Identification of Loads Based on Historical Data

A Machine Learning Approach to Segmenting and Forecasting Electricity Demand in Response to Price Signals

Master's thesis in Computer science and engineering

Meixi Lin
Pirapon Supakkeittikul

Department of Computer Science and Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
UNIVERSITY OF GOTHENBURG
Gothenburg, Sweden 2025

Master's thesis 2025

Identification of Loads Based on Historical Data

A Machine Learning Approach to Segmenting and Forecasting Electricity Demand in Response to Price Signals

Meixi Lin

Pirapon Supakkeittikul

UNIVERSITY OF GOTHENBURG

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Computer Science and Engineering
Chalmers University of Technology
University of Gothenburg
Gothenburg, Sweden 2025

Identification of Loads Based on Historical Data
A Machine Learning Approach to Segmenting and Forecasting Electricity Demand in Response to Price Signals

Meixi Lin
Pirapon Supakkeittikul

Supervisor: Peter Damaschke, Department of Computer Science and Engineering
Advisor: Mihail Chigrichenk, Gothenburg Energy
Examiner: Marina Axelson-Fisk, Department of Mathematical Sciences

Master's Thesis 2025
Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Gothenburg, Sweden 2025

Identification of Loads Based on Historical Data
A Machine Learning Approach to Segmenting and Forecasting Electricity Demand in Response to Price Signals

Meixi Lin
Pirapon Supakkeittikul

Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg

Abstract

To tackle the forthcoming electricity grid tension in Gothenburg, Sweden, this thesis introduces a two-step machine learning approach for managing electricity demand. Initially, consumers are grouped using K-Means clustering according to their past patterns of usage in order to identify categories with the most fluctuating behaviour. Subsequently, several Long Short-Term Memory (LSTMs)—Vanilla, Stacked, Bidirectional, and CNN-LSTM—are trained for predicting electricity demand for such high-impact consumer groups in response to real-time, varying price signals. These models are evaluated using mean absolute error (MAE), root mean square error (RMSE), and loss measures. Among these examined architectures, CNN-LSTM exhibits the most consistent and stable performance across test and prediction datasets. This approach minimises the data and computation needed for deep learning but allows for more customised forecasting. The proposed solution provides a resource-efficient and scalable solution for energy suppliers who wish to monitor changes in demand in response to price changes.

Keywords: Electricity demand forecasting, customer segmentation, K-Means clustering, LSTM, price signals, demand response.

Acknowledgements

We would like to express our profound gratitude to Prof. Peter Damaschke, our supervisor, for his timely comments and helpful guidance throughout this thesis. He helped us stay focused and effectively overcome obstacles during the project with his timely responses and practical advice.

Additionally, we would like to thank Mihail Chigrichenk of Gothenburg Energy, our company advisor, for providing us with useful background information, data, and industry support.

We would especially like to thank our examiner, Marina Axelson-Fisk, for her comments and support in maintaining our high academic standards.

Finally, we would like to thank our families and friends for their support and understanding throughout this project.

Meixi Lin and Pirapon Supakkeittikul, Gothenburg, 2025-08-28

Contents

List of Figures

List of Tables

1

Introduction

1.1 Background

According to Sweden's Updated National Energy and Climate Plan 2021-2030, Sweden will be at risk of electricity shortages [3], especially in major cities such as Stockholm and Uppsala due to the insufficiency of transmission grid capacity and generation capability in the cities. The Swedish electricity market is divided into four areas from north to south, called bidding zones. The electricity is typically transmitted from north to south when demand exceeds the capacity of the transmission grid, leading to the occurrence of a bottleneck. To reduce strain in the grid, price signals were introduced. Price signals help balance the supply and demand of electricity under the limited transmission capacity [4].

Gothenburg is one of the biggest industrial cities in Sweden. The city is transitioning from fossil fuels to clean energy. More electricity is needed to support urban development. Gothenburg Energy imported 89% of its electricity. Svenska kraftnät, the owner of Sweden's transmission system, is expanding the capacity of the transmission grid. Unfortunately, it is not expected to be finished before 2035. Thus, Gothenburg anticipates an electricity supply challenge between 2027 and 2035 [5]. Therefore, price signals will be effective by January 1, 2027 so that consumers can cooperate and contribute to reducing grid strain during peak hours [6].

Gothenburg Energy started to study its customers' behaviour. In collaboration with RISE (Research Institutes of Sweden), Gothenburg Energy gained valuable insights into consumer consumption patterns. The next step involves forecasting the changes in demand and load diversity upon the introduction of the price signals in 2027 [6].

To address the anticipated electricity grid capacity challenge in Gothenburg from 2027 to 2035, we employ a two-step approach: K-Means clustering for customer segmentation and LSTM (Long Short-Term Memory) for demand forecasting. K-Means is an unsupervised machine learning algorithm that groups data points into clusters based on their similarities. Electricity usage varies across different consumers [7].

By applying K-Means clustering, we can identify consumer segments based on usage patterns and detect groups that are more sensitive to price signals. LSTM is a type of recurrent neural network (RNN) that is particularly good at handling time-series data. It improves traditional RNNs by addressing the long-term dependency problem, allowing it to remember important patterns over extended periods. LSTM is well-suited for this task because it can learn patterns in electricity demand over time. Also, it handles long-term dependencies, making it effective for forecasting future demand.

1.2 Related Work

1.2.1 K-Means Clustering for Customer Segmentation

K-Means is widely used as a method for clustering electricity usage patterns [8, 9, 10] due to its simplicity. Hossain [8] used K-Means to cluster different usage patterns of customers to a) showing strong correlation of adjacent moments and b) weak correlation of adjacent time intervals before forecasting customers' electricity usage. Kong et al. [9] also used K-Means to analyse the different patterns of electricity usage among different areas in Qinghai, China. Given the large and unstructured nature of the dataset, K-Means clustering was applied to group usage patterns beforehand, which facilitated subsequent energy consumption structure analysis. Wang et al. [10] proposed 2-step K-Means to cluster electricity usage for electricity grid service estimation. They clustered by energy consumption level and then load profile characteristics.

1.2.2 LSTM for Time-Series Forecasting

The LSTM-based algorithm is extensively used to predict the electricity demand because it is suitable for time-series data [11, 12, 13, 14, 15, 16, 17, 18, 19, 20]. Kisku and Dash [11] compared the performance of LSTM, fuzzy logic, and random forest in peak demand forecasting. LSTM outperformed the other methods in peak demand shaving. Kim et al. [12] conducted a comparative analysis of electricity demand forecasting using LSTM, GRU, and Seq2Seq because these algorithms are good at time-series prediction. As a result, Seq2Seq surpasses LSTM and GRU with approximately 30% better performance. Pati and Mistry [13] compared the performance of LSTM and SVR from other research to predict the electricity demand. The result was that LSTM was better than SVR. However, the LSTM and SVR performance were not evaluated on the same dataset. Simani et al. [14] tried using two variations of LSTM to predict load profile: a) aggregated all load profiles before forecasting and b) forecast the individual load profile of the aggregated result. They found that the aggregated forecasting approach performed better in terms of percentage error. Atef and Eltawil [21] analysed the performance among deep-stacked unidirectional LSTM, deep-stacked bidirectional LSTM, unidirectional LSTM, and bidirectional LSTM on electricity load forecasting to see if the deep-stacked model architecture could enhance the performance of LSTM. However, the result revealed that the deep-stacked LSTM model costs twice the time of the vanilla LSTM but shows no significant improvement in performance. Silva and Meneses [22] developed machine learning models to predict both electricity price and demand. They investigated five LSTM architectures: Vanilla LSTM, Stacked LSTM, Bidirectional LSTM, CNN-LSTM, and attention-based LSTM. For price forecasting, the stacked LSTM outperforms other models, followed by the vanilla LSTM, and the vanilla LSTM, followed by the stacked LSTM, outperforms other models for load forecasting. More advanced LSTM-based research was conducted to improve the accuracy of electricity demand forecasting. Li et al. [15] found that the combination of LSTM-SVR was far better than the single LSTM. Ye et al. [16] compared bidirectional LSTM with one-direction LSTM, and it was 41.7% more accurate. CNN-LSTM is widely used among hybrid LSTM. Babu et al. [17] found that the A-CNN-LSTM performed best compared to single LSTM and CNN in forecasting electricity usage. Alhussein et al. [18] predict individual household electricity load with CNN-LSTM and LSTM. They experimented with the variation of look back and look forward configurations. The outcome was CNN-LSTM got an improvement ranging from 1.21% to 8.03%. Agga et al. [19] implemented a load forecasting model using CNN-LSTM with LSTM as a baseline. With data from the past two weeks, CNN-LSTM performed better. On the other hand, with the last three weeks of data, LSTM worked better. The recent case study of electricity and heating usage in six Swedish schools found that CNN-LSTM achieves higher accuracy than other models [20].

1.3 Research questions and objectives

This thesis focuses on two main questions:

- **Understanding Load Diversity**: How can we segment customers into groups with similar usage patterns? This insight will help Gothenburg Energy to proactively approach the customer group that tends to strain the grid during peak hours.

- **Predicting Electricity Demand**: How can electricity demand be predicted under changing price signals? This work involves understanding how customers respond to the change in price and building models to forecast future demand.

To answer these questions, the customers' usage will be clustered using K-Means clustering on the Spark engine. After that, their usage will be predicted with variations of LSTMs, which are vanilla LSTM, stacked LSTM, bidirectional LSTM, and CNN-LSTM. The goal of this study is to develop a tool that identifies customers exhibiting similar usage behaviour through K-Means clustering, followed by predicting their electricity demand, taking price into account as one influencing factor.

2

Theory

2.1 K-Means

K-Means is a widely used unsupervised learning algorithm for clustering tasks. The objective of the algorithm is to divide a dataset into k distinct, non-overlapping clusters, where each data point belongs to the cluster with the nearest mean, also known as the centroid.

Formally, given a dataset $X = \{x_1, x_2, \ldots, x_n\}$ with n data points, K-Means aims to minimize the following objective function:

$$J = \sum_{i=1}^{k} \sum_{x \in C_i} \|x - \mu_i\|^2$$

where C_i denotes the set of data points assigned to cluster i, and μ_i is the centroid of cluster i.

The algorithm proceeds iteratively with the following steps[23]:

1. Initialize k centroids $\{\mu_1, \mu_2, \ldots, \mu_k\}$, typically chosen at random.

2. Assign each data point to the nearest centroid: For each data point x, compute its distance to all centroids and assign it to the cluster with the closest centroid. Formally, this means:

$$C_i = \{x \in X \mid \|x - \mu_i\|^2 \leq \|x - \mu_j\|^2 \quad \forall j \in \{1, \ldots, k\}\}$$

In other words, each point is grouped into the cluster of the centroid that is nearest in terms of Euclidean distance.

3. Update each centroid based on the mean of the points assigned to it:

$$\mu_i = \frac{1}{|C_i|} \sum_{x \in C_i} x$$

4. Repeat steps 2 and 3 until convergence.

K-Means is simple and fast, but it works best when the clusters are roughly equal in size and shape. It might not perform well on datasets with complex or unevenly distributed clusters.

2.2 CNNs

CNNs are a type of deep learning model especially good at handling data that has a grid-like structure, such as images or time series. CNNs use convolutional layers to automatically learn features from the input by applying filters that move across the data. These filters help detect patterns like edges, textures, or more abstract features as the network depth increases.

Mathematically, for a 2D input image I and a filter K of size $m \times n$, the convolution operation can be expressed as:

$$S(i,j) = (I * K)(i,j) = \sum_{u=0}^{m-1} \sum_{v=0}^{n-1} I(i+u, j+v) \cdot K(u,v)$$

Here, $S(i,j)$ is the output feature map, and the summation represents the weighted sum of the local region covered by the kernel at position (i,j).

After convolutional layers, CNNs typically use a non-linear activation function such as the ReLU[24]:

$$\mathrm{ReLU}(x) = \max(0, x)$$

To reduce spatial dimensions and control overfitting, CNNs often incorporate pooling layers. A common choice is max pooling, which selects the maximum value within a local window. For a window size of $p \times p$, max pooling is defined as:

$$P(i,j) = \max_{0 \le u < p,\, 0 \le v < p} S(i+u, j+v)$$

Finally, the extracted features are usually passed through one or more fully connected layers to perform classification or regression, depending on the task.

CNNs often include pooling layers to reduce the size of the data and make the model more efficient. They are widely used in computer vision tasks like image classification, object detection, and also in some sequence-based problems.

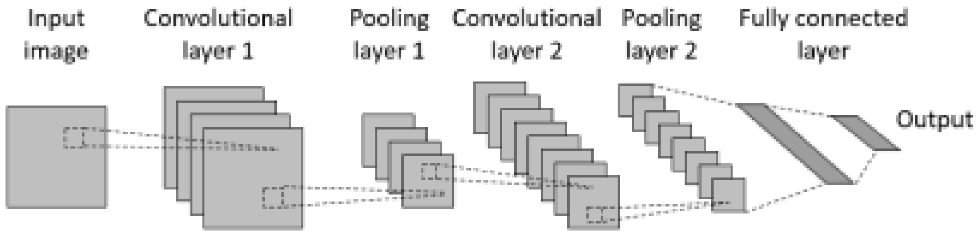


Figure 2.1: CNN architecture according to [1]

2.3 LSTM

LSTM networks are a type of RNN designed to handle sequence data such as time series, speech, or text. Compared to other types of neural networks, LSTMs are well-suited for tasks where system outputs depend on historical sequences rather than just the current input. Important properties considered in this context include the ability to handle nonlinear and time-dependent behaviour, process multidimensional inputs, support multi-step ahead predictions, and manage error propagation over time.

LSTM is a type of RNN designed to handle sequence data, such as time series or text. Unlike traditional RNNs, LSTMs can remember long-term dependencies in the data, thanks to their special architecture that includes memory cells and three gates: the input gate, forget gate, and output gate. These gates control the flow of information, allowing the network to keep or discard information as needed. LSTMs are particularly useful when the current output depends not just on the recent input, but on something that happened a long time ago in the sequence [25]. In the next section, we examine state-of-the-art neural networks built upon the LSTM gating mechanism, Bidirectional LSTM and CNN-LSTM.

2.3.1 Bidirectional LSTM

Bidirectional LSTM networks extend the standard LSTM architecture by propagating the cell state not only forward in time but also backward. This bidirectional processing allows the network to consider dependencies from both past and future time steps, enhancing its ability to model temporal relationships. As a result, Bidirectional LSTMs can capture more complex time-dependent patterns and resolve them with greater precision compared to unidirectional LSTMs [25].

Research [2] has demonstrated that bidirectional LSTMs can effectively handle spatially and temporally distributed information, even in the presence of missing data. This is achieved through a dynamic connection mechanism that adapts to data gaps

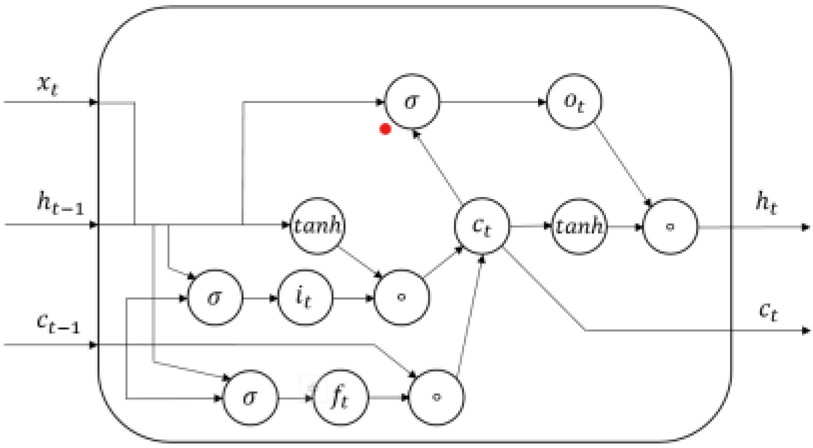


Figure 2.2: LSTM architecture according to [1]

by modifying the links between memory cells. Furthermore, studies such as [26] show that bidirectional architectures are well-suited for multidimensional problems, where features from various dimensions are processed in parallel and then integrated using a bidirectional structure. Given these capabilities, Bidirectional LSTMs show

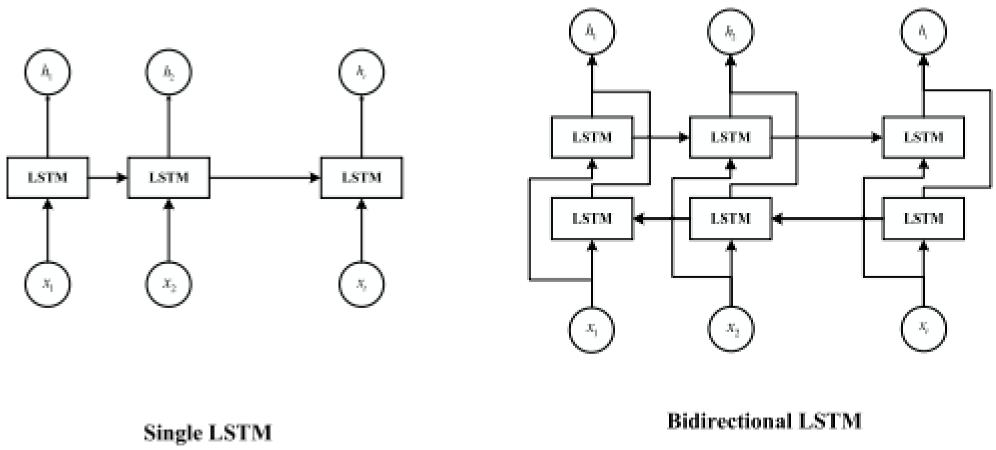


Figure 2.3: LSTM architecture according to [2]

potential for representing spatial-temporal dynamics in manufacturing processes.

2.3.2 CNN-LSTM

To enhance the prediction of long-term dependencies, input data collected over extended periods can be filtered and reduced through convolutional operations integrated into LSTM architectures. These convolutional enhancements either modify the overall network structure or are directly embedded within the LSTM cell itself. The goal is to improve sequence processing by transforming high-dimensional input data into more compact, informative feature representations [25].

One such approach, as proposed in [1], extends the standard LSTM cell by embedding convolution operations. In this model, current input sequences, recurrent outputs, and weight matrices are convolved to extract relevant temporal and spatial correlations. The resulting features are then used as inputs to the LSTM gates, providing a more focused and efficient representation of the information, which improves the effectiveness of the cell state updates.

Further developments are discussed in [27] and [28], where convolutional layers are combined with LSTM networks in stacked architectures. These configurations enable the modeling of locally distributed patterns and temporal dependencies simultaneously. The convolutional components help extract spatial features, while the LSTM layers learn how those features evolve over time. This structure supports better long-range forecasting by allowing a broader time window to be considered in the predictions.

Moreover, convolutional LSTM networks can effectively model multiple quantities, such as spatially and temporally distributed relationships. However, when multiple outputs are required in their original units rather than in a compressed feature form, additional decoding or deconvolution layers are necessary to reconstruct the full-resolution predictions [25].

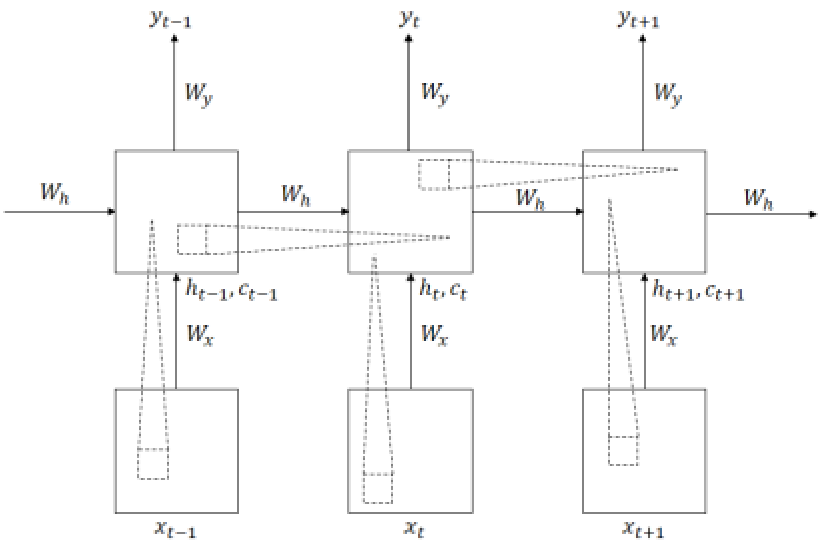


Figure 2.4: Convolution operations within LSTM cells according to the approach of [1]

3

Methods

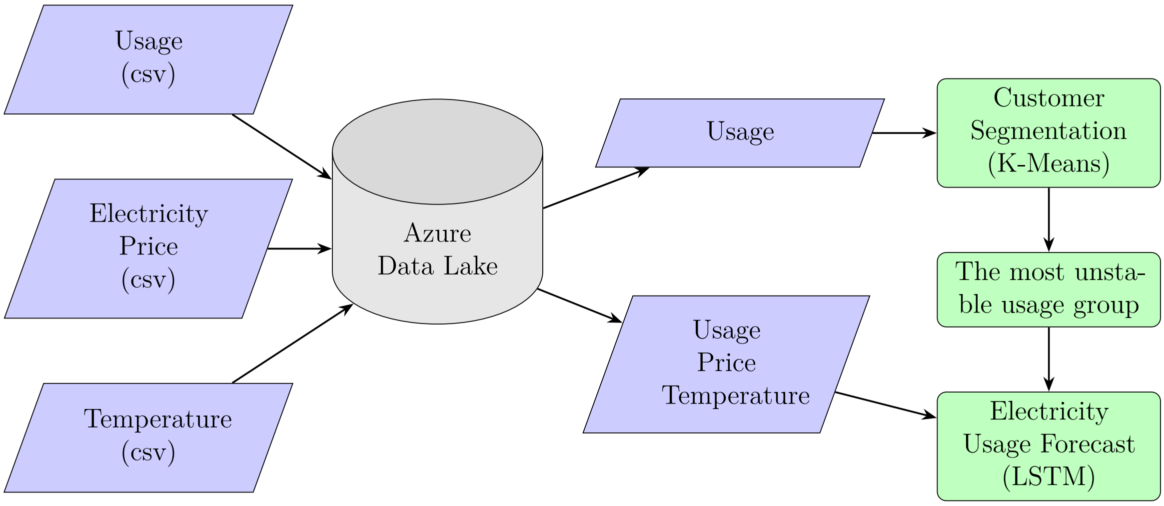


Figure 3.1: Methodology overview

The data used in this thesis were usage data, electricity price, and temperature data. The usage data and electricity price were provided by Gothenburg Energy. The temperature data was downloaded from SMHI. All of them were in 1-hour intervals. The data were in CSV and XLSX formats, which were imported into Microsoft Azure Data Lake. The code was developed in a Jupyter notebook on the Microsoft Fabric, which operates on a fully managed Apache Spark compute platform [29]. The K-Means model was trained using the Spark engine to accelerate the processing time. However, Spark does not support TensorFlow, which is the selected library to train the LSTM models. The LSTM model will be trained on Python runtime. The overall process is depicted in 3.1. The data from December 2022 to February 2023, with 363 unique customers, were selected to train the K-Means models. This period was chosen since it covers the peak winter of the most recent available data. Then the customer group with the highest fluctuation in usage was taken to predict the peak. Four variations of LSTM were examined using customer usage data. The best variation of LSTM was then used to train the whole most unstable usage group.

3.1 Data Ingestion

First, Spark DataFrames were created to load electricity usage, temperature, and price data. They were then stored in Azure Data Lake. The usage and price were already in standard time (UTC+1), but the temperature was in clock time, which will be UTC+2 in summer time [30].
To keep the time consistent, it was converted to the same format as the usage and price. After that, the data were preprocessed differently for K-Means and LSTM.

3.2 Data Preparation for K-Means

The K-Means clustering process will rely on usage data. The cleaning started with deduplication and then filtered out unwanted dates and times. Since the focus is only during peak hours in the winter, the period considered is from December to February, excluding the grid holidays. The missing hours were imputed using linear interpolation. The data at the edges that cannot be interpolated were imputed with 0.

Table 3.1: List of Grid Holidays

Date	**Holiday**
24 December	Christmas Eve
25 December	Christmas Day
26 December	Second Day of Christmas
31 December	New Year's Eve
1 January	New Year's Day
6 January	Epiphany Day

Next, the usage data will be pivoted to the columnar structure shown in Figure 3.2. The usage for the same hour will be encoded in a single list. Each column will store the list according to its corresponding hour. Then, all hours will be wrapped into a dense vector before being converted to a feature vector for K-Means training.

anonymised_ID	**hour_7**	**hour_8**	**hour...**
12345	[10.1, 10.6, 10, 9, 11,]	[15.1, 15.6, 15, 14.9, 13,]	[...]

$\downarrow$ flatten

Flattened Feature Vector: [10.1, 10.6, 10, 9, 11, 15.1, 15.6, 15, 14.9, 13, ...]

Figure 3.2: Flattened feature vector for K-Means example

3.3 K-Means Clustering

K-Means clustering was run on the Spark engine using PySpark's K-Means. The elbow method was used as the primary method to obtain a candidate range of k values, and then the Silhouette score was calculated to select the optimal k based on the most obvious elbow and the maximum Silhouette Score. The results will be stored in the data lake for further use in data analysis, such as year-to-year comparisons in Microsoft Power BI.

3.4 Data Preparation for LSTM

For LSTM, the data preparation steps are different from K-Means. The usage, price, and temperature data will be used. The data was taken from November 2021 to February 2023 without filtering any data out, like in K-Means, to preserve the continuity of usage values. Starting with usage data, deduplication was done before outlier removal and imputation. Two-step outlier removal was conducted to smooth usage data, using Isolation Forest and IQR (Fig.3.4). For temperature data, the outlier removal was done using Isolation Forest only. Subsequently, usage, temperature, and price were imputed using the same method as for K-Means. When all of the data were cleaned and imputed, they were joined into a single DataFrame. Then, Z-score normalisation was applied. Moreover, frequency features were added to enhance the dataset with time-related information. Date, day of the week, month, and year were transformed into sine and cosine waves as frequency features. The train/validation/test sets were split by a 70:10:20 ratio from November 2021 to November 2023. Next, each dataset—training, validation, and testing—was transformed into a tensor with short- and long-term input in one window to capture both short- and long-term changes in the data over the period. The plan was to predict the next 90 days based on the previous 270 days. Therefore, there will be 6480 hours (270 days) for the long input and 2160 hours for the prediction. But, since the memory and compute resources are limited, the maximum possible hour for a long input is 720 hours (30 days) and 168 hours (7 days) for the prediction. The short input covers 24 hours to capture daily changes. The window structure is depicted in Fig.3.3

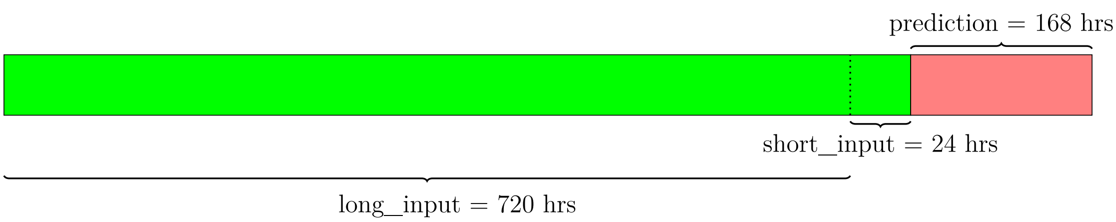


Figure 3.3: A single window structure

3.5 LSTM models

Four variations of LSTM were compared in this step: Vanilla, Stacked, CNN-LSTM, and Bidirectional. Each variation will be trained using the same set of hyperparameters. The models' architectures were kept the same as much as possible.

Figure 3.4: Boxplot on different techniques of outlier detection

4

Results and Evaluation

4.1 Evaluation Metrics

This section presents the results obtained from the K-Means clustering and LSTM-based prediction models. The evaluation is divided into two main parts: clustering analysis for identifying usage patterns and time series forecasting to predict future electricity consumption. The performance of both models is assessed using relevant evaluation metrics and visual inspection to verify the effectiveness of the proposed methodology.

In the first part, seasonal usage data is grouped using K-Means clustering. The objective is to classify similar consumption behaviours within the desired period. The elbow method and Silhouette score are used to determine and validate the optimal number of clusters.

In the second part, four LSTM variants are developed to predict electricity usage of the most fluctuating usage cluster from K-Means. Based on a comparison of performance metrics, the best LSTM variation was selected for final forecasting. The predicted values are then compared to actual usage to evaluate model accuracy.

4.1.1 K-Means

To determine the optimal number of clusters (k), the elbow method was used. This method helps identify the point where increasing the number of clusters no longer significantly reduces the within-cluster sum of squares, which indicates a good balance. In addition, the Silhouette score was calculated to evaluate the quality of the clustering result for the chosen k. A higher Silhouette score indicates that the objects are well matched to their own clusters and poorly matched to neighbouring clusters, implying better-defined clustering. The bubble chart from standard deviation and variance is also used to visualise the spread and size of clusters; this chart is also easier to understand for a general audience.

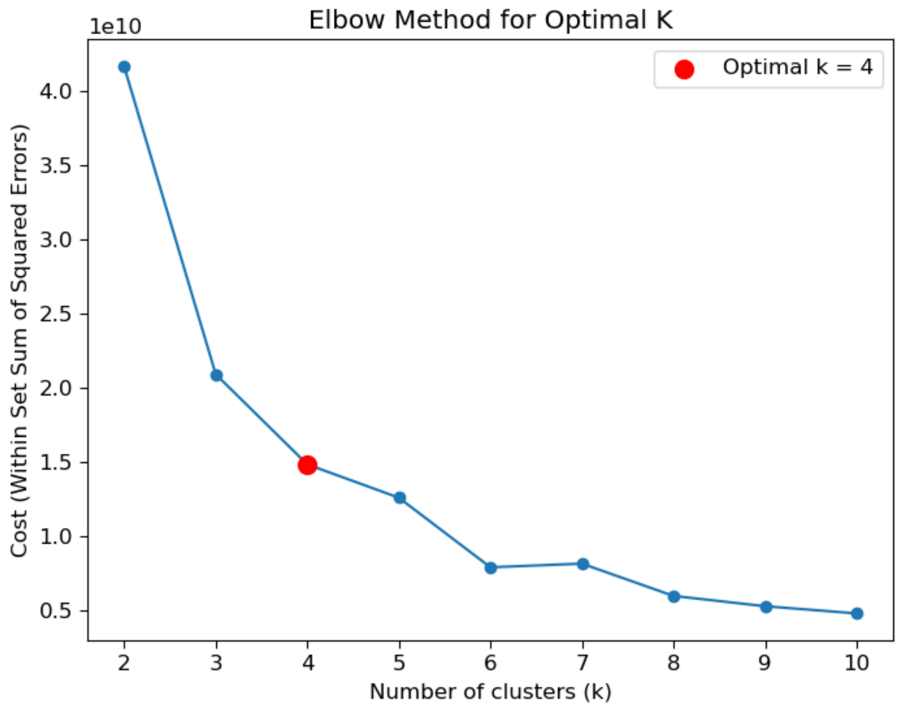


Figure 4.1: Determining optimal k using elbow method

4.1.2 LSTM

The models were evaluated by three metrics: loss, MAE (mean absolute error), and RMSE (root mean square error). Loss was used to determine how close the predicted and actual values are. Then MAE was used to indicate the error of the models on average, and RMSE was used to verify that large prediction errors remained within acceptable levels.

4.2 Results

4.2.1 K-Means

The optimal number of clusters from the elbow method for peak winter (December 2022–February 2023) is four (Fig.4.1). Then the bubble chart was plotted (Fig.4.2) to examine the distribution and size of each cluster. The bubble chart shows that cluster 2 is the most fluctuating group, with the smallest number of users (by bubble size). Since resources were very limited, only cluster 2 was taken to predict the usage with LSTM, as this cluster had the worst usage behaviour.

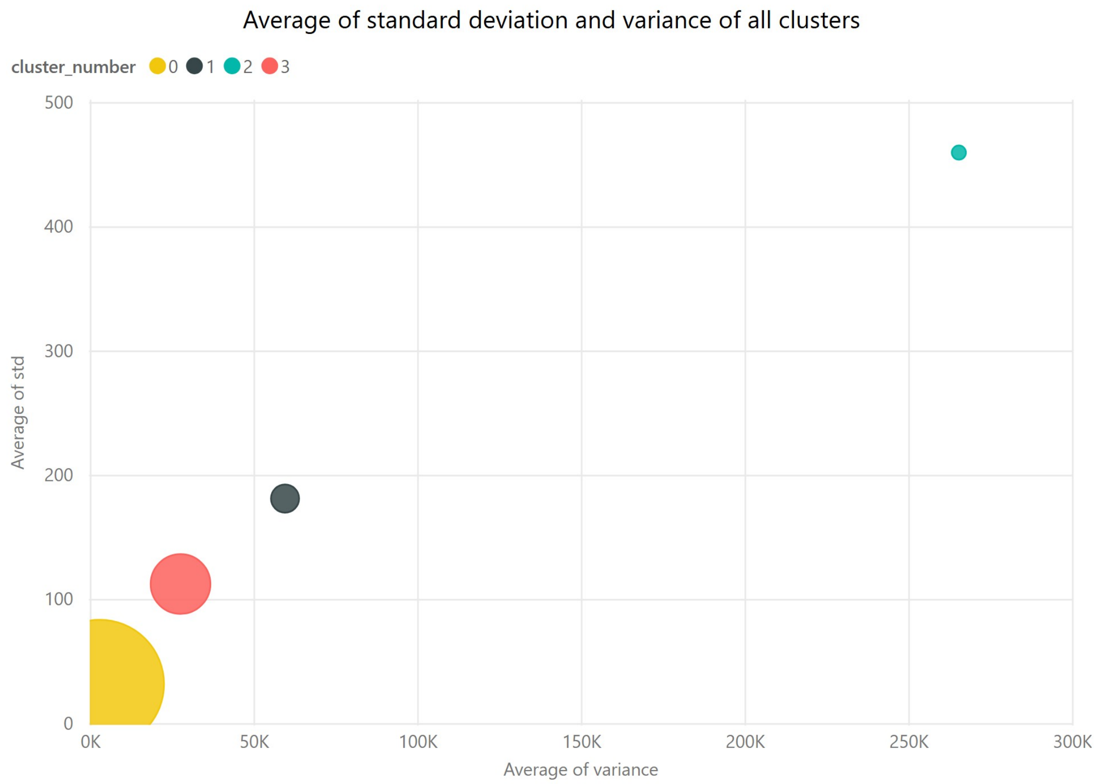


Figure 4.2: Bubble chart to examine the distribution and size of each cluster

4.2.2 LSTM

4.2.3 Model Selection

To determine the most suitable LSTM-based model, four model architectures were trained and evaluated while keeping the model structure as similar as possible (Table 4.1). As shown in Table 4.2, Vanilla LSTM underperforms relative to the others, while Stacked LSTM is more consistent across the test and prediction datasets. Bidirectional LSTM performs well on the test set but overfits on the prediction dataset. Finally, CNN-LSTM has the most balanced and consistent performance, with the lowest or near-lowest error across all metrics. The plots for each variation on the test and prediction dataset are shown in Appendix 1.

Table 4.1: Comparison of LSTM Model Architectures and Hyperparameters

Model	**LSTM Layers**	**Hidden Size**	**Convolutional Layers**	**Directionality**
Vanilla LSTM	1	64	None	Unidirectional
Stacked LSTM	2	64	None	Unidirectional
CNN-LSTM	1	64	1 Conv1D + MaxPool	Unidirectional
Bidirectional LSTM	1	64	None	Bidirectional

Table 4.2: Evaluation metrics of different LSTM-based models on test and prediction datasets

Metrics	Vanilla LSTM		Stacked LSTM		Bidirectional LSTM		CNN-LSTM	
	Test Dataset	Prediction Dataset	Test Dataset	Prediction Dataset	Test Dataset	Prediction Dataset	Test Dataset	Prediction Dataset
Loss	0.4103	0.7824	0.1931	0.7413	0.1821	1.1656	0.1773	0.8691
MAE	0.5318	0.7277	0.3502	0.6948	0.3454	0.7974	0.3109	0.7342
RMSE	0.6405	0.8845	0.4394	0.8610	0.4268	1.0796	0.4211	0.9323

4.2.4 CNN-LSTM Model Training

The same model structure from the previous step was used for hyperparameter tuning. The number of convolutional layers, learning rate, batch size, and early stopping are the four hyperparameters that will be tuned.

The data ranged from "2021-11-01" to "2022-12-01" and was divided into 70:10:20 train/validation/test splits. The next 168 hours were taken as a prediction dataset to evaluate each variation. All customer usage in the cluster were used to train the model, and a single user was sampling for the prediction dataset.

The first trial is to add an extra convolutional layer to the original model. The result indicates that the original hyperparameter set performs better. The hyperparameter set is reverted to the original, and the learning rate is reduced to 0.00001 for the second attempt. The result was the same as the first one; the original hyperparameter set performed better. Then, the batch size is increased to 64, and the result is still the same, so the original hyperparameter set was taken to compare against a version with increased early stopping, and the result is that the original hyperparameter set performed better. Tables 4.3-4.6 show the comparisons of loss, MAE, and RMSE for each try.

Table 4.3: Evaluation metrics of the first trial

Metrics	Original Hyperparameter Set		Added a Convolutional Layer	
	Test Dataset	Prediction Dataset	Test Dataset	Prediction Dataset
Loss	0.4952	0.8411	0.4994	0.9957
MAE	0.4393	0.7513	0.4486	0.7986
RMSE	0.7037	0.9171	0.7067	0.9979

Table 4.4: Evaluation metrics of the second trial

Metrics	Original Hyperparameter Set		Reducing Learning Rate	
	Test Dataset	Prediction Dataset	Test Dataset	Prediction Dataset
Loss	0.4952	0.8411	0.3736	0.9285
MAE	0.4393	0.7513	0.3949	0.7521
RMSE	0.7037	0.9171	0.6113	0.9636

Table 4.5: Evaluation metrics of the third trial

Metrics	Original Hyperparameter Set		Increased Batch Size (64)	
	Test Dataset	Prediction Dataset	Test Dataset	Prediction Dataset
Loss	0.4952	0.8411	0.5492	0.9988
MAE	0.4393	0.7513	0.4697	0.8105
RMSE	0.7037	0.9171	0.7411	0.9994

Table 4.6: Evaluation metrics of the fourth trial

Metrics	Original Hyperparameter Set		Increasing Early Stopping	
	Test Dataset	Prediction Dataset	Test Dataset	Prediction Dataset
Loss	0.4952	0.8411	0.53	1.2429
MAE	0.4393	0.7513	0.4586	0.8704
RMSE	0.7037	0.9171	0.728	1.1148

Then, the whole peak winter period was predicted with the original hyperparameter set. Fig.4.3 shows the comparison between actual and prediction for the whole period, and Fig.4.4 shows the first 168 hours' comparison.

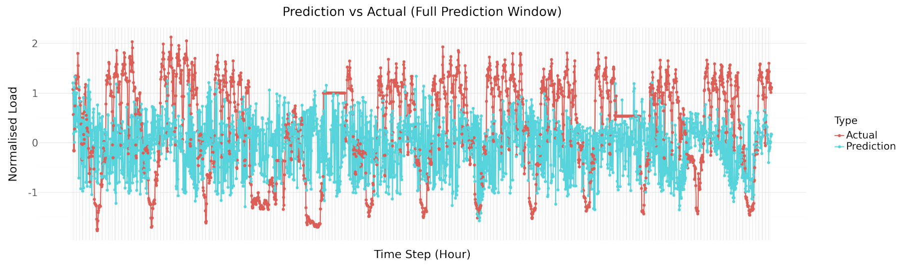


Figure 4.3: Prediction and actual usage during peak winter (December 2022–February 2023)

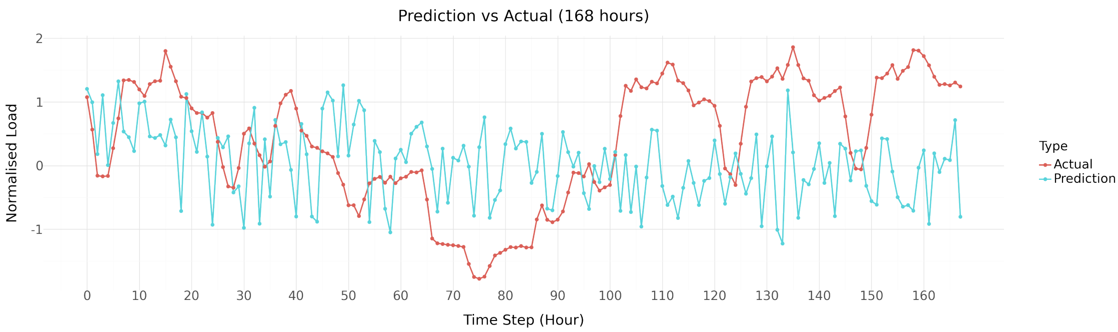


Figure 4.4: Prediction and actual usage of the first week (168 hours) of peak winter (1-7 December 2022)

5

Discussion

5.1 Interpretability of Clustering Results

The application of K-Means clustering to the user consumption data resulted in the identification of four distinct customer segments. These clusters were differentiated not only by average consumption levels but also by temporal variability and behavioural consistency. This segmentation proved instrumental in enhancing the interpretability of the dataset and improving the subsequent forecasting performance.

By grouping users with similar consumption characteristics, the clustering approach enabled the development of specialized LSTM models tailored to each cluster's dynamics. This targeted modeling strategy outperformed a single global model applied to the unsegmented dataset, as the LSTM models were better able to learn intra-cluster temporal dependencies and recurring usage patterns.

From Fig.4.2, Cluster 0 represents customers with low usage, characterized by the smallest variance and standard deviation. However, Cluster 2 exhibited the highest variability in both variance and standard deviation, indicating irregular usage patterns that significantly reduced forecast accuracy. Cluster 1 shows moderately high variability, reflecting irregular but fewer fluctuations compared to Cluster 2, while Cluster 3 represents intermediate variability, falling between the stable Cluster 0 and Cluster 1.

The choice of K-Means clustering, despite its inherent limitations, such as sensitivity to initial centroids and the assumption of spherical cluster shapes, was justified in this context. The relatively low dimensionality of the input features and the need for computational scalability made K-Means a suitable candidate. Its fast convergence and interpretability further supported its use, especially when cluster boundaries and centroids could be meaningfully interpreted in relation to real-world user behaviour.

However, it is important to acknowledge that K-Means does not necessarily produce the most semantically meaningful segmentation, particularly in datasets with overlapping or non-globular patterns. Future work could explore more advanced clustering techniques, such as Gaussian Mixture Models or density-based methods, to potentially capture more nuanced behavioural structures. Additionally, incorporating domain-specific features or metadata could further enrich the clustering process and enhance both model interpretability and performance.

Overall, the clustering step played a crucial role in uncovering latent structure in the consumption data, facilitating not only more accurate predictions but also a deeper understanding of user behaviour patterns.

5.2 LSTM Performance and Generalisation

The LSTM models worked well in clusters where users had clear and regular patterns of electricity use. This shows that deep learning, especially LSTMs, can be very effective for forecasting time series data like energy consumption. By learning from past data, LSTMs were able to capture trends and seasonal changes that simpler models might not detect.

However, getting the models to perform well was not straightforward. It required careful tuning of key hyperparameters—such as sequence length (how much past data to look at), batch size (how much data to process at once), and learning rate (how quickly the model updates during training). These settings had a big impact on how fast the model learned and how well it performed on new, unseen data. This required substantial time and effort in experimentation.

To avoid overfitting, where the model learns the training data too well but fails on new data, we used techniques like early stopping and dropout layers. Early stopping helps stop the training when the model's performance stops improving, and dropout randomly ignores some connections during training to make the model more general and less likely to overfit. These techniques were especially helpful for smaller clusters where there was less data to train on.

The main limitation in our setup was the environment in which we trained the models. Since all data are on Microsoft Fabric, we need to train our model on Microsoft Fabric, which has no GPU support since it is designed to work with big data with the Spark engine. Hence, we cannot create a longer sequence of training data, which meant being unable to create a window with many years of input. Therefore, the LSTM model could not capture the temporal features as much as it would. Another limitation was the small number of extra input features, also known as exogenous variables. We only used temperature and electricity price, which are important but not enough to fully explain why people use electricity the way they do. Other factors—like whether people are home (occupancy), what kinds of appliances they use, or whether it's a holiday—could have added useful context and helped the model make better predictions.

In future work, including a wider range of features could make the forecasts more accurate and reliable. Also, exploring different model structures or combining LSTMs with other methods could help in capturing more complex behaviours, especially in clusters with irregular or unpredictable usage.

5.3 Practical Utility for Energy Providers

The results of this project can be useful for Gothenburg Energy. By using clustering to group customers based on their electricity usage patterns, energy providers can develop more targeted and efficient strategies. For example, in our clustered forecasts, Cluster 2 users (high variability) were found to deviate strongly during peak hours. This suggests that they could be prioritized for targeted demand response incentives, while Cluster 0 (stable users) might be better suited for fixed pricing plans.

Forecasting electricity demand at the cluster level also helps with managing total electricity loads more effectively. This is especially important for local energy networks and microgrids, where balancing supply and demand in real time is critical.

However, using this type of system in the real world is not without its challenges. People's energy usage habits can change over time due to many factors, such as changes in weather, household routines, or even the adoption of new technologies. To keep the forecasts accurate, the models would need to be updated regularly. This would involve setting up systems to monitor model performance, detect when the model becomes less accurate—which is known as concept drift—and retrain the models using new data. These kinds of features were not part of this project but are important for any real-world application.

Despite these challenges, the results of this study show that combining clustering with LSTM forecasting has strong potential for use in smart energy systems. With further development, such models could help Gothenburg Energy make better decisions, reduce costs, and support the transition to more efficient and sustainable power grids.

5.4 Limitations

In practice, model drift happens when a machine learning model becomes less accurate over time because the data it was trained on no longer represents the real world. In our case, this risk is particularly relevant once dynamic pricing signals are introduced. For example, during our summer test period, it might be that several households in Cluster 2 reduced their evening electricity consumption when the spot price was unusually high. This would represent a form of concept drift, as the relationship between input features (price, time of day) and the target (consumption) could shift compared to the training set.

Basically, there are three main types of model drift: concept drift, data drift, and upstream data change. Data drift happens when the overall distribution of input data changes, making the model less effective. Upstream data changes occur when the format or structure of the data changes, such as switching from one currency to another or from Celsius to Fahrenheit, which can cause errors if the model is not adjusted accordingly [31].

Data drift could occur if adoption of rooftop solar panels or batteries increases. In such cases, the overall distribution of grid consumption decreases, even if household activity remains the same. Additionally, if customers attempt to avoid using electricity during high-price periods and instead opt for lower-cost alternatives, this can lead to a change in the distribution of usage, which in turn can result in a data drift. Both concept drift and data drift cause model drift, which leads to inaccuracy in the model's prediction.

In our proposed solution, K-Means clustering would first be affected if the customers' behaviours changed, and then LSTM would be degraded by new usage distributions. We would resolve these issues by assessing the performance of models periodically; if there is any declining trend in performance, the models should be retrained.

5.5 Risk Analysis and Ethical Considerations

This section outlines the potential risks associated with the project and discusses how ethical concerns, particularly related to data privacy, have been addressed. As this study involves large-scale electricity usage data and sensitive behavioural patterns, it is essential to acknowledge the technical and ethical implications of the methodology.

5.5.1 Risk analysis

5.5.1.1 Computational cost

Training deep learning models such as LSTMs on large volumes of historical usage data presents a significant computational challenge. The size of the dataset requires substantial memory resources, especially when handling long input sequences and multiple customer groups.

During the data cleaning phase, only a portion of the files is processed at a time to avoid overwhelming system memory. This step is repeated until the entire dataset is cleaned and preprocessed. The cleaned dataset is then stored in a Microsoft Fabric table to support efficient access and scalability.

To handle the computational demands of model training, the project makes use of the PySpark Engine, a distributed computing tool built into the Microsoft Fabric ecosystem. PySpark allows data to be processed in parallel across multiple nodes, enabling the full dataset to be loaded and used for training without memory overload. This infrastructure ensures that model training remains efficient and scalable, even as data volume increases.

5.5.1.2 Correctness of data

Another critical risk concerns the correctness and reliability of the raw data. Since the electricity usage data was automatically collected from smart meters and sensor systems, it is susceptible to various types of errors. These include duplicated records, missing values, outliers, and sensor glitches often caused by networking issues or device malfunctions.

To mitigate this risk, rigorous data cleaning was conducted before any modeling was performed. The data cleaning process included duplicate removal, outlier detection, and basic sanity checks. Special care was taken to ensure that only accurate and consistent data was used for training, as poor data quality would directly impact the performance and trustworthiness of the models.

Following the cleaning phase, an additional round of Exploratory Data Analysis (EDA) was conducted. This involved randomly sampling data subsets to manually inspect and verify the quality of the cleaned dataset. This iterative approach helped ensure that the models were trained on reliable data.

5.5.2 Ethical considerations

Given that energy usage data can reflect sensitive information about household habits and behaviours, privacy and ethical concerns are necessary. Gothenburg Energy, the data provider for this study, took steps to protect customer identities by assigning each user an anonymised ID. Each user was assigned a 10-character anonymised ID generated via a hashing function. No names, addresses, or meter numbers were stored in the dataset.

All modeling and analysis were carried out using these anonymised IDs. Any insights derived from customer behaviour patterns are intended strictly for internal use by Gothenburg Energy, with the goal of improving energy services and system efficiency.

When sharing results through this thesis, only cluster-level findings are reported, and no customer-specific data or patterns are disclosed. This approach ensures transparency in research while fully respecting the confidentiality of individuals involved in the dataset.

Additionally, ethical considerations extend beyond privacy. Care must also be taken to avoid bias in modeling and ensure fair treatment of all customer groups. Although a fairness analysis was not within the scope of this study, in future work, we plan to evaluate whether clusters with smaller household sizes or lower historical consumption are systematically over- or under-predicted, and adjust the model to reduce such bias.

6

Conclusion and Future Work

6.1 Conclusion

This thesis has explored a practical and data-driven approach to electricity demand forecasting by integrating unsupervised learning techniques with deep learning models. Specifically, it demonstrated how customer segmentation through K-Means clustering can be used to enhance the performance of LSTM-based time-series forecasting models. The segmentation of electricity users based on their historical consumption patterns enabled the development of more specialised forecasting models that are better suited to capturing the distinct behaviours of each group. This approach led to improved prediction accuracy and provided greater interpretability of the underlying consumption dynamics.

The project successfully delivered the following key outcomes:

- A scalable and reusable data processing pipeline for electricity consumption data.
- Behavioural segmentation of customers using unsupervised learning techniques.
- Cluster-specific LSTM models were trained to capture intra-group consumption dynamics.
- A comparative evaluation showing the benefits of clustering prior to forecasting.

The results strongly support the main hypothesis of the study: that customer segmentation can enhance the accuracy of energy demand forecasting. By tailoring models to the unique characteristics of different user groups, utilities can gain more precise insights into consumption trends, allowing for more informed decision-making. For example, energy providers could use this approach to develop targeted demand response programs, offer customized pricing plans, or plan more efficiently for peak load scenarios.

Moreover, the findings highlight the potential of combining traditional data mining techniques with modern machine learning models in the context of smart grids and energy management. Although the current study employed a limited feature set and static clustering, it lays a strong foundation for future work in more dynamic, real-time applications.

In summary, this thesis contributes to the growing body of research focused on intelligent energy forecasting. The integration of clustering and deep learning techniques not only improves forecasting performance but also brings practical value to energy providers. With further development—such as the inclusion of additional data sources, adaptive retraining mechanisms, and deployment in live systems—this framework could support more flexible, efficient, and sustainable energy systems in the future.

6.2 Future Work

Building upon the insights and limitations of this study, several avenues for future research and development are recommended. These directions aim to enhance model performance, improve real-world applicability, and ensure responsible and ethical deployment of forecasting solutions.

Model Enhancement

While LSTM models proved effective for capturing sequential patterns in consumption data, there is room for improvement through the use of more advanced architectures. For instance, CNN-LSTM [32] hybrids could enable the model to better detect local trends and patterns within time windows, while Transformer-based architectures [33] could improve long-range dependency modeling through self-attention mechanisms. These models have demonstrated state-of-the-art performance in various time-series and sequence-modeling tasks and could further boost accuracy and robustness in energy forecasting.

Feature Expansion

The current study was limited to a small set of exogenous variables—namely, temperature and price. Expanding the feature space could significantly improve the model's ability to explain and forecast consumption behaviour. Potential additions include:

- Occupancy data: Reflecting real-time presence or activity levels in households or buildings.

- Calendar features: Including weekday/weekend flags, holiday indicators, and seasonality tags.

- External events: Large public events (e.g., sports matches, festivals) that might affect energy usage patterns.

- Appliance-level data: If available, could allow for disaggregated modeling and more detailed analysis of usage drivers.

Model Retraining and Drift Monitoring

Real-world data is dynamic, and user behaviour may evolve over time due to economic changes, technological adoption, or policy interventions. Therefore, it is crucial to develop pipelines that continuously monitor for concept drift and automatically retrain models when performance degrades. This could involve the following:

- Scheduled retraining, e.g., models could be retrained monthly.
- Performance-based retraining triggered by error thresholds
- Online learning methods that incrementally update the model as new data becomes available.

Deployment and Real-Time Forecasting

For practical adoption, the forecasting system should be deployed in a real-time environment. A real-time dashboard on Microsoft Fabric could integrate cluster-level predictions with live electricity prices and sensor feeds, enabling Gothenburg Energy to issue demand-response signals or provide personalized recommendations to users. Additionally, integration with energy management systems or smart meters would allow providers to act on predictions. For example, by issuing demand response signals or sending personalized recommendations to end-users.

Bibliography

[1] Kai-Fung Chu, Albert YS Lam, and Victor OK Li. Deep multi-scale convolutional lstm network for travel demand and origin-destination predictions. *IEEE Transactions on Intelligent Transportation Systems*, 21(8):3219–3232, 2019.

[2] Zhiyong Cui, Ruimin Ke, Ziyuan Pu, and Yinhai Wang. Deep bidirectional and unidirectional lstm recurrent neural network for network-wide traffic speed prediction. *arXiv preprint arXiv:1801.02143*, 2018.

[3] Ministry of Climate and Enterprise. Sweden's final updated national energy and climate plan (necp) 2021-2030, 2023. URL https://commission.europa.eu/document/download/26d2c93e-641d-489f-a160-a7052fde58bb_en?filename=SE_FINAL%20UPDATED%20NECP%202021-2030%20%28English%29.pdf. Available online.

[4] Sveriges Riksbank. The swedish electricity market: Today and in the future. *Sveriges Riksbank Economic Review*, 2023(1): 5–36, 2023. URL https://www.riksbank.se/globalassets/media/rapporter/pov/artiklar/engelska/2023/230512/2023_1-the-swedish-electricity-market--today-and-in-the-future.pdf.

[5] Göteborg Energi. Kapacitetsutmaningen, 2023. URL https://www.goteborgenergi.se/om-oss/vad-vi-gor/hallbarhet/kapacitetsutmaningen. Available online.

[6] RISE Research Institutes of Sweden. Algorithm aids grid stability and smarter energy consumption, 2023. URL https://www.ri.se/en/our-stories/algorithm-aids-grid-stability-and-smarter-energy-consumption. Available online.

[7] Yozika Arvio, Iriansyah BM Sangadji, Dine Tiara Kusuma, and Zalas Heribertus. K-means clustering analysis to identify electrical load consumption patterns based on primary energy consumption. In *2024 11th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI)*, pages 645–650. IEEE, 2024.

[8] M. S. Hossain. Analysis of electricity usage time series with k-means clustering. Master's thesis, Uppsala University, 2022. URL https://www.diva-portal.org/smash/get/diva2:1662806/FULLTEXT01.pdf.

[9] Weizheng Kong, Yaohua Wang, Hongcai Dai, Liujun Zhao, and Chunming Wang. Analysis of energy consumption structure based on K-means clustering algorithm. In *E3S Web of Conferences*, volume 267, page 01054. EDP Sciences, 2021. doi: 10.1051/e3sconf/202126701054. URL `https://doi.org/10.1051/e3sconf/202126701054`.

[10] Jiyu Wang, Xiangqi Zhu, and Barry Mather. A two-step time-series data clustering method for building-level load profile. In *Proceedings of the 2023 IEEE Power & Energy Society General Meeting*, Orlando, Florida, USA, July 2023. IEEE, National Renewable Energy Laboratory. Preprint, NREL/CP-5D00-84634.

[11] T. Kisku and S. Dash. Optimization and leveraging of peak demand using lstm and fuzzy logic approaches in demand-side management. In *2024 IEEE International Conference on Smart Power Control and Renewable Energy (ICSPCRE)*, pages 1–5, 2024. doi: 10.1109/ICSPCRE62303.2024.10675009.

[12] A. R. Kim, D. Park, C. S. Lee, and H. Chang. A comparative analysis of time series forecasting methods for short-term electricity demand prediction. In *2023 14th International Conference on Information and Communication Technology Convergence (ICTC)*, pages 1235–1238, 2023. doi: 10.1109/ICTC58733.2023.10392314.

[13] U. Pati and K. D. Mistry. A novel demand-side management strategy for residential load with an integrated deep-learning based forecasting engine. In *2022 22nd National Power Systems Conference (NPSC)*, pages 542–547, 2022. doi: 10.1109/NPSC57038.2022.10069201.

[14] K. N. Simani, Y. O. Genga, and Y. C. J. Yen. Using lstm to perform load modelling for residential demand side management. In *2023 31st Southern African Universities Power Engineering Conference (SAUPEC)*, pages 1–5, 2023. doi: 10.1109/SAUPEC57889.2023.10057875.

[15] H. Li, P. Li, Y. Zhang, W. Zu, and Y. Zheng. Research on electricity demand forecasting based on lstm-svr. In *2022 2nd International Conference on Algorithms, High Performance Computing and Artificial Intelligence (AHPCAI)*, pages 233–239, 2022. doi: 10.1109/AHPCAI57455.2022.10087856.

[16] Q. Ye et al. Data-driven approaches predict hourly electricity demand profiles at industry and city-level. In *2023 8th Asia Conference on Power and Electrical Engineering (ACPEE)*, pages 905–910, 2023. doi: 10.1109/ACPEE56931.2023.10135675.

[17] U. R. Babu, N. Kumar, K. Padmaja, V. Bhoopathy, M. S. Alam, and S. R. Devi. Predicting electricity consumption in commercial and residential buildings using a-cnn-lstm model. In *2024 International Conference on Intelligent Algorithms for Computational Intelligence Systems (IACIS)*, pages 1–6, 2024. doi: 10.1109/IACIS61494.2024.10721849.

[18] Musaed Alhussein, Khursheed Aurangzeb, and Syed Irtaza Haider. Hybrid cnn-lstm model for short-term individual household load forecasting. *IEEE*

Access, 8:180544–180557, 2020. doi: 10.1109/ACCESS.2020.3028281. URL https://ieeexplore.ieee.org/document/9187862.

[19] Ali Agga, Ahmed Abbou, Moussa Labbadi, and Yassine El Houm. Short-term load forecasting based on hybrid cnn-lstm neural network. In *2021 6th International Conference on Power and Renewable Energy (ICPRE)*, pages 886–891. IEEE, 2021. doi: 10.1109/ICPRE52634.2021.9635488.

[20] Zahraa Khais Shahid, Saguna Saguna, and Christer Åhlund. Forecasting electricity and district heating consumption: A case study in schools in sweden. In *2023 IEEE Green Technologies Conference (GreenTech)*, pages 169–175. IEEE, 2023. doi: 10.1109/GreenTech56823.2023.10173792.

[21] S. Atef and A. Eltawil. Assessment of stacked unidirectional and bidirectional lstm networks for electricity load forecasting. *Electric Power Systems Research*, 187:106489, 2020. doi: 10.1016/j.epsr.2020.106489. URL https://doi.org/10.1016/j.epsr.2020.106489.

[22] Danilo G. Silva and Alexandre A. M. Meneses. Comparing lstm and blstm deep neural networks for power consumption prediction. *Energies*, 17(17):4360, 2024. doi: 10.3390/en17174360. URL https://www.mdpi.com/1996-1073/17/17/4360.

[23] Abiodun M. Ikotun, Absalom E. Ezugwu, Laith Abualigah, Belal Abuhaija, and Jia Heming. K-means clustering algorithms: A comprehensive review, variants analysis, and advances in the era of big data. *Information Sciences*, 622:178–210, 2023. ISSN 0020-0255. doi: https://doi.org/10.1016/j.ins.2022.11.139. URL https://www.sciencedirect.com/science/article/pii/S0020025522014633.

[24] Keiron O'Shea and Ryan Nash. An introduction to convolutional neural networks. *ArXiv*, abs/1511.08458, 2015. URL https://api.semanticscholar.org/CorpusID:9398408.

[25] Benjamin Lindemann, Timo Müller, Hannes Vietz, Nasser Jazdi, and Michael Weyrich. A survey on long short-term memory networks for time series prediction. *Procedia CIRP*, 99:650–655, 2021. ISSN 2212-8271. doi: https://doi.org/10.1016/j.procir.2021.03.088. URL https://www.sciencedirect.com/science/article/pii/S2212827121003796. 14th CIRP Conference on Intelligent Computation in Manufacturing Engineering, 15-17 July 2020.

[26] Xiaolei Ma, Jiyu Zhang, Bowen Du, Chuan Ding, and Leilei Sun. Parallel architecture of convolutional bidirectional lstm neural networks for network-wide metro ridership prediction. *IEEE Transactions on Intelligent Transportation Systems*, 20(6):2278–2288, 2018.

[27] Chiou-Jye Huang and Ping-Huan Kuo. A deep cnn-lstm model for particulate matter (pm2. 5) forecasting in smart cities. *Sensors*, 18(7):2220, 2018.

[28] Tae-Young Kim and Sung-Bae Cho. Predicting residential energy consumption using cnn-lstm neural networks. *Energy*, 182:72–81, 2019.

[29] Microsoft Learn. What is spark compute in microsoft fabric? `https://learn.microsoft.com/en-us/fabric/data-engineering/spark-compute`, 2024. Accessed: 2025-05-11.

[30] Nord Pool. Elspot day-ahead market at the transitions to/from summer time (daylight saving time) and standard time. `https://www.nordpoolgroup.com/en/pages/elspot-day-ahead-market-at-the-transitions-tofrom-summer-time-daylight-saving-t`, n.d. Accessed: 2025-05-11.

[31] Jim Holdsworth, Ivan Belcic, and Cole Stryker. What is model drift? `https://www.ibm.com/think/topics/model-drift#:~:text=Model%20drift%20refers%20to%20the,decision-making%20and%20bad%20predictions`, 2024. Accessed: 2025-02-18.

[32] Ruoyu Yang, Shubhendu Kumar Singh, Mostafa Tavakkoli, Nikta Amiri, Yongchao Yang, M. Amin Karami, and Rahul Rai. Cnn-lstm deep learning architecture for computer vision-based modal frequency detection. *Mechanical Systems and Signal Processing*, 144:106885, 2020. ISSN 0888-3270. doi: https://doi.org/10.1016/j.ymssp.2020.106885. URL `https://www.sciencedirect.com/science/article/pii/S0888327020302715`.

[33] Guillem Capellera, Luis Ferraz, Antonio Rubio, Antonio Agudo, and Francesc Moreno-Noguer. Footbots: A transformer-based architecture for motion prediction in soccer. In *2024 IEEE International Conference on Image Processing (ICIP)*, pages 2313–2319. IEEE, 2024.

A

Appendix 1

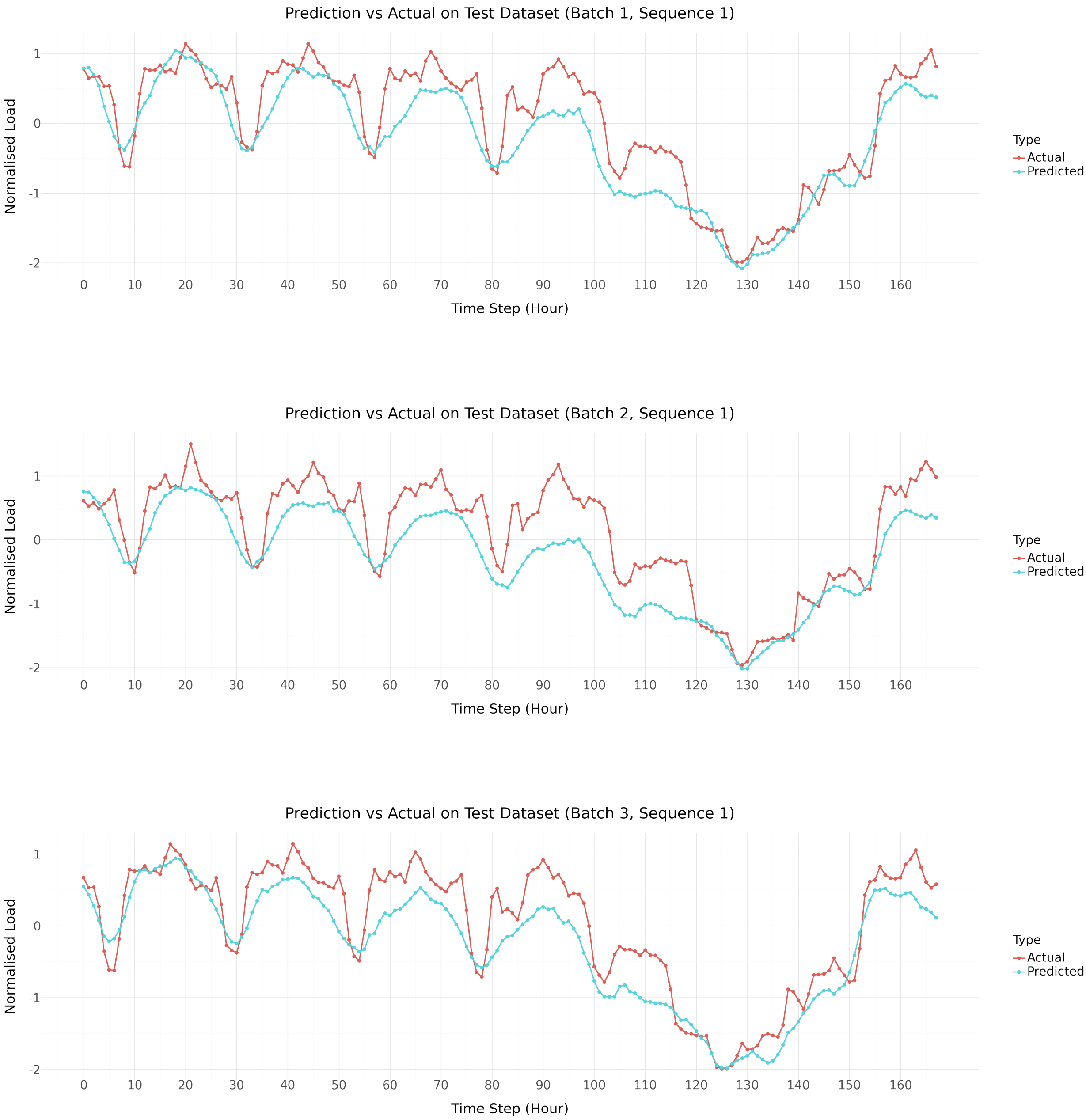
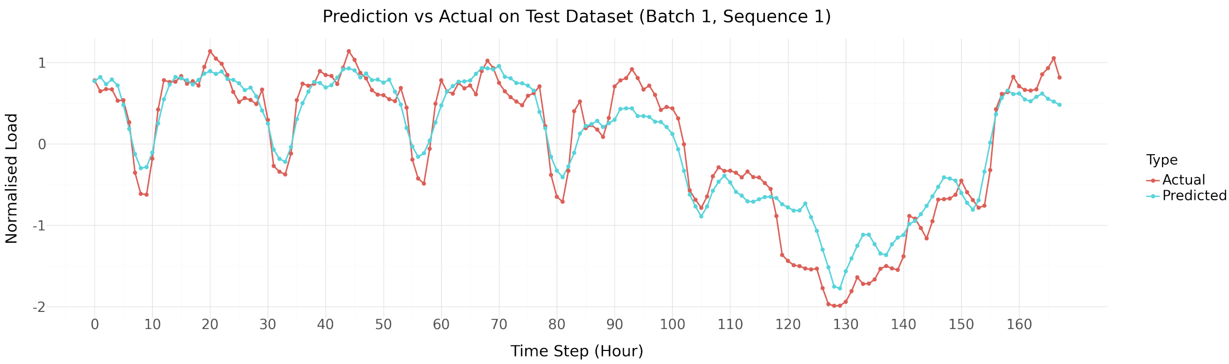
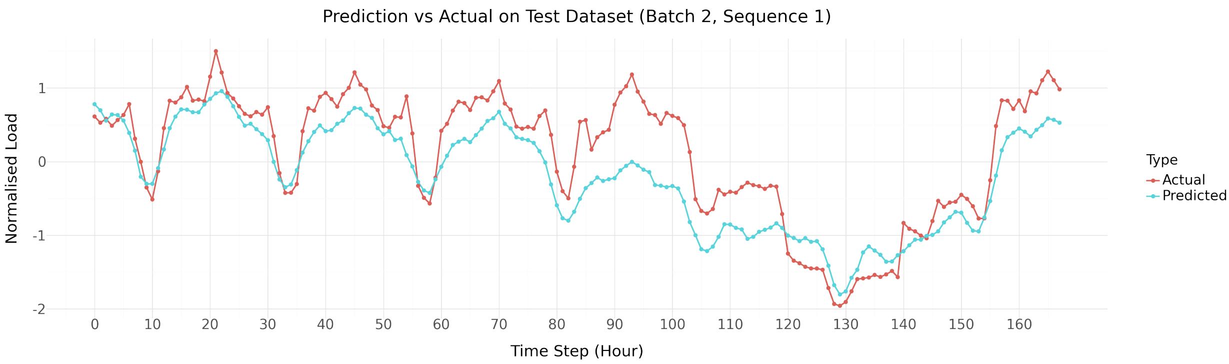


Figure A.1: Vanilla LSTM: Prediction vs Actual on Test Dataset for three different batches.

Figure A.2: Vanilla LSTM: Prediction vs Actual on Prediction Dataset for three different batches.





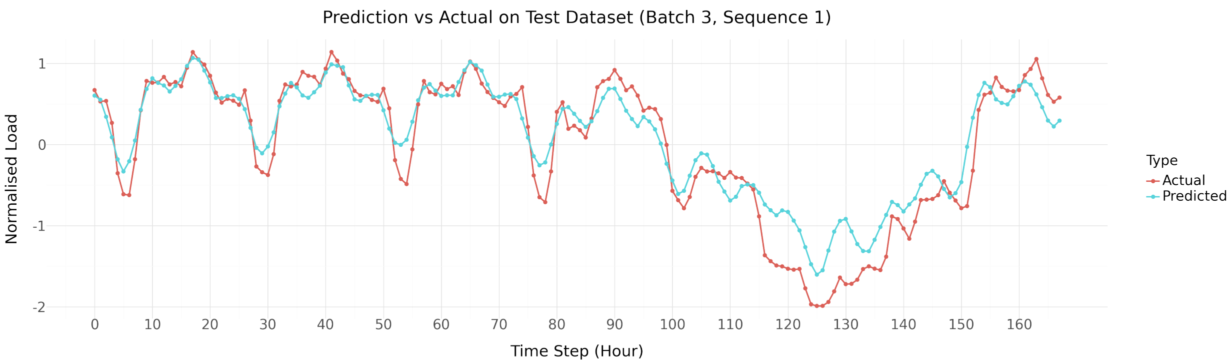


Figure A.3: Stacked LSTM: Prediction vs Actual on Test Dataset for three different batches.

Figure A.4: Stacked LSTM: Prediction vs Actual on Prediction Dataset for three different batches.

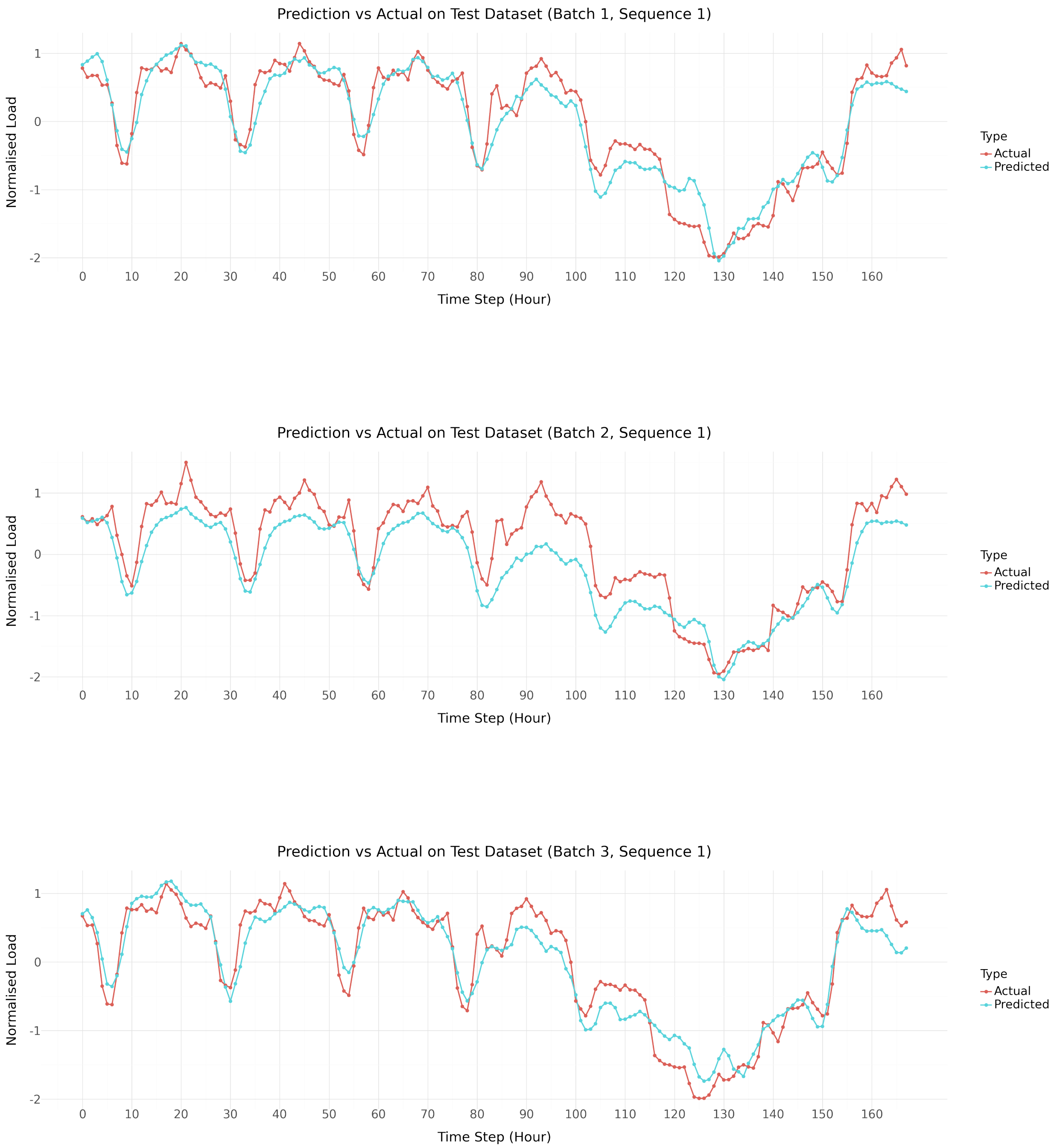
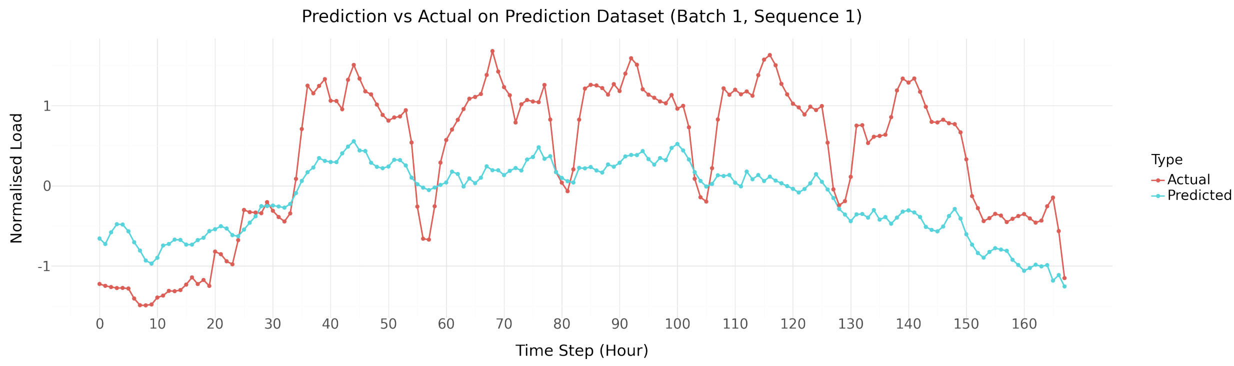
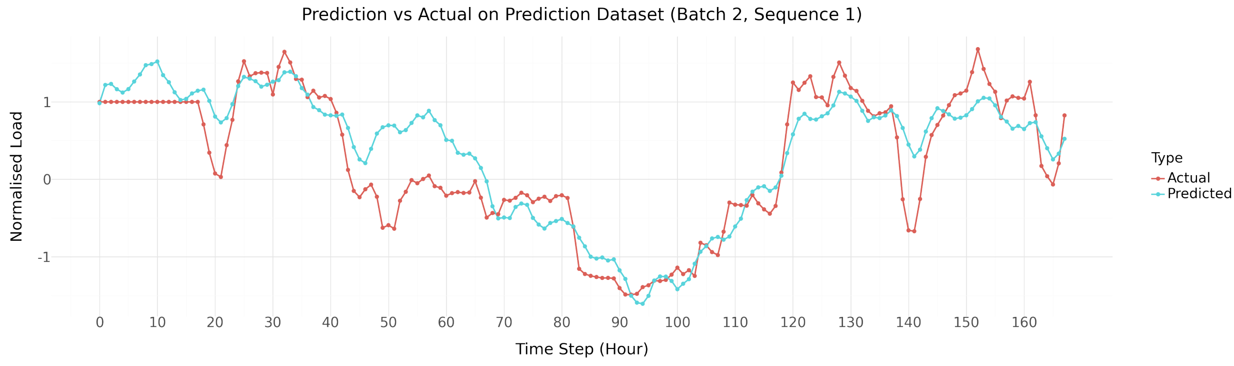


Figure A.5: Bidirectional LSTM: Prediction vs Actual on Test Dataset for three different batches.





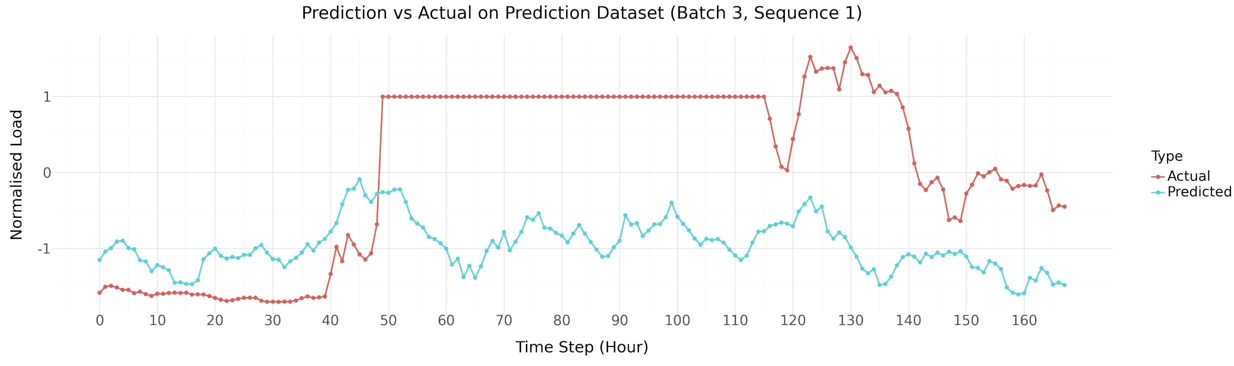
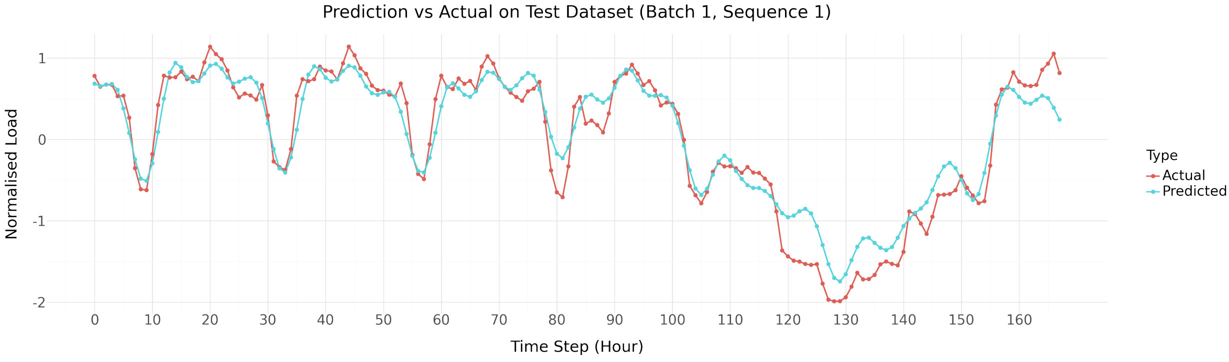
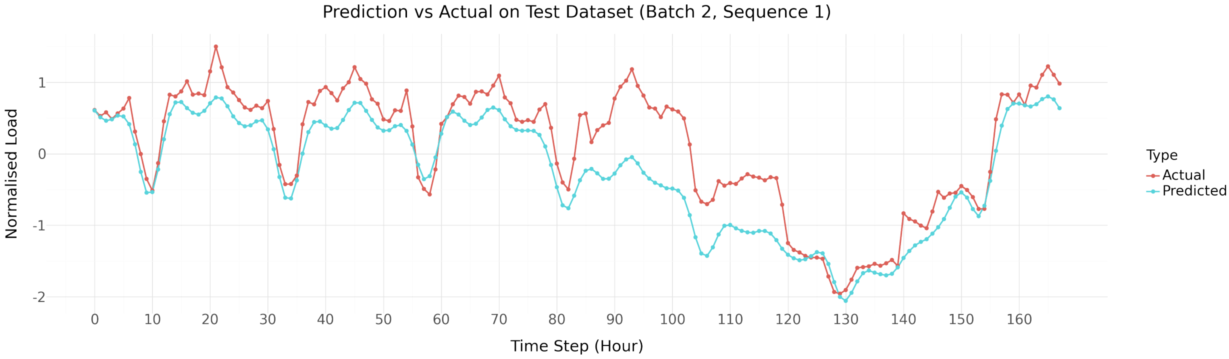


Figure A.6: Bidirectional LSTM: Prediction vs Actual on Prediction Dataset for three different batches.





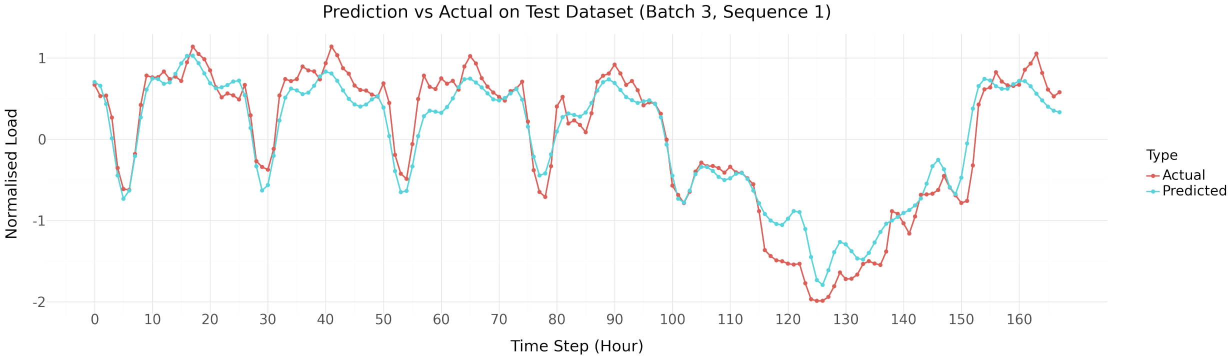


Figure A.7: CNN-LSTM: Prediction vs Actual on Test Dataset for three different batches.

Figure A.8: CNN-LSTM: Prediction vs Actual on Prediction Dataset for three different batches.