chebyshev graph convolution (ChebConv) layers. These layers perform graph convolutions using Chebyshev polynomials, which efficiently capture the local structure of the graph and propagate information across nodes. The model includes a filter size parameter, K, which controls the size of the Chebyshev filters and affects the extent of neighborhood aggregation within the graph. In this implementation, K indicates neighborhood influence, while the number of output channels, out_channels represents the model's capacity to learn complex patterns across stations.

The model uses parameters like the in_channels, which represents the number of features for each node, and the out_channels, which defines hidden size. Additionally, the model includes learnable weights and biases for each gate, initialized using the Glorot and zero initializations for improved convergence.

The preprocessing stage includes constructing an adjacency matrix that captures the spatial relationships between EV stations. The adjacency matrix is then converted into a sparse edge index format suitable for PyTorch geometric operations. Furthermore, residual power consumption values are normalized using MinMax scaling to ensure that the model handles input data within a uniform range, enhancing the stability of the learning process.

Overall, this implementation highlights a data-driven approach that leverages graph-based learning and temporal sequence modeling, making it well-suited for the dynamic and spatially correlated nature of power consumption at EV charging stations.

Parameters	Training RMSE	Training MAE	Validation RMSE	Validation MAE
{out_channels: 128, K: 3, lr: 0.0095, dropout rate = 0.3, L2_regularization = 0.2 }	0.01299926	0.04593	0.02085906	0.05178

Table 4.2: Hyperparameter Configurations and Their Metrics (computed on the scaled data)

The GCLSTM model demonstrates strong performance with low training and validation RMSE (0.01299 and 0.02086) and MAE (0.04593 and 0.05178), indicating effective capture of spatiotemporal patterns. The small gap between training and validation metrics suggests good generalization with minimal overfitting, supported by well-tuned parameters like a learning rate of 0.001, dropout rate of 0.3, and graph filter size k = 3. Overall, the model is robust and well-suited for capturing complex charging patterns. The out channels are more than the LSTM because here we need more neurons due to the complexity of input structure as it is predicting all stations at once. filter size k = 3 indicates 3-hop neighbors is enough to generalize because, as mentioned in the geographical location in the map, out of 9 stations, the nearest neighborhood stations range from 1 to 5 But it also says that a maximum of 3 neighborhood stations have strong co-relationships and generalizations.

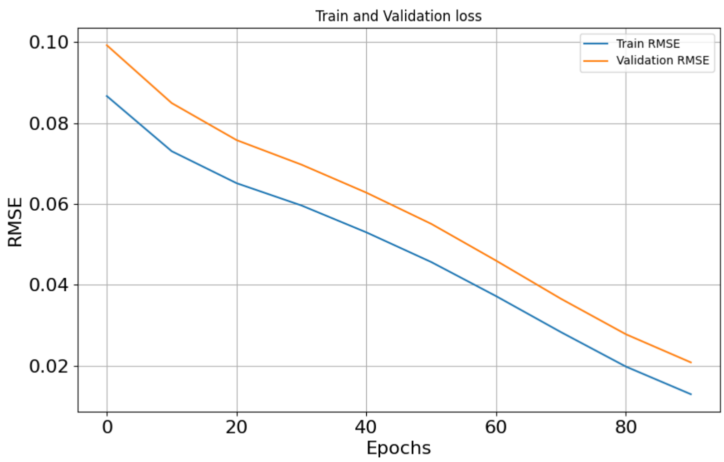

Figure 4.2: Training curves of LSTM model (blue line correspond to the training loss, orange line to the validation loss)

This plot shows the training and validation RMSE trends for the GCLSTM model over 90 epochs. Unlike the earlier case, the validation RMSE is consistently higher than the training RMSE, indicating a small generalization gap. Both metrics decrease steadily throughout the training process, demonstrating effective learning and convergence of the model. The parallel decline of both curves reflects that the model's capacity is well-suited to the problem, with no indications of overfitting. The smooth reduction of RMSE across epochs indicates that the model continues to refine its predictions effectively for both the training and validation sets.

4.4 Models performances

In this section, model performances are discussed for different days in the test dataset. For better evaluation, consider one weekday (2024-03-25, Monday), a weekend (2024-03-30, Saturday), and a day with high variability in the data across towers (2024-03-15, Friday). Before doing so, in order to evaluate the performance of the predictions made by LSTM and GCLSTM models, two metrics are used Mean absolute error (MAE). These are given by the formulas, which are explained in Section 2.7.

Initially, let's discuss model performances on average on testing samples. Table 4.3 displays the various metrics for different models, such as $R\hat{2}$ score and MAE.

The Long Short-Term Memory (LSTM) model outperforms all other models in this analysis, demonstrating its robustness in predicting power consumption data at EV charging stations. With an MAE of 0.3456, its R^2 value of 0.8461 shows that the model explains 84.61% of the variance in the data, reflecting its ability to effectively learn and capture the underlying patterns and long-term dependencies in sequential power data. This performance is largely due to the LSTM's unique architecture, which incorporates memory cells and gating mechanisms to manage the flow of information, ensuring that long-term dependencies are retained while irrelevant information is forgotten.

Model	$\mathbf{R^2}$	**MAE**
LSTM	0.8461	0.3456
GCLSTM	0.7361	0.4593
RNN	0.4560	0.5494

Table 4.3: Model Performance Metrics Averaged on Testing Samples

The Graph Convolutional LSTM (GCLSTM) model, while not as effective as the LSTM, still performs quite well. It has an MAE of 0.4593 with an R^2 value of 0.7361, meaning it explains 73.61% of the variance. The GCLSTM is specifically designed to handle spatiotemporal data, making it useful when spatial relationships between data points—such as distances between EV stations—are important. The model leverages Graph Convolutional Networks (GCNs) to understand spatial dependencies, combined with LSTM's temporal learning capabilities, which gives it an edge in scenarios where both spatial and temporal features play a significant role.

In contrast, the basic recurrent neural network (RNN) performs well with respect to data complexity. Its high error value MAE: 0.5494 and a low R^2 of 0.2753 indicate that it fails to capture high variance in data. This is primarily because traditional RNNs struggle with learning long-term dependencies due to the vanishing gradient problem, where important information from earlier time steps is lost as it propagates through the network. It will serve as a neural network model benchmark for the more advanced models

4.5 Numerical results

The performance comparison of RNN, LSTM, and GCLSTM models across stations and dates—2024-03-25 (Monday), 2024-03-15 (Friday), and 2024-03-30 (Saturday)—highlights the strengths and limitations of each approach in predicting EV station power consumption. LSTM consistently outperforms the other models, achieving lower errors across most stations and dates by effectively capturing temporal dependencies. GCLSTM performs competitively, excelling in stations with correlated power consumption patterns by leveraging spatial relationships, but it struggles slightly with isolated anomalies, such as sharp peaks in consumption, due to predicting all stations at once. Across models, higher mean absolute errors (MAE) are observed in stations with higher peak usage, reflecting better in capturing larger trends. Temporal variability further reveals that LSTM maintains stable performance across weekdays and weekends, whereas GCLSTM is sensitive to noise in spatial dependencies during busy periods. These results emphasize the importance of tailoring models to the temporal and spatial characteristics of EV station data, with LSTM providing robust generalizability and GCLSTM might help in scenarios requiring spatially-aware predictions. But with an extra dataset and with huge co-relation with adjacent stations, GCLSTM might outperform LSTM.

Station	RNN	LSTM	GCLSTM
98	0.512	0.297	0.441
137	0.301	0.263	0.247
175	0.291	0.282	0.277
213	0.287	0.193	0.256
214	0.268	0.2747	0.256
215	0.289	0.169	0.259
216	0.246	0.250	0.208
217	1.414	0.892	1.878
218	0.446	0.407	0.829

Table 4.4: Performance comparison of RNN, LSTM, and GCLSTM models across different stations on 2024-03-25(Monday).

Station	RNN	LSTM	GCLSTM
98	1.199	0.389	0.563
137	1.461	0.652	0.935
175	0.372	0.216	0.331
213	0.282	0.219	0.230
214	0.288	0.265	0.260
215	0.312	0.249	0.260
216	0.218	0.274	0.247
217	0.289	0.332	0.489
218	0.964	0.688	0.721

Table 4.5: Performance comparison of RNN, LSTM, and GCLSTM models across different stations on 2024-03-15(Friday).

Station	RNN	LSTM	GCLSTM
98	0.528	0.306	0.378
137	0.517	0.312	0.303
175	0.218	0.228	0.251
213	0.243	0.254	0.258
214	0.434	0.348	0.237
215	0.867	0.419	0.601
216	0.309	0.222	0.340
217	0.315	0.264	0.349
218	2.447	0.630	0.681

Table 4.6: Performance comparison of RNN, LSTM, and GCLSTM models across different stations on 2024-03-30(Saturday).

4.6 Graphical results

For visualization purposes, Figures 4.3, 4.4, and 4.5 illustrate the performance of both models (LSTM and GCLSTM) on each EV station on different days (weekends, weekdays, and random high variability days). Although the exhibited samples show that the models can track power consumption, they also display the weakness of the models, which is the inability to predict zero values exactly (such as in Figure 4.3 for stations 137, 175, 213, 214, 215, 216).

During the weekend (Saturday), stations 213–218, which are near the airport, have parallel activity across stations. Still, GCLSTM predicted power consumption peaks without any large errors, suggesting that GCLSTM helps in predicting a network of stations using their interstation relations.
For all the three days we can notice the repeated power consumption at station 98 during midday which is the reason LSTM is able to predict well because of the repeating patterns.

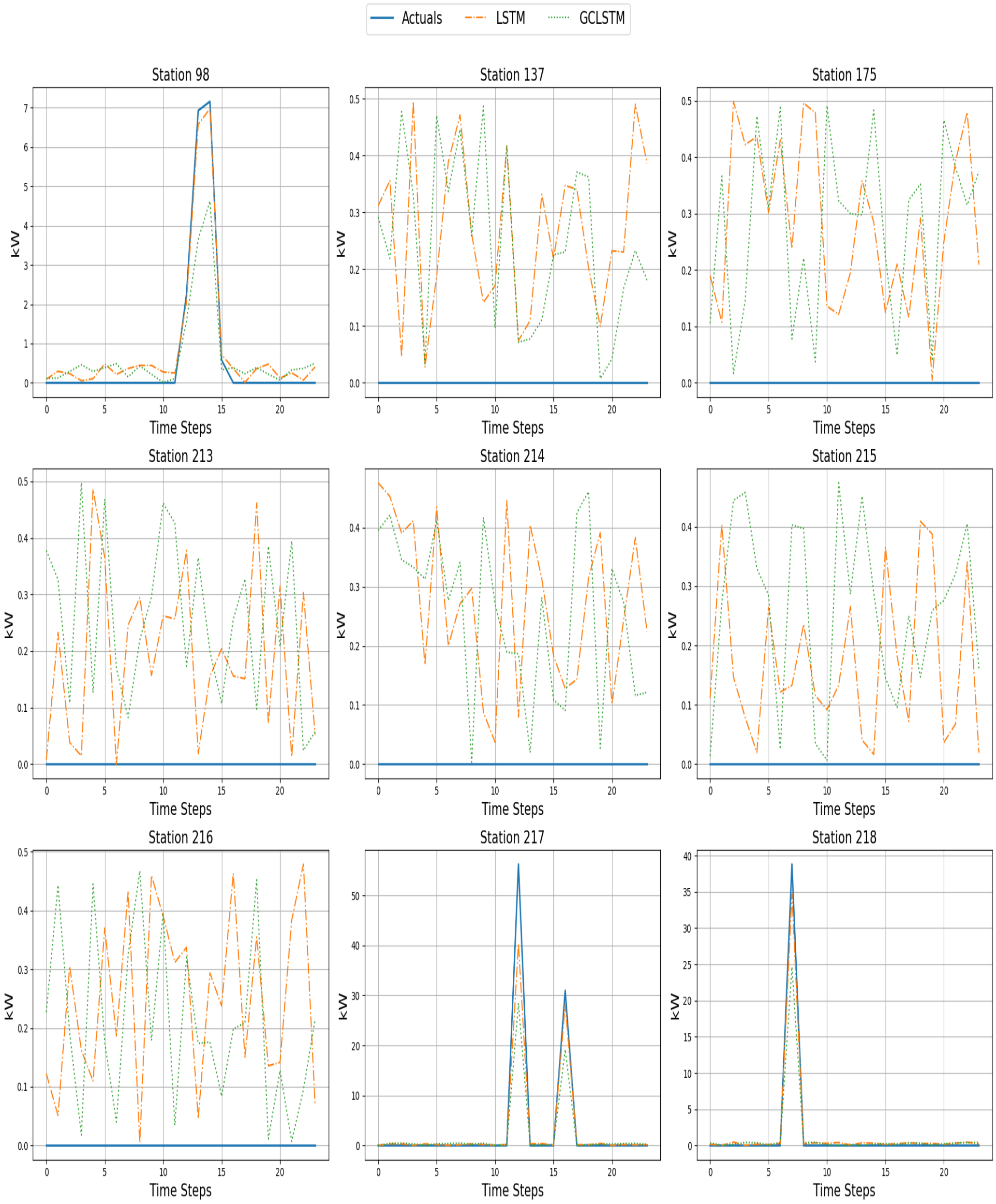

Figure 4.3: Comparison Of Model Predictions With Actual Values Over Time for Monday (2024-03-25)

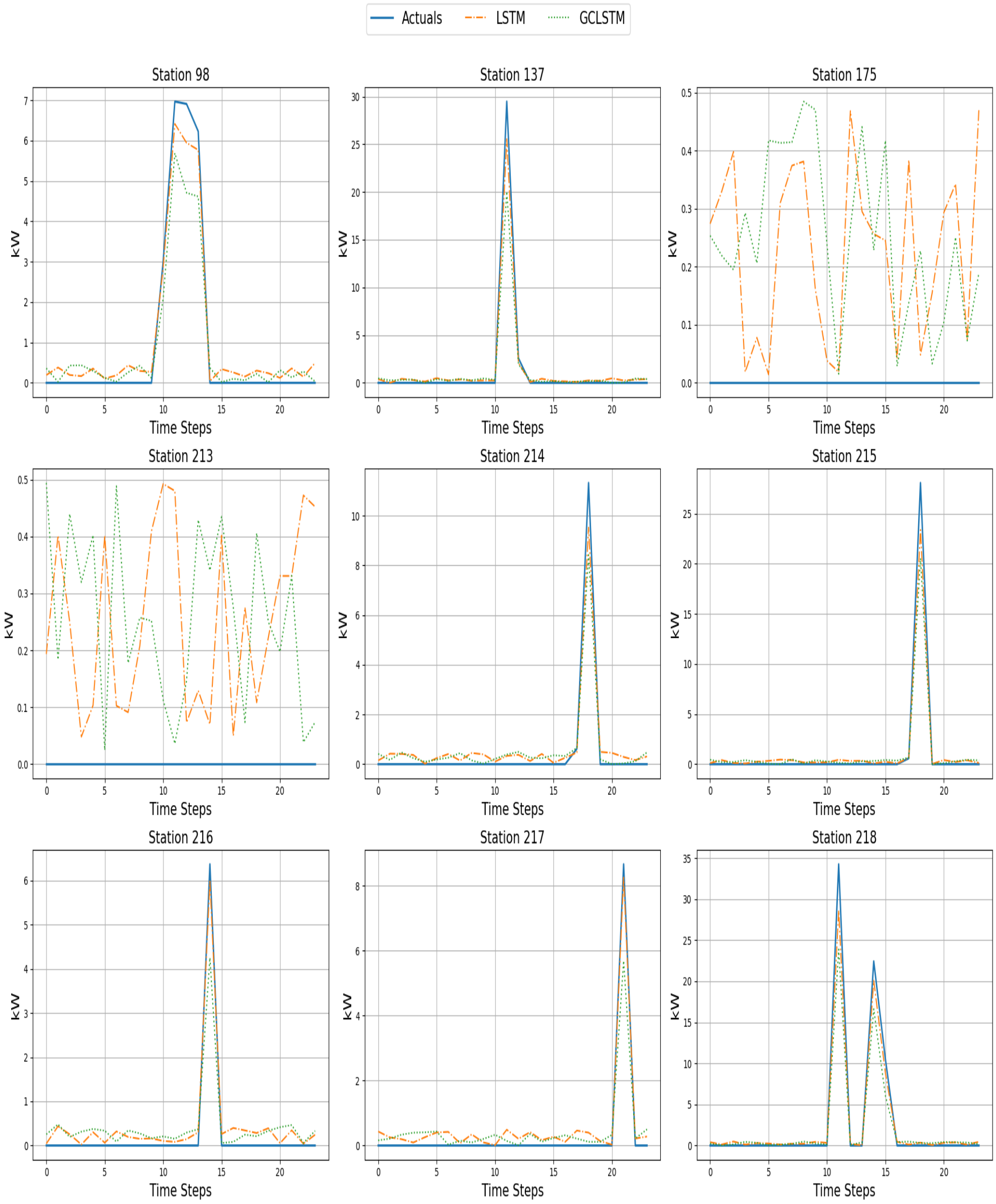

Figure 4.4: Comparison Of Model Predictions With Actual Values Over Time for Saturday (2024-03-30)

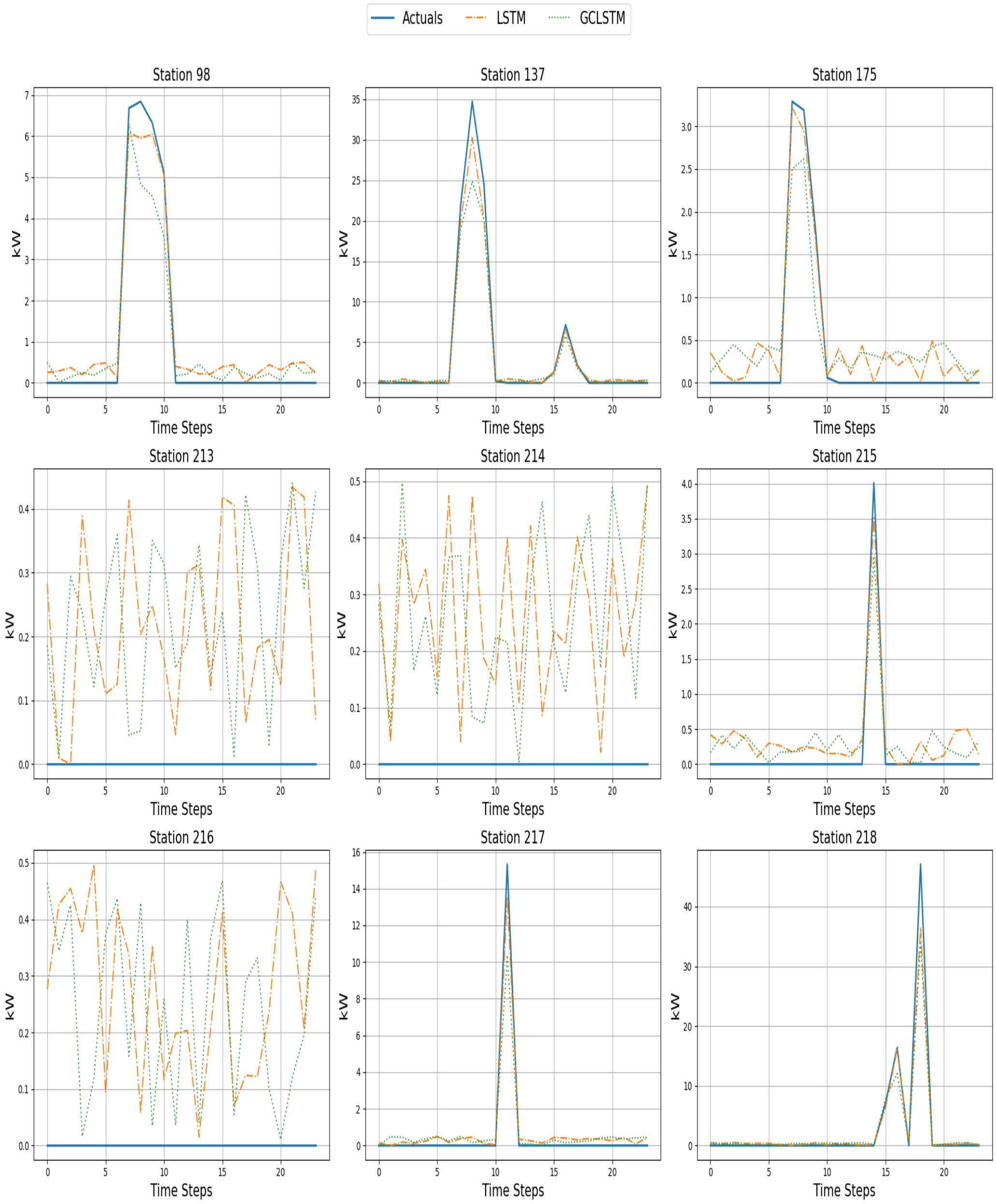

Figure 4.5: Comparison Of Model Predictions With Actual Values Over Time for Friday (2024-03-15)

Chapter 5

Discussion

This discussion chapter synthesizes the results of the time series prediction models evaluated in this study, addressing their comparative effectiveness and the limitations encountered throughout the research process. It provides a comprehensive interpretation of findings, the practical implications of model performance, and outlines constraints that influence the generalizability and applicability of these methods.

5.1 Time Series Model Comparison

The study employed time series prediction models for the power consumption at electric vehicle (EV) charging stations, including Long Short-Term Memory (LSTM), Graph Convolutional LSTM (GCLSTM), simple Recurrent Neural Network (RNN), and a naive persistence model. The LSTM model emerged as the most effective, achieving a mean absolute error (MAE) of 0.3456 while explaining 84.61% of the variance in the data ($R^2 = 0.8461$). which captures long-term dependencies essential for modeling temporal autocorrelation in power consumption patterns. The GCLSTM model, designed to handle both spatial and temporal dependencies, performed well but was slightly less accurate, with an $R^2 = 0.7361$. This model's ability to incorporate spatial features, such as the geographical distribution of EV stations, allowed it to capture spatial-temporal correlations, although minor prediction inaccuracies were noted, especially during peak demand times.

The basic RNN model, however, performed with a R^2 of only 0.4560. This performance is likely due to the vanishing gradient problem, which limits traditional RNNs in learning long-term dependencies, leading to significant discrepancies during peak periods. Overall, the LSTM model's performance highlights the importance of architectures that manage sequential dependencies in time series data when applied separately, while the GCLSTM model illustrates the potential of hybrid approaches where spatial factors play a crucial role in the data.

5.2 Feature importance

When predicting power use at EV charging stations, certain model features play key roles in capturing the unique patterns of demand. Temporal features, like recent power usage, are useful for capturing short-term and repeating daily or weekly patterns in charging demand. Spatial data, including the exact locations of stations and how close they are to the city center, helps capture how demand changes across different locations. Additionally, a spatial relationship feature called the adjacency matrix, is used in some models to learn from charging behavior at nearby stations, affecting predictions when stations influence each other. By breaking down time patterns into trends (long-term changes), seasonality (repeating cycles), and residuals (random noise), the model can better predict power use by handling stable patterns separately from unpredictable variations. Together, these features offer a balanced approach: temporal features are great for short-term predictions, while spatial data and trend-seasonality breakdowns help with longer-term and network-wide predictions.

5.3 Limitations

The research encountered several limitations. First, **data quality and availability** posed challenges, as incomplete and inconsistent data led to the need for resampling and cleaning. Missing timestamps and power consumption values introduced assumptions that may not perfectly reflect real-world behaviors, potentially impacting model accuracy. **Temporal and spatial variability** also added complexity, with power consumption patterns differing significantly across stations. Although features like geographic coordinates were included to address this variability, capturing all external factors influencing demand remained difficult, impacting model generalizability. Furthermore, **computational constraints** were a limiting factor; training deep learning models such as LSTM and GCLSTM required substantial computational resources and time, with hyperparameter tuning being particularly resource-intensive. This constraint makes real-time application challenging, especially in environments with limited resources. Additionally, **general-**

izability of the models is uncertain, as the dataset was specific to Italian cities, and different geographical or socio-economic settings may influence model performance. Validating these models across diverse contexts would be necessary to ensure robustness. Finally, **model complexity and interpretability** posed challenges for models like GCLSTM. Their sophisticated architecture, while effective, complicates interpretability, which may limit usability in decision-making where transparency is important for stakeholders in power management.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

This thesis studied the use of advanced models to predict power use at electric vehicle (EV) charging stations. By testing models like Long Short-Term Memory (LSTM), Graph Convolutional LSTM (GCLSTM), simple Recurrent Neural Networks (RNN), this research highlighted what each model can and cannot do when predicting loads for EV stations.

The results show that the LSTM model is the most accurate and reliable, as it effectively captures long-term patterns in power use. The GCLSTM model also performed well, as it can account for both location and time-based patterns, which is especially useful for networks of charging stations. However, GCLSTM required more computing power. In contrast, the basic RNN and persistence models serves as benchmark models, showing that simpler models struggle with the complexity of EV charging data and its difficulties to capture peaks in data.

This research highlights the benefits of using deep learning methods, especially for managing power consumption in urban areas. Still, there are challenges like data quality, model complexity, and high computing needs that need to be addressed to make these methods more practical and scalable. Future research should focus on improving data preparation, making models more efficient, and testing them in different locations and settings to ensure they work well in various real-world situations.

Overall if deep learning models need to be deployed for predicting power consumption. LSTM models should be considered when high accuracy is required and temporal patterns play more crucial role than spatial corelation. But GCLSTM's are ideal for interconnected network of stations where co relation between stations is crucial. Results depicts that if only temporal dependencies are considered the model performs well if spatial relationships are also considered there should be large number of charging events in the data which will help to co relate between stations and achieve better accuracy.

6.2 Future Work

There are several directions for future research to build on the findings of this thesis. Expanding and improving data by including a larger and more diverse dataset, covering various regions, climates, and socio-economic contexts, could make the models more robust and applicable in different real-world settings. Adding additional features, such as weather conditions, energy prices, and user behavior, could improve prediction accuracy, with future studies focusing on selecting the most relevant factors for power consumption prediction. To make these models more practical for real-time use, future research should work on reducing computational demands and increasing speed through methods like model pruning, quantization, and edge computing. Developing hybrid and ensemble models, which combine the strengths of different approaches, could lead to improved reliability and performance. Additionally, enhancing model transparency and interpretability would make these tools more accessible for energy managers, with techniques like SHAP and LIME providing insights into model decisions. Finally, integrating these prediction models with smart grid systems could enable demand-response strategies that improve grid stability and efficiency. By pursuing these research directions, the field can move closer to creating effective and intelligent energy management solutions that support the rising demand for EV infrastructure and contribute to sustainable urban energy systems.

Bibliography

[1] Mahnoosh Alizadeh, Anna Scaglione, and Ali Davoudi. Optimal real-time electric vehicle charging for renewable energy integration. *IEEE Transactions on Smart Grid*, 9(2):1518–1530, 2018.

[2] Towsley D. Atwood, J. "Diffusion-convolutional neural networks. In Advances in neural information processing systems". pp. 1993–2001, 2016.

[3] M. Aziz, T. Oda, T. Mitani, Y. Watanabe, and T. Kashiwagi. Utilization of electric vehicles and their used batteries for peak-load shifting. *Energies*, 8(5):3720–3738, 2015. Available at: `https://www.mdpi.com/1996-1073/8/5/3720`.

[4] Micheli A Podda M. A Bacciu D, Errica F. "Gentle introduction to deep learning for graphs.". https://pubmed.ncbi.nlm.nih.gov/32559609/, Sep, 2020.

[5] R.J. Bessa, C. Moreira, B. Silva, and M.A. Matos. Assessing the impacts of electric vehicle charging loads on the grid and mitigating strategies: A machine learning approach. *Energy*, 192:116685, 2020.

[6] Grattarola D. Livi L. Alippi C. Bianchi, F. M. "Graph neural networks with convolutional arma filters. IEEE Transactions on Pattern Analysis and Machine Intelligence.", 2021.

[7] S. Bouktif, A. Fiaz, A. Ouni, and M. A. Serhani. Optimal deep learning lstm model for electric load forecasting using feature selection and genetic algorithm: Comparison with machine learning approaches. *Energies*, 11(7), 2018. Available at: `https://www.mdpi.com/1996-1073/11/7/1636`.

[8] G. E. P. Box and D. R. Cox. An analysis of transformations. *Journal of the Royal Statistical Society. Series B (Methodological)*, 26:211–252, 1964. Available at: `http://www.jstor.org/stable/2984418`.

[9] Jinyin Chen, Xueke Wang, and Xuanheng Xu. Gc-lstm: Graph convolution embedded lstm for dynamic link prediction. *arXiv preprint arXiv:1812.04206*, 2018.

[10] Lin Y. Li W. Li P. Zhou J. Sun X. Chen, D. " Measuring and relieving the over-smoothing problem for graph neural networks from the topological view. In Proceedings of the aaai conference on artificial intelligence", 2020.

[11] T. Chen and C. Guestrin. Xgboost. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016. Available at: http://dx.doi.org/10.1145/2939672.2939785.

[12] Tao Chen, Lingzhi Jin, and Shiqi Ou. Electric vehicle charging infrastructure development: An overview. *Energy Reports*, 7:14–29, 2021.

[13] Xiaolong Chen, Jinming He, Yang Li, and Jian Shen. A comprehensive review on load forecasting and energy management in smart grids: Proposing a novel data-driven approach. *Energy Reports*, 6:2735–2750, 2020.

[14] Bresson X. Vandergheynst P. Defferrard, M. "Convolutional neural networks on graphs with fast localized spectral filtering. In Advances in neural information processing systems". pp. 3844–3852, 2016.

[15] Maclaurin D. Iparraguirre J. Bombarell R. Hirzel T. AspuruGuzik A. Adams R. P. Duvenaud, D. K. "Convolutional networks on graphs for learning molecular fingerprints. In Advances in neural information processing systems". pp. 2224–2232, 2015.

[16] Rik Fonteijn, Thomas Castelijns, Marinus Grond, Phuong Nguyen, Johan Morren, and Han Slootweg. Short-term load forecasting on mv/lv transformer level. 2019. Paper n° 559, Horizon 2020 InterFlex Project.

[17] Mohammad Ghofrani, Marzieh Esmaili, and Mahmood Etezadi-Amoli. Data analytics and machine learning applications in electric vehicle integration into smart grids: A survey. *Journal of Energy Storage*, 40:102819, 2021.

[18] Schoenholz S. S. Riley P. F. Vinyals O. Dahl G. E Gilmer, J. " Neural message passing for quantum chemistry. In Proceedings of the 34th international conference on machine learning (icml)". pp. 1263–1272., 2017.

[19] Bengio Y. Courville A. Goodfellow, I. "Deep learning. MIT Press.". http://www.deeplearningbook.org, 2016.

[20] Christopher Gough, Martyn Dick, David Vines, Gerardo Z. de Rubens, and Benjamin K. Sovacool. An exploration of energy demand and charging loads at an electric vehicle highway charging station. *Energy Policy*, 118:280–293, 2017.

[21] Ying Z. Leskovec J. Hamilton, W. "Inductive representation learning on large graphs. In Advances in neural information processing systems". (pp. 1024–1034), 2017.

[22] M. A. Hammad, B. Jereb, B. Rosi, and D. Dragan. Methods and models for electric load forecasting: A comprehensive review. *Logistics Sustainable Transport*, 11:51–76, 2020.

[23] Scott Hardman, Eric Shiu, and Robert Steinberger-Wilckens. Comparing high-end and low-end early adopters of battery electric vehicles. *Transportation Research Part A: Policy and Practice*, 122:99–111, 2019.

[24] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9:1735–1780, 1997.

[25] Junjie Hu, Hugo Morais, Tiago Sousa, and Morten Lind. Electric vehicle fleet management in smart grids: A review of services, optimization and control aspects. *Renewable and Sustainable Energy Reviews*, 56:1207–1226, 2016.

[26] R. Hyndman and G. Athanasopoulos. *Forecasting: Principles and Practice*. OTexts, 2nd edition, 2018.

[27] International Energy Agency (IEA). Global ev outlook 2021: Accelerating ambitions despite the pandemic, 2021. Available at: `https://www.iea.org/reports/global-ev-outlook-2021`.

[28] Zeyu Jiang, Jing Wu, Jun Zhang, Yan Li, and Pengfei Liu. Impact of ev load uncertainty on optimal planning for electric vehicle charging station. *Science China Technological Sciences*, 64:2032–2044, 2021.

[29] Minseok Kang and Wei Zhang. Multiscale spatio-temporal enhanced short-term load forecasting of electric vehicle charging stations. *arXiv preprint arXiv:2405.19053*, 2023.

[30] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu. Lightgbm: A highly efficient gradient boosting decision tree. 30, 2017. Available at: `https://proceedings.neurips.cc/paper/2017/file/6449f44a102fde848669bdd9eb6b76fa-Paper.pdf`.

[31] Welling Kipf, T. N. "Semi-supervised classification with graph convolutional networks. In Proceedings of the 5th international conference on learning representations (iclr).", 2017.

[32] Welling M. Kipf, T. N. "Semi-supervised classification with graph convolutional networks. In Proceedings of the 5th international conference on learning". ICLR, 2017.

[33] C. Kuster, Y. Rezgui, and M. Mourshed. Electrical load forecasting models: A critical systematic review. *Sustainable Cities and Society*, 35:257–270, 2017. Available at: `https://www.sciencedirect.com/science/article/pii/S2210670717305899`.

[34] X. Liao, N. Cao, M. Li, and X. Kang. Research on short-term load forecasting using xgboost based on similar days. In *2019 International Conference on Intelligent Transportation, Big Data Smart City (ICITBS)*, pages 675–678, 2019.

[35] Yang Liu, Yanbo Xu, Ruifeng Li, and Mingshan Qiu. Deep learning for electric vehicle charging load prediction in urban distribution systems. *IET Generation, Transmission & Distribution*, 12(21):5712–5718, 2018.

[36] Henrik Lund and Willett Kempton. Integration of renewable energy into the transport and electricity sectors through v2g. *Energy Policy*, 39(6):357–356, 2015.

[37] S. M. Lundberg, G. G. Erion, and S. Lee. Consistent individualized feature attribution for tree ensembles. *CoRR*, abs/1802.03888, 2018. Available at: `http://arxiv.org/abs/1802.03888`.

[38] S. M. Lundberg and S.-I. Lee. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems*, volume 30, 2017. Available at: `https://proceedings.neurips.cc/paper/2017/file/8a20a8621978632d76c43dfd28b67767-Paper.pdf`.

[39] Shuzhen Luo and Jianhui Wang. Deep-learning-based probabilistic forecasting of electric vehicle charging load with a novel queuing model. *IEEE Transactions on Cybernetics*, 51(3):1561–1571, 2021.

[40] Francis Mwasilu, Jackson J. Justo, Eun-Kyung Kim, Tae-Jun Do, and Jong-Wook Jung. Electric vehicles and smart grid interaction: A review on vehicle to grid and renewable energy sources integration. *Renewable and Sustainable Energy Reviews*, 34:501–516, 2014.

[41] K. Oureilidis, K.-N. Malamaki, K. Gallos, A. Tsitsimelis, C. Dikaiakos, S. Gkavanoudis, M. Cvetkovic, J. M. Mauricio, J. M. Maza Ortega, J. L. M. Ramos, G. Papaioannou, and C. Demoulias. Ancillary services market design in distribution networks: Review and identification of barriers. *Energies*, 13(4), 2020. Available at: `https://www.mdpi.com/1996-1073/13/4/917`.

[42] Chen Qin, Hui Zhang, Jian Liu, and Yifan Wang. Application of artificial intelligence for ev charging and discharging scheduling and dynamic pricing: A review. *Energy Reports*, 8:4841–4857, 2022.

[43] Ghazal Razeghi, Behrouz Babadi, Matthias Galus, and Göran Andersson. Smart charging of plug-in electric vehicles to minimize charging cost in an urban distribution network. *Applied Energy*, 197:237–246, 2017.

[44] Gori M. Tsoi A. C. Hagenbuchner M. Monfardini G. Scarselli, F. "graph neural network model. IEEE Transactions on Neural Networks,". pp. 61–80., 2008.

[45] J. Schlund, R. Steinert, and M. Pruckner. Coordinating e-mobility charging for frequency containment reserve power provision. pages 556–563, 2018.

[46] M. Schuster and K. K. Paliwal. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681, 1997.

[47] S. Siami-Namini, N. Tavakoli, and A. S. Namin. The performance of lstm and bilstm in forecasting time series. In *2019 IEEE International Conference on Big Data (Big Data)*, pages 3285–3292, 2019.

[48] Benjamin K. Sovacool, Jonn Axsen, and Willett Kempton. The future promise of vehicle-to-grid (v2g) integration: A sociotechnical review and research agenda. *Annual Review of Environment and Resources*, 43(1):377–406, 2018.

[49] Daniel-Ioan Stroe, Maciej Swierczynski, Vaclav Knap, Remus Teodorescu, and Dirk Uwe Sauer. Operation of grid-connected battery energy

storage systems: A review. *IEEE Transactions on Sustainable Energy*, 10(1):438–454, 2019.

[50] J. Tomić and W. Kempton. Using fleets of electric-drive vehicles for grid support. *Journal of Power Sources*, 168(2):459–468, 2007. Available at: `https://www.sciencedirect.com/science/article/pii/S0378775307005575`.

[51] Jianhui Wang, Changzheng Liu, Wei Li, Wei Wei, and Pu Wang. Comprehensive review of smart charging strategies and optimal energy management models for electric vehicles. *Energy Storage and Sustainable Development*, 10(2):58–69, 2021.

[52] Yingchun Wang, Kun Qian, Mengying Han, and Xiaolin Peng. Short-term load forecasting of electric vehicle charging stations based on hybrid machine learning model. *Energy*, 191:116548, 2020.

[53] Zhenyu Wang, Cheng Liu, and Ying Zha. Load forecasting for electric vehicle charging stations using a data-driven approach. *IEEE Transactions on Transportation Electrification*, 6(3):123–133, 2020.

[54] D. Wei, C. Zhang, B. Sun, and N. Cui. A time-of-use price based multi-objective optimal dispatching for charging and discharging of electric vehicles. *Dianwang Jishu/Power System Technology*, 38:2972–2977, 2014.

[55] Rick Wolbertus, Maarten Kroesen, Robert van den Hoed, and Caspar Chorus. Policy effects on charging behaviour of electric vehicle owners and on purchase intentions of prospective owners: Natural and stated choice experiments. *Transportation Research Part D: Transport and Environment*, 78:102–168, 2020.

[56] M. Xue, L. Wu, Q. P. Zhang, J. X. Lu, X. Mao, and Y. Pan. Research on load forecasting of charging station based on xgboost and lstm model. In *Journal of Physics: Conference Series*, volume 1757, page 012145, 2021.

[57] Wenjing Yao, Hongyu Zhao, Peng Shi, and Zhengping Li. Data-driven load forecasting model for electric vehicle charging stations in smart grid. *Journal of Modern Power Systems and Clean Energy*, 7(3):531–542, 2019.

[58] He R. Chen K. Eksombatchai P. Hamilton W. L. Leskovec J. Ying, R. "Graph convolutional neural networks for web-scale recommender

systems. In Proceedings of the 24th acm sigkdd international conference on knowledge discovery data mining". (pp. 974–983), 2018.

[59] Chunyu Zhang, Lei Xie, Yi Gu, Guoyang Bai, and Hongyu Zhou. Coordinated control for power management of ev charging stations integrated with pv and battery energy storage systems. *IEEE Transactions on Industrial Informatics*, 14(11):5101–5110, 2018.

[60] W. Zhang, D. Zhang, B. Mu, L. Y. Wang, Y. Bao, J. Jiang, and H. Morais. Decentralized electric vehicle charging strategies for reduced load variation and guaranteed charge completion in regional distribution grids. *Energies*, 10(2), 2017. Available at: `https://www.mdpi.com/1996-1073/10/2/147`.

[61] Jianzhi Zhao, Wee Keong Lee, Zhongming Zeng, and Hongbin Sun. Data-driven load profiling and anomaly detection in electric vehicle charging data. *IEEE Transactions on Smart Grid*, 9(4):3556–3565, 2018.

[62] Cui G. Zhang Z. Yang C. Liu Z. Wang L. . . . Sun M. Zhou, J. "Graph neural networks: A review of methods and applications.". arXiv preprint arXiv:1812.08434 ., 2018.

[63] J. Zhu, Z. Yang, M. Mourshed, Y. Guo, Y. Zhou, Y. Chang, Y. Wei, and S. Feng. Electric vehicle charging load forecasting: A comparative study of deep learning approaches. *Energies*, 12:2692, 2019.

[64] Agrawal M. Leskovec J. Zitnik, M. "Modeling polypharmacy side effects with graph convolutional networks. Bioinformatics,". i457–i466, 2018.

Appendix A

AI tools used for thesis report

- Grammarly : proper sentence formation.
- Quillbot : plagarism checking and citations.
- Writefull : for further improving grammar re phrasing the senetences.
- ChatGpt : Used as a tool to understand the basics of the Deep learning topics.

List of Figures

List of Tables

Universität Bremen

Nachname	**Mantri**	Matrikelnr.	**6088866**
Vorname	**Hrushikesh**		

Hinweise zu den offiziellen Erklärungen

1. Die folgende Seite mit den offiziellen Erklärungen
   - **A)** Eigenständigkeitserklärung
   - **B)** Erklärung zur Veröffentlichung von Bachelor- oder Masterarbeiten
   - **C)** Einverständniserklärung über die Bereitstellung und Nutzung der Bachelorarbeit / Masterarbeit in elektronischer Form zur Überprüfung durch eine Plagiatssoftware

   ist entweder direkt in jedes Exemplar der Bachelor- oder Masterarbeit fest mit einzubinden oder unverändert im Wortlaut in jedes Exemplar der Bachelor- oder Masterarbeit zu übernehmen.

   Bitte achten Sie darauf, jede Erklärung in allen drei Exemplaren der Arbeit zu unterschreiben.

2. In der digitalen Fassung kann auf die Unterschrift verzichtet werden. Die Angaben und Entscheidungen müssen jedoch enthalten sein.

Zu B)

Die Einwilligung kann jederzeit durch Erklärung gegenüber der Universität Bremen, mit Wirkung für die Zukunft, widerrufen werden.

Zu C)

Das Einverständnis der dauerhaften Speicherung des Textes ist freiwillig.

Die Einwilligung kann jederzeit durch Erklärung gegenüber der Universität Bremen, mit Wirkung für die Zukunft, widerrufen werden.

Weitere Informationen zur Überprüfung von schriftlichen Arbeiten durch die Plagiatsoftware sind im Nutzungs- und Datenschutzkonzept enthalten. Diese finden Sie auf der Internetseite der Universität Bremen.

Universität
Bremen

Nachname	Mantri	Matrikelnr.	6088866
Vorname	Hrushikesh		

A) Eigenständigkeitserklärung

Ich versichere, dass ich die vorliegende Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe. Alle Teile meiner Arbeit, die wortwörtlich oder dem Sinn nach anderen Werken entnommen sind, wurden unter Angabe der Quelle kenntlich gemacht. Gleiches gilt auch für Zeichnungen, Skizzen, bildliche Darstellungen sowie für Quellen aus dem Internet, dazu zählen auch KI-basierte Anwendungen oder Werkzeuge. Die Arbeit wurde in gleicher oder ähnlicher Form noch nicht als Prüfungsleistung eingereicht. Die elektronische Fassung der Arbeit stimmt mit der gedruckten Version überein. Mir ist bewusst, dass wahrheitswidrige Angaben als Täuschung behandelt werden.

☒ Ich habe KI-basierte Anwendungen und/oder Werkzeuge genutzt und diese im Anhang "Nutzung KI basierte Anwendungen" dokumentiert.

B) Erklärung zur Veröffentlichung von Bachelor- oder Masterarbeiten

Die Abschlussarbeit wird zwei Jahre nach Studienabschluss dem Archiv der Universität Bremen zur dauerhaften Archivierung angeboten. Archiviert werden:

1) Masterarbeiten mit lokalem oder regionalem Bezug sowie pro Studienfach und Studienjahr 10 % aller Abschlussarbeiten
2) Bachelorarbeiten des jeweils ersten und letzten Bachelorabschlusses pro Studienfach und Jahr.

☒ Ich bin damit einverstanden, dass meine Abschlussarbeit im Universitätsarchiv für wissenschaftliche Zwecke von Dritten eingesehen werden darf.

☒ Ich bin damit einverstanden, dass meine Abschlussarbeit nach 30 Jahren (gem. §7 Abs. 2 BremArchivG) im Universitätsarchiv für wissenschaftliche Zwecke von Dritten eingesehen werden darf.

☐ Ich bin nicht damit einverstanden, dass meine Abschlussarbeit im Universitätsarchiv für wissenschaftliche Zwecke von Dritten eingesehen werden darf.

C) Einverständniserklärung zur Überprüfung der elektronischen Fassung der Bachelorarbeit / Masterarbeit durch Plagiatssoftware

Eingereichte Arbeiten können nach § 18 des Allgemeinen Teil der Bachelor- bzw. der Masterprüfungsordnungen der Universität Bremen mit qualifizierter Software auf Plagiatsvorwürfe untersucht werden.
Zum Zweck der Überprüfung auf Plagiate erfolgt das Hochladen auf den Server der von der Universität Bremen aktuell genutzten Plagiatssoftware.

☒ Ich bin damit einverstanden, dass die von mir vorgelegte und verfasste Arbeit zum oben genannten Zweck dauerhaft auf dem externen Server der aktuell von der Universität Bremen genutzten Plagiatssoftware, in einer institutionseigenen Bibliothek (Zugriff nur durch die Universität Bremen), gespeichert wird.

☐ Ich bin nicht damit einverstanden, dass die von mir vorgelegte und verfasste Arbeit zum o.g. Zweck dauerhaft auf dem externen Server der aktuell von der Universität Bremen genutzten Plagiatssoftware, in einer institutionseigenen Bibliothek (Zugriff nur durch die Universität Bremen), gespeichert wird.

03/12/2024	M Hrushikesh
Datum	Unterschrift