Norwegian University of Life Sciences

Master's Thesis 2023 30 ECTS
Faculty of Science and Technology (REALTEK)

Predicting electricity demand using machine learning: Case study of Oslo Airport Gardermoen

Prediksjon av elektrisitetsbruk ved bruk av maskinlæring: Casestudie av Oslo Lufthavn Gardermoen

Sigurd Grøtan
Environmental Physics and Renewable Energy

Abstract

With an increasing need for electricity in society and a more complex energy production mix, future electrical power systems require intelligent systems for efficient resource management. In order to realize the potential of flexible resources in the power grid, robust and accurate prediction methods are required. This thesis presents a case study of Oslo Airport Gardermoen (OSL) to explore the potential of Long Short-Term Memory (LSTM) machine learning models in predicting electricity demand, particularly focusing on peak demand forecasting. The models are trained on data from 2022 and 2023, utilizing electricity consumption measurements and exogenous factors including passenger numbers, outdoor temperature, and electricity prices. The models demonstrate high accuracy in demand prediction, particularly for peak hours.

To improve peak prediction capabilities, the thesis implements two main strategies. First, models are trained using four different loss functions: Mean Squared Error (MSE), Mean Absolute Percentage Error (MAPE), Negative Log Likelihood (NLL), and a new proposed Weighted Mean Squared Error (WMSE). Second, a comprehensive grid search and cross-validation routine is performed to robustly determine the optimal model architectures. The best-performing models are characterized by simple model architectures with just 1 hidden layer and 64 or 128 units, suggesting that less complex models can efficiently capture the patterns of the data. These models achieve adequate MAPE scores, with the lowest being 4.53%.

The new proposed WMSE loss function emphasizes peak hours and significantly enhances peak prediction reliability. Additionally, NLL enables probabilistic outputs, offering valuable uncertainty estimations for practical applications. This thesis provides a robust and versatile framework adaptable to various energy systems, enabling the development of optimized LSTM models for efficient electricity demand forecasting.

The implications of this work extend beyond OSL, offering insights for managing flexible resources for efficient and sustainable power system operation. The thesis highlights the promising potential in using advanced machine learning methods for energy management systems, and demonstrates their ability in large-scale commercial buildings.

Sammendrag

Med økende elektrifisering i samfunnet og mer kompleks energiproduksjon trenger fremtidens kraftsystemer intelligente systemer for effektiv ressursbruk. For å kunne bruke fleksible løsninger i kraftnettet er det et behov for robuste og treffsikre prediksjonsmetoder. Denne masteroppgaven er en casestudie av Oslo Lufthavn Gardermoen (OSL) som undersøker potensialet i å benytte maskinlæringsmodeller basert på Long Short-Term Memory (LSTM) for å predikere strømforbruk, med et spesielt fokus på forbrukstoppene. Modellene trenes på data fra 2022 og 2023, og bruker målinger av tidligere strømforbruk og forklaringsvariabler som passasjertall, utetemperatur, og strømpriser. Modellene predikerer strømforbruket med god treffsikkerhet, særlig med tanke på forbrukstoppene.

For å forbedre prediksjonene av toppene benyttes to hovedmetoder. Den første er at modellene trenes med fire forskjellige tapsfunksjoner: *Mean Squared Error* (MSE), *Mean Absolute Percentage Error* (MAPE), *Negative Log Likelihood* (NLL), og en ny foreslått tapsfunksjon *Weighted Mean Squared Error* (WMSE). Den andre metoden er at det gjennomføres en omfattende *grid search* med kryssvalidering for å finne optimaliserte modellarkitekturer. Modellene med best ytelse har enkle modellarkitekturer bestående av bare 1 skjult lag og enten 64 eller 128 noder, noe som antyder at mindre komplekse modeller er i stand til å effektivt lære de underliggende mønstrene i datasettet.

Den nye tapsfunksjonen WMSE legger mer vekt på topplasttimene, og gir en betydelig økning i pålitelighet når det kommer til prediksjon av toppene. Videre gir NLL sannsynlighetsbaserte prediksjoner, som tilfører prediksjonene et verdifullt usikkerhetsestimat for praktiske anvendelser. Metodikken i denne oppgaven legger fram et robust og allsidig rammeverk for å utvikle optimaliserte modeller som kan forutsi strømforbruket for en rekke systemer som OSL.

Betydningen av dette arbeidet går forbi case-studiet av OSL, og gir verdifull innsikt mot å benytte fleksible ressurser for effektiv og bærekraftig drift av kraftsystemet. Resultatene understreker et lovende potensial for å bruke avanserte maskinlæringsmodeller for effektiv styring av kraftsystemer, og demonstrerer treffsikker anvendelse i store kommersielle bygg.

Acknowledgements

I would like to thank my supervisors Heidi S. Nygård and Leonardo Rydin Gorjão for their helpful advice throughout this work. Also, I want to thank Thomas Martinsen and the rest of the NeX2G group for giving me the opportunity to work on such an exciting project. Finally, I must thank my housemates Markus, Daniil, and Kenneth for creating a wonderful living situation during these stressful times.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

The global community is facing an alarming pace of global warming. Contemporary power systems need to undergo an extensive transition from carbon-emitting resources to renewable energy, and several sectors of society such as transportation and industry require electrification. In line with these global initiatives, the Norwegian government has committed to reducing its greenhouse gas emissions by at least 50% by 2030, compared to the levels recorded in 1990 [1].

Driven by decreasing prices and political support, renewable energy sources such as solar and wind are gaining prominence in the energy landscape. The integration of renewables is likely to increase, which poses challenges in maintaining stability and safe operation in the power grid. Achieving a constant balance between generated and consumed energy becomes increasingly difficult with the growing complexity of energy generation. The introduction of new electrical loads, like electric vehicles (EVs), further intensifies the demands on the power grid, complicating the problem [2].

To ensure an instantaneous balance between electricity supply and demand, power grids must be capable of handling not only the total electricity consumption but also peak power demands. Expanding the capacity of the grid is a time-intensive and costly process. Therefore, it is essential to identify and reduce peak power demands [3].

Thus, there is a need for intelligent systems and tools to schedule ahead and optimize the use of available electricity sources in the power system. Forecasting of electricity demand is required for efficient operation of the power grid. By gaining insight into future electricity needs, proactive measures can be taken to avoid large peaks in power consumption [4]. Extensive research has been done to successfully predict power demand, with accurate prediction of the peaks recognized as both important and challenging [5].

While predicting electricity consumption is a complex task, technological advancements and the growing availability of data have paved the way for promising results using machine learning. Machine learning models, requiring no physical knowledge of energy systems, can provide successful predictions through training on historical data [6]. Long Short-Term Memory

(LSTM) models, particularly renowned for effectively handling large time intervals of data in time series predictions, have demonstrated enormous success in forecasting time series [7]. Hence, the primary purpose of this thesis is to research the effectiveness of utilizing LSTM algorithms for forecasting electricity demand.

1.2 Motivation

This thesis contributes to the NeX2G research project [8], funded by the Norwegian Research Council under grant number 320825. The objective of the project is to investigate the flexibility potential in electric vehicles and other facilities at Oslo Airport Gardermoen (OSL). Successful forecasting of electricity consumption is crucial for estimating flexibility potential. The thesis focuses on developing a functional machine learning model for predicting electricity consumption at the airport.

This thesis extends the research conducted by Kvisberg [9], which explored the use of LSTM models to forecast electricity demand at OSL. While these models successfully captured general consumption patterns, they were less effective in predicting power peaks. Therefore, the aim of this thesis is to investigate methods for enhancing the peak prediction capabilities of the LSTM models.

1.3 Problem Statement

The objective of this thesis is to develop a machine learning model that can accurately predict electricity consumption at OSL, particularly focusing on identifying future power peaks. The model utilizes an LSTM algorithm and is trained using data from 2022 and 2023. It aims to forecast electricity usage 24 hours ahead.

To improve peak prediction in the LSTM models, this research employs two primary strategies. First, the models are trained with four distinct loss functions to evaluate their impact on performance. Among these, one function enables probabilistic model outputs, while another is specifically weighted to focus on peak hours. Second, a comprehensive optimization process is conducted. A grid search is used to identify the optimal parameters for constructing the LSTM model architecture, and this process is carried out separately for each loss function. This approach results in four distinct, yet optimized LSTM models. The performance of these models is assessed using error metrics and empirical analysis, with special attention given to their accuracy in predicting power consumption peaks.

Chapter 2

Theory

2.1 Power Systems

This section presents key concepts in electric power systems. It begins by outlining the general characteristics of electric power grids, then examines electricity consumption patterns in buildings and industries. The final part of the section explores the concept of flexibility within power systems.

2.1.1 The Power Grid

An electric power supply system is divided into three primary categories: production, transmission, and consumption. Electricity is generated in the production sector, distributed through the transmission grid, and spent by the consumers. As many fundamental functions of modern society depend on constant access to electricity, a reliable supply of electrical energy is crucial [10].

The Norwegian power grid is divided into three levels. Firstly, the transmission grid connects producers and consumers in a nation-wide system and interconnects the Norwegian grid with surrounding countries. The lines in the transmission grid carry a high voltage of typically 300 or 420 kV with some parts carrying 132 kV. In Norway, Statnett acts as the transmission system operator (TSO). Next, the regional distribution grid commonly functions as a link between the transmission grid and the distribution grids, and it carries a voltage of 33 to 132 kV. Industrial consumers operating at higher voltages and producers may be connected to this grid as well. Finally, the local distribution grid supplies electricity to smaller end users. It carries a voltage of up to 22 kV, and commonly supplies voltages of 230 to 400 V to ordinary customers. The regional and local distribution grids are managed by entities known as distribution system operators (DSOs), which are commonly owned by the municipalities and county authorities [10].

Since electricity is challenging to store, the production must be equal to the demand at all times. This is known as the instantaneous balance in the power system, and it is critical to ensure safe and reliable operation of the grid. Aberrations from the instantaneous balance cause the frequency of the system to deviate from the nominal 50 Hz [11]. This leads to instability in the system which can potentially result in damage and failure of electrical components in the power grid [3].

Hence, the capacity of the power grid must be able to handle the instantaneous power demand. The peak demand determines the required capacity of the grid, although large peaks occur a fraction of the time. The necessary grid capacity also varies geographically, as some areas may have higher electric power requirements than others. To accommodate increased power demand, physical expansions can be made to enhance grid capacity, but this approach is both expensive and time-consuming. An alternative is to implement measures to reduce large peaks, thereby optimizing the use of existing capacity [12].

2.1.2 Electricity Consumption in Buildings and Industry

In Norway today, approximately 50% of electricity consumption occurs in buildings, both residential and commercial, and is largely attributed to electrical heating systems [1]. The Norwegian Water Resources and Energy Directorate (NVE) estimates that more efficient energy usage in buildings could potentially reduce the total electricity consumption in Norway by as much as 10% [13]. Since the majority of building consumption in Norway is used for heating, electricity usage is highly dependent on weather factors, a correlation that is stronger in Norway compared to other countries [14].

Another 40% of electricity consumption in Norway is attributed to industrial applications, including sectors such as petroleum production, manufacturing plants, and data centers. The expected increase in electricity consumption in the coming years is primarily due to these sectors [1]. This increase is a result of industrial applications transitioning from carbon-emitting energy sources to electrification, as well as the establishment of new industries [12].

Airports, in addition to hosting airplane traffic, need to facilitate a large variety of services. As a result, they can resemble small cities with significant electricity demands for heating, ventilation, and more. Implementing various measures such as intelligent energy management systems could enable significant reductions in electricity consumption [15]. Yildiz et al. [16] proposed energy-saving projects for an airport in Turkey, and reported that the energy consumption could be reduced by as much as 57%.

2.1.3 Flexibility

Flexibility is defined by the Centre for Intelligent Electricity Distribution (CINELDI) as *the ability and willingness to modify production and/or consumption patterns, at an individual or aggregated level, often in response to an external signal, in order to provide a service to the power system or maintain stable grid operation* [17].

Flexibility within the power system can be offered by various sectors. Electricity producers, like hydro power plants; energy storage solutions, such as batteries; and consumers, including industrial buildings, can all contribute to this flexibility. This potential can be utilized to address challenges regarding voltage quality, bottlenecks, and grid capacity [18].

To prevent bottlenecks, attrition of electrical components, and potential grid failure, the transmission capacity must be capable of handling the peak power for a certain line. If demand were allowed to occur freely, the peak demand would be significantly larger than during most hours throughout the year, resulting in poor management of grid resources. Thus, to better make use of the available capacity, flexibility can be used to reduce the magnitude of the peaks [19]. Different strategies for reducing peaks are shown in figure 2.1. Peak shaving refers to the practice of reducing the amount of energy used during peak demand times. Conversely, valley filling means increasing electricity use during periods of low demand. Load shifting involves transferring electricity usage from one period to another, typically from peak to off-peak hours, and includes both peak shaving and valley filling. The objective is to level the overall demand curve by minimizing the discrepancy between peak and off-peak demand.

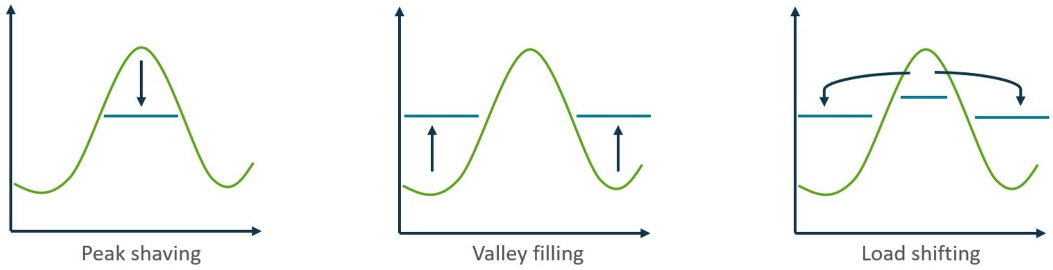

FIGURE 2.1: Strategies for reducing peaks in the electric power consumption. Reproduced from [20].

In Norway, the largest flexibility potential for buildings are assumed to be connected to thermal storage. This includes systems such as water heaters, cooling systems, and other loads that can be disconnected for short periods of time without impacting user comfort [19]. To reduce the peak power consumption of a building, strategies like load shifting, as depicted in figure 2.1, can be applied. In turn, by reducing the peak consumption, the building is providing the overall power system flexibility.

Another example of a flexible load is the charging of EVs, as it can be time-shifted to periods of low demand and prices. Provided the vehicle is done charging when it is needed, the comfort of the user has not been impacted at all. In addition, a new technology called vehicle-to-grid (V2G) further increases the flexibility potential of EVs. While normally the charging can only go from the grid to the vehicle, V2G allows the charge to flow both ways. As a result, EVs can act as batteries to supply the power grid in times of high demand [19]. This could prove to be a significant flexibility tool, especially when managing several EVs together [21]. V2G solutions, along with other technologies such as stationary batteries and hydrogen energy storage, can help balance the intermittent nature of renewable energy sources like solar and wind. They achieve this by charging during periods of high energy production and discharging according to demand [22].

There are two primary types of flexibility. The first is explicit flexibility, which is activated upon receving an external signal from a third party, such as a grid company, system operator, or energy supplier. This activation could involve either increasing or decreasing electricity production or consumption. Entities providing explicit flexibility are typically compensated for their service. Although this strategy can efficiently and reliably trigger a flexibility response, it requires both technical and administrative coordination. Implicit flexibility, in contrast, is triggered by price signals in the electricity market. Producers, consumers, and battery owners themselves decide when and how much flexibility to provide. The motivation behind offering flexibility here is economic gain, which can automatically encourage participation [19]. In the context of this thesis, implicit flexibility is the most relevant.

In Norway, the total electricity cost for consumers in the power grid is determined both by the amount of energy and the power demanded. It consists of several components: the power price, grid rental fee, and taxes to the government. The power price [NOK] is a variable cost per unit of electricity [kWh] consumed. The grid rental fee includes a fixed component and a variable component. The fixed component varies based on the peak power [kW] usage and is intended to cover the costs of operating the power grid. The variable part, known as the energy component, is intended to offset the marginal loss costs incurred in transmitting electricity to the customer and increases with higher electricity usage [23]. The structure of the grid rental fee, being influenced by peak power consumption, provides consumers with a financial incentive to distribute their peaks more evenly.

The future power systems are often referred to as *smart grids*. These systems are distinguished by their use of digital technology to enhance the efficiency and reliability of electricity management. A central feature of smart grids is the efficient management of flexible loads, which helps balance electricity production and consumption while minimizing peak demand levels. A fundamental requirement for the effective functioning of smart grids is the accurate prediction of electricity consumption patterns [24]. This accurate forecasting is essential for effective resource management, such as implementing measures to reduce peak demand, as illustrated in figure 2.1.

2.2 Machine Learning

The following section will describe the relevant topics in machine learning. The descriptions closely follow the explanations in [25] and uses the notations and illustrations of [26]. Unless specified, the following descriptions are cited from these two sources.

2.2.1 Fundamentals of Machine Learning

Machine learning (ML) evolved as a sub-field of artificial intelligence (AI) tasks and is centered around self-learning algorithms that are able to extract knowledge from data in order to make predictions. The initial applications of machine learning algorithms were primarily in image classification, but they have since been applied to a wide variety of problems. In this context, learning means the ability of the algorithm to improve its output by incorporating new data. Machine learning is often divided into three different types: supervised learning, unsupervised learning, and reinforcement learning.

Supervised learning consists of learning a model from labeled training data in order to make predictions about unseen or future data. The term supervised refers to a set of training data where the desired output of the model is already known. Therefore, the training data consists of input data known as features, and the desired output which is known as targets. The output of the model is known as predictions, and the goal of the algorithm is to minimize the error between the predictions and the targets. Further, unsupervised learning deals with unlabeled data or data of unknown structure. These techniques are used to extract meaningful information or patterns from data without the help of a known outcome. An example of unsupervised learning is clustering, a method where data points that share a certain similarity are grouped together without any prior knowledge of group memberships. Finally, in reinforcement learning, the goal is to develop a system which improves its performance based on its interactions with the environment. An example of reinforcement learning is a chess engine, which attempts a series of moves based on the state of the board and is rewarded according to its performance.

In this thesis, the focus is on supervised learning, which can be further divided into classification and regression. In classification, the goal is to predict discrete class labels based on previous observations. In these problems, the class memberships are unordered and can be binary or multiclass, and an example of a classification model could be recognizing handwritten digits. In regression, on the other hand, the goal is to predict a continuous value. An example of a regression model could be predicting the air temperature the next day.

2.2.2 Artificial Neural Networks

The basics of artificial neural networks can be explained with the Adaline algorithm [27], which can be seen as a single-layer neural network. This network consists of several input nodes, a single net input function, and an activation function generating a single output. An Adaline algorithm for binary classification is shown in figure 2.2.

The vector x signifies the input values of the algorithm, where the subscript m refers to the number of features in the training data. In addition to x, there

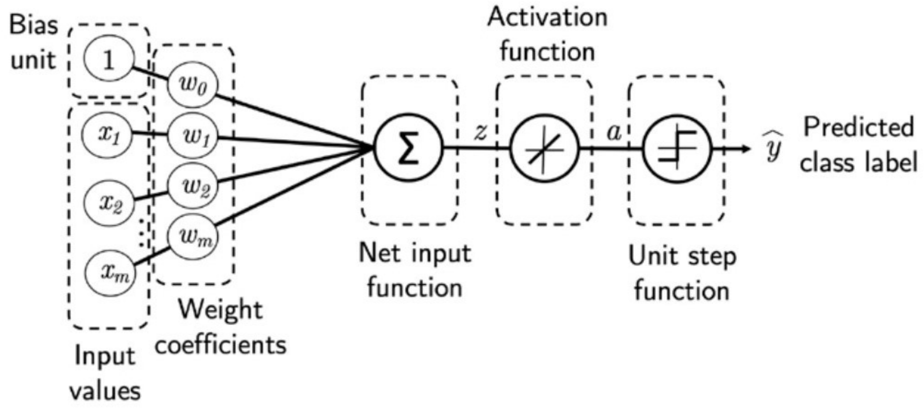

FIGURE 2.2: Illustration of the Adaline algorithm. The net input of the network is calculated as the linear combination of the weights and the inputs plus bias. The signal is then passed through the algorithm to produce an output. Reproduced from [26].

is a bias unit with a constant value of 1. Through a net input function, a linear combination of x plus the bias unit and the weight vector w is calculated, giving a single net input value z. Next, z is passed into an activation function. For Adaline, the output a of the activation function is the same as input z, but this is not necessarily the case for other algorithms. The algorithm in figure 2.2 also contains a unit step function squashing a into a binary output. By removing the unit step function a would become the output $\hat{y}$ of the algorithm giving a continuous value, making it a regression algorithm. Since this thesis uses regression models this will be the focus moving forward.

During model training, the weight vector w is updated based on an error calculated by a loss function. The loss function is used to quantify the performance of the algorithm by calculating the error between the predicted value $\hat{y}$ and the true value y. Through a technique called gradient descent, the aim is to move in the opposite direction of the gradient of the loss function. The gradient is found by calculating the partial derivatives of the loss function for each weight in the vector w. Although the Adaline algorithm can be described as a simple single-layer neural network the same underlying concepts apply to neural networks of more complex structures.

To introduce the multilayer neural network, a multilayer perceptron (MLP) is depicted in figure 2.3. The MLP is a feedforward neural network, meaning that the signals flow from one end of the network to another. The MLP in figure 2.3 consists of one input layer, one hidden layer, and one output layer. The hidden layer is fully connected to the input layer, and the output layer is fully connected to the hidden layer. Neural networks containing more than one hidden layer are referred to as deep artificial neural networks.

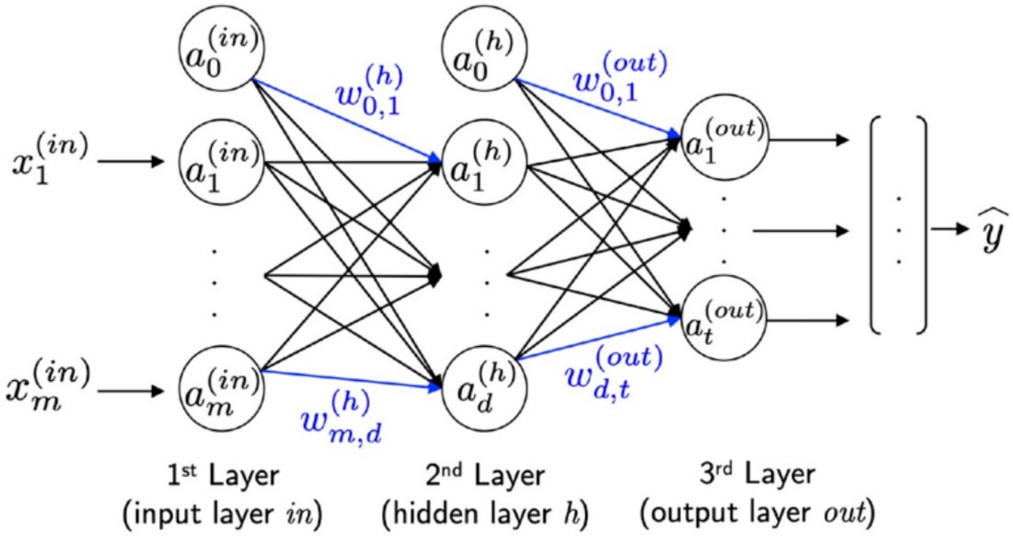

FIGURE 2.3: Illustration of an MLP. For this neural network, the input layer has m nodes corresponding to each input value plus one bias. The hidden layer has d nodes plus one bias. The output layer has t nodes. Reproduced from [26].

The learning process of an MLP is similar to that of the Adaline algorithm, but the flow of information is more complicated. As seen in figure 2.3, the node $a_d^{(h)}$ receives a signal from every node in the preceding layer and has a weight for every signal. Node $a_d^{(h)}$ uses a linear combination of all the signals and weights to create its own activation, which it passes on to the nodes in the next layer. This happens for all the nodes in the hidden layer except for the bias node $a_0^{(h)}$, and in all other layers except for the input layer which simply receives one input from each of the features in the data set with no weights. Thus, signals from the training data are propagated forward through the network to generate a prediction. Next, the loss function calculates the error between the prediction and the target value. Since the output in one layer is a function of the activation of the preceding one, the error is sent backwards in the network through a technique called the backpropagation algorithm. As follows, the derivatives of the loss function with respect to each weight in the network are found, and through gradient descent the model is updated. A more detailed explanation going deeper into the mathematical operations of backpropagation and the training of neural networks can be found in [26].

2.2.3 Recurrent Neural Networks

A recurrent neural network (RNN) is a type of neural network which receives sequential data as input. Sequential data could for instance be samples of text or a time series, and is characterized by the order of the samples being a significant attribute of the data. In order to handle multiple time steps RNNs allow information to flow through something called a recurrent edge, which is illustrated in figure 2.4.

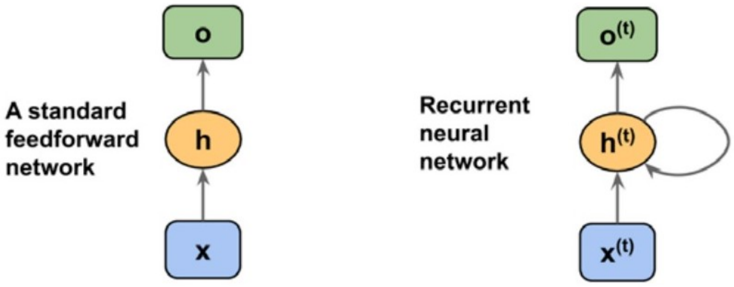

FIGURE 2.4: Illustration of the RNN concept. Unlike standard feedworward neural networks, RNNs have a recurrent edge allowing a flow of information between time steps. Reproduced from [26].

In a standard feedforward network such as the MLP discussed in the previous section, information flows from the input layer to the hidden layer, and from the hidden layer to the output layer. In an RNN, the hidden layer receives information from the input layer of the current time step and the hidden layer of the previous time step. Each hidden layer has weights associated with both the preceding layer at the current time step and the hidden layer at the previous time step. As a result, the RNN is able to keep a memory of past events. Similar to the MLP, an RNN can contain several hidden layers. This is shown in figure 2.5, which also unfolds the recurrent edge to illustrate how multiple time steps are connected.

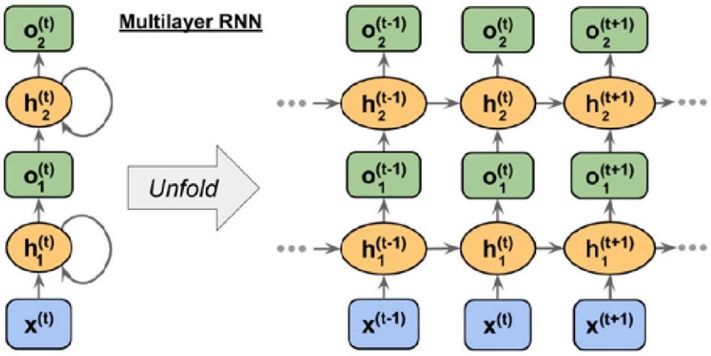
Multilayer RNN
Unfold
$o_2^{(t)}$
$h_2^{(t)}$
$o_1^{(t)}$
$h_1^{(t)}$
$x^{(t)}$
$o_2^{(t-1)}$ $o_2^{(t)}$ $o_2^{(t+1)}$
$h_2^{(t-1)}$ $h_2^{(t)}$ $h_2^{(t+1)}$
$o_1^{(t-1)}$ $o_1^{(t)}$ $o_1^{(t+1)}$
$h_1^{(t-1)}$ $h_1^{(t)}$ $h_1^{(t+1)}$
$x^{(t-1)}$ $x^{(t)}$ $x^{(t+1)}$

FIGURE 2.5: Illustration showing a multilayer RNN unfolded, displaying the flow of information between time steps in a multi-layer structure. Reproduced from [26].

RNNs face significant challenges when dealing with longer sequences, primarily due to the vanishing and exploding gradient problems [28]. RNNs leverage a specialized form of backpropagation called backpropagation through time (BPTT). BPTT effectively unfolds the temporal layers of the network into a traditional feedforward architecture, enabling the application of standard backpropagation for weight updates. However, this process becomes problematic with lengthy sequences, as it involves multiplying a large number of partial derivatives. This multiplication can lead to gradients that are excessively large (exploding) or too small (vanishing), which makes the network unable to learn long-term dependencies.

To mitigate these issues, one of the popular solutions is the LSTM algorithm [28]. LSTMs are specifically designed to address the shortcomings of traditional RNNs in learning long-term dependencies, offering a more robust architecture for handling extended sequences. The core component of the LSTM is the memory cell, which is depicted in figure 2.6.

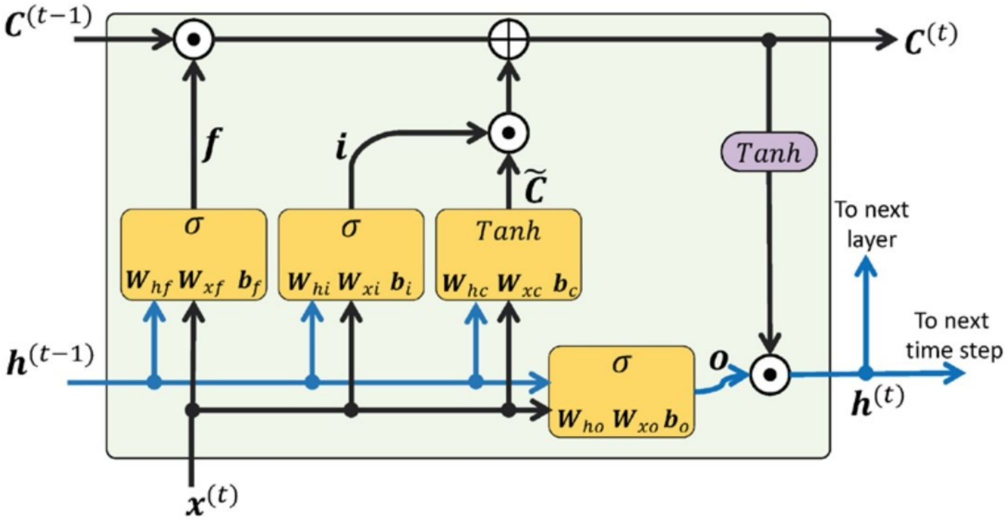

FIGURE 2.6: Illustration of the LSTM architecture, displaying the flow of information in the memory cell. The cell state C_t is regulated by a series of computational units called gates. Reproduced from [26].

In an LSTM network, the memory cell plays a critical role, essentially taking over the function of the hidden layer found in standard RNNs. The values that flow along this part of the network are known collectively as the cell state. One of the key features of the cell state is its ability to traverse through all the time steps without being directly subjected to weight multiplication, which is a significant factor in its ability to manage information flow effectively. This flow is regulated by a series of computational units called gates. There are several variants of the memory cell architecture in an LSTM network, and the following explanations follow the one presented in [26].

The forget gate f_t decides which information is retained or discarded. It processes the previous hidden state h_{t-1} and the current input x_t through a sigmoid function, and then applies the resultant value to modify the previous cell state C_{t-1}. Next, the input gate i_t and a candidate value $\tilde{C}_t$ work together to update the cell state. The input gate filters the incoming data (x_t and h_{t-1}) through a sigmoid function, while the candidate value uses a tanh function for the same inputs. The outputs of these two are then multiplied and added to the cell state, leading to its update. Finally, the output gate is responsible for determining the update to the hidden units, forming the next hidden state (h_t). It processes h_{t-1} and x_t through a sigmoid function, and the output from this is multiplied by the tanh of the cell state. This complex mechanism allows the LSTM to effectively navigate and address the challenges of vanishing and exploding gradients that are common in traditional RNNs [28].

2.2.4 The Learning Process

The learning process of a deep learning algorithm typically includes several key components in addition to the model itself. The aim of the training phase is to minimize the loss function, in which the parameters known as learning rate and epoch play a significant role. The learning rate controls how much the weights of the model are updated with respect to the loss gradient, and is therefore crucial for the convergence of the model to an optimal set of weights. This convergence is represented in figure 2.7, where the significance of the learning rate becomes evident. A too-small learning rate may cause the algorithm to get stuck in a local minimum, while a too-large learning rate can cause the algorithm to completely overshoot the global minimum. An epoch refers to one complete pass through the entire data set. The number of epochs therefore decides how many times the weights are adjusted, and multiple epochs are often necessary for the algorithm to learn the patterns of the data.

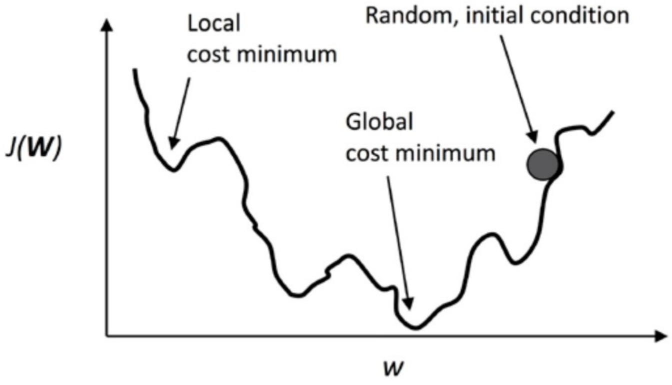

FIGURE 2.7: This illustration depicts the concept of convergence in model training. The vertical axis represents the value of the loss function, and the horizontal axis represents the value of a weight coefficient. The objective of the learning process is to find the value of the weight value that minimizes the loss function. Reproduced from [26].

2.2.5 Model Optimization

In machine learning, the primary goal is to find a function that accurately predicts not just the data it is trained on, but also unseen data. This involves splitting the full dataset into two sets: a training set for learning the model weights and a validation set to estimate how well the model will perform on new data.

The effectiveness of a model is judged on its ability to minimize errors in the training set while also performing effectively on unseen data. A significant difference in performance between these two sets indicates a problem. If the model performs well on the training data but poorly on the test data, the model is overfitting, often due to excessive complexity. Conversely, underperforming on the training data but having a small error on the validation set indicates underfitting, often due to a model being too simple. The model complexity dictates how intricate the learned predictions can be. Balancing this complexity is crucial, as it affects both overfitting and underfitting tendencies. This is illustrated in figure 2.8. The best machine learning model for a particular task is not always straightforward and can vary depending on the specific needs of the problem.

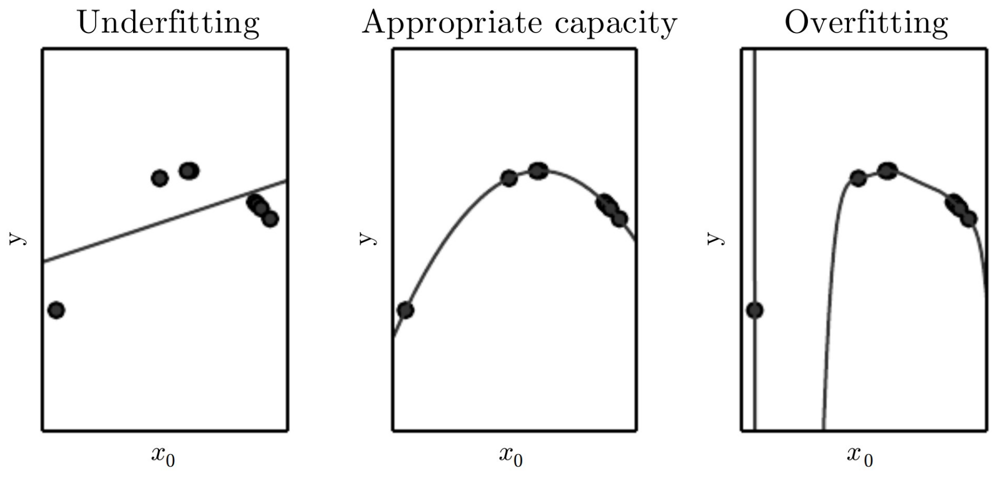

FIGURE 2.8: Illustrations depicting underfitting, overfitting, and the appropriate capacity of the model. The capacity of a model is related to the complexity. Reproduced from [29].

Optimizing models involves tuning parameters that influence their complexity and learning process. These parameters include the number of hidden layers, the number of units in each hidden layer, the number of epochs, and the learning rate, and are known as hyperparameters because they are set before the learning begins. Grid search is a popular method for hyperparameter optimization. It aims to find the optimal set of hyperparameters through an exhaustive search, where various values for different hyperparameters are systematically evaluated to find the best combination. Although this technique is simple, grid search can be computationally intensive as it requires training and evaluating a large number of models. Another approach is the random search, which randomly selects a limited number of combinations within the specified range. While it does not test every possible combination like grid search, random search can be an efficient alternative, saving computational resources.

Some hyperparameters can be optimized using alternative methods, like the number of epochs. Since this hyperparameter influences how many times the weights of the model are updated, it significantly impacts the model complexity. This can be managed through a technique known as early stopping. Early stopping halts the training process when the model performance on the validation set stops improving. The point, as depicted in figure 2.9, is considered the optimal level of complexity for the model. Consequently, controlling the number of epochs helps prevent both overfitting and underfitting.

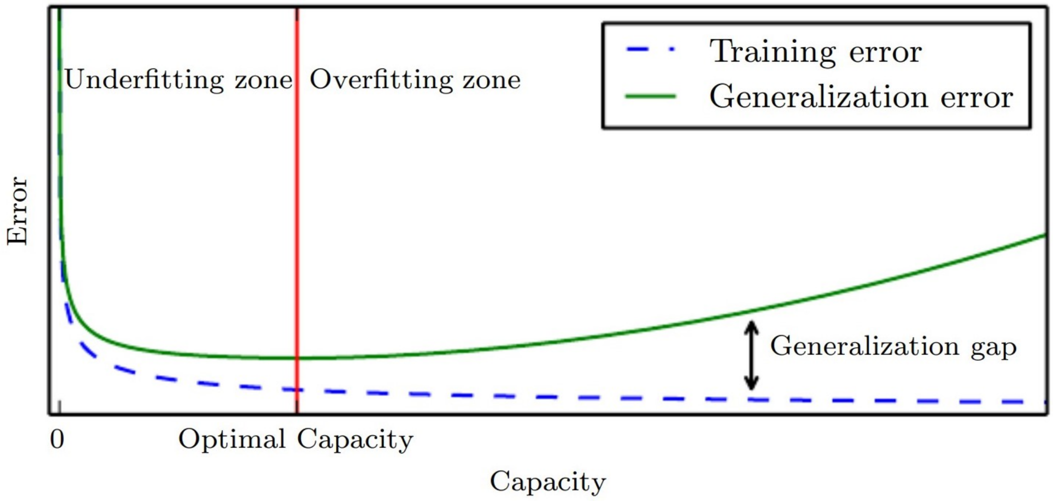

FIGURE 2.9: Illustration displaying the optimal learning capacity of a model. The generalization error is a measurement of the model performance on the validation set. Reproduced from [29].

Regularization techniques are additional methods for preventing overfitting in ML models, and involve setting specific hyperparameter values. A widely used regularization technique in deep learning is dropout. The key idea behind dropout is to randomly switch off a set of neurons in a layer during training. This is done according to a probability value specified by the user. The concept of dropout is illustrated in figure 2.10.

Here, the dropout probability is set to 0.5, meaning that half of the neurons in a layer are randomly deactivated during training. Outside of the training phase, however, all of the units are activated. The purpose of dropout is to

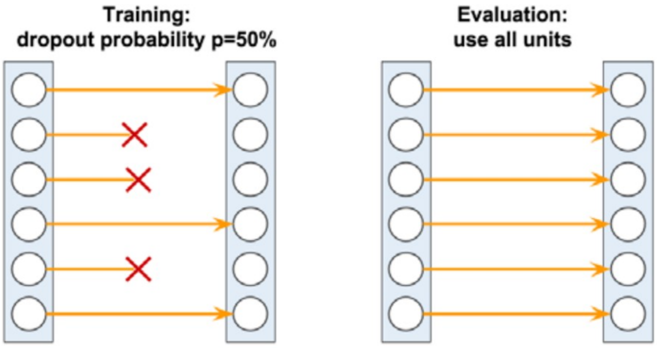

FIGURE 2.10: Illustration showing the concept of the dropout technique. During training, half of the units (nodes) are randomly deactivated. When the model is evaluated, all units are used. Reproduced from [26].

ensure that the model does not become too dependent on a single neuron. Each forward pass during training effectively uses a different thinned-out version of the network. This can be seen as a way of training a large ensemble of networks, with the average of this ensemble being utilized during network testing. Thus, dropout is a simple yet highly effective tool for improving the generalization capability of deep neural networks.

A common approach to splitting a dataset for developing an ML model is illustrated in figure 2.11. In this method, the dataset is divided into training and validation sets for model development, with a separate test set reserved for the final evaluation of model performance. However, a drawback of this approach is that the estimation of model performance can be greatly influenced by how the training dataset is split into training and validation subsets. Consequently, the performance estimate might vary significantly with different data partitions.

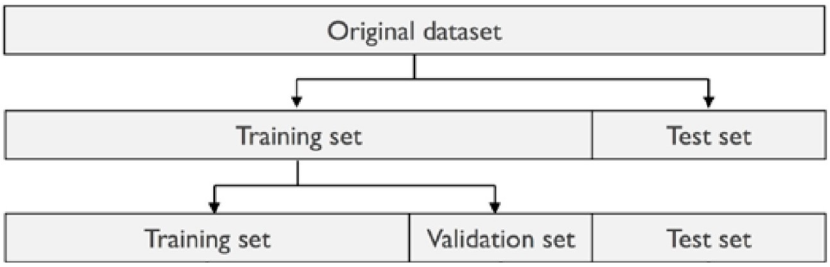

FIGURE 2.11: Illustration showing a common approach to splitting a dataset. Reproduced from [26].

In order to minimize the effects of randomness and variations in the data, methods employing cross-validation are used. In these methods, the data is divided into a number of partitions, ensuring that the model is trained and validated on all parts of the data. The concept is illustrated in figure 2.12. For each partition, a new model is trained and evaluated, with the final performance being calculated as the average of the performance of each model. As such, this approach is a robust way to evaluate a model. Empirical evidence suggests that a good standard value for the number of partitions is 10 [30].

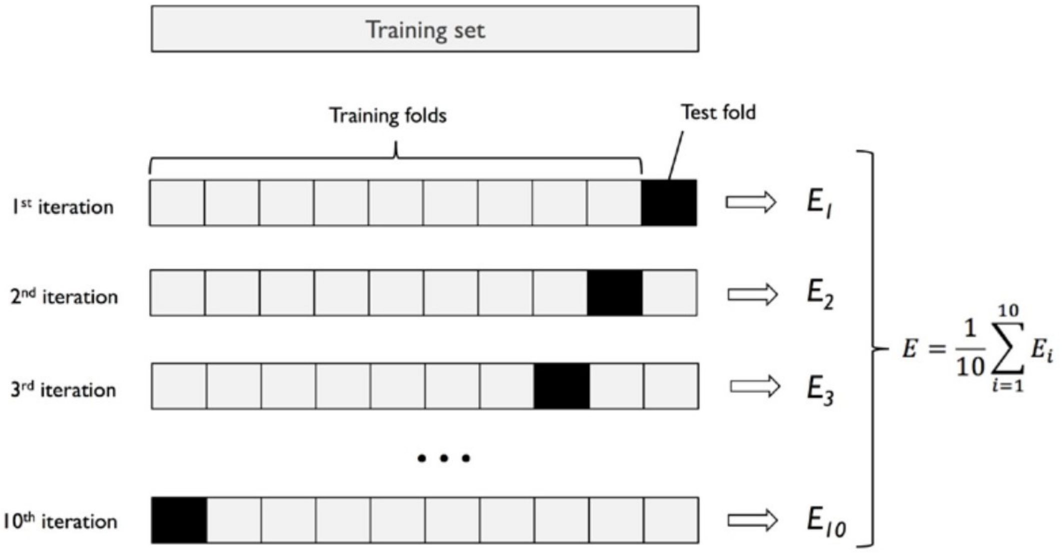

FIGURE 2.12: Illustration displaying the concept of cross-validation. For each of the 10 iterations, a new model is trained and evaluated. The final score is the average of all evaluations. Reproduced from [26].

2.2.6 Error Metrics

To assess the performance of a model, it is essential to quantify the error between its predictions and the actual values. There are various techniques for this quantification, known as error metrics. Each error metric has distinct characteristics and calculates the error in a different way. This section will present the error metrics that are utilized both for evaluating performance and as loss functions in this thesis.

The Mean Squared Error (MSE) is the average of the square of the error between the prediction and the true value, and is given by

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2, \tag{2.1}$$

where n is the number of observations, y_i is the true value for every observation i, and $\hat{y}_i$ is the predicted value. Since MSE will give values in the unit of y_i squared, a more intuitive score can be given by the Root Mean Squared Error (RMSE). RMSE is simply the root of MSE, and can be written as

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}. \tag{2.2}$$

Another error metric is the Mean Absolute Error (MAE). This score is calculated as the mean of the absolute value of the difference between the prediction and the true value, and can be calculated as

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|. \tag{2.3}$$

Further, the Mean Absolute Percentage Error (MAPE) is calculated as the average of the absolute differences between predicted and true values, expressed as a percentage of the true values. The calculation is given by

$$\text{MAPE} = \frac{100\%}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right|. \tag{2.4}$$

Following Trebbien [25], the Negative Log Likelihood (NLL) is included as a loss function to enable a model to predict both the mean and standard deviation of a Gaussian distribution. The calculation is given by

$$\text{NLL} = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{(y_i - \hat{y}_i)^2}{2\sigma_i^2} + \frac{1}{2}\ln(2\pi\sigma_i^2)\right), \tag{2.5}$$

where $\hat{y}_i$ is the predicted mean and σ_i is the predicted standard deviation.

2.3 Predicting Electricity Demand Using Machine Learning

Data-driven approaches for predicting electricity demand have recently attracted significant attention. Their ability to detect statistical patterns from available datasets, as opposed to relying on on-site physical information, offers a powerful advantage [6]. This section introduces a selection of articles that explore methods similar to those in this thesis, particularly focusing on short-term electricity demand prediction. Such predictions typically span a time horizon ranging from a few hours to several weeks [31].

Torres et al. [32] utilized a deep LSTM network to predict electricity demand for the Spanish power grid. They determined the optimal hyperparameters through a random search, complemented by a metaheuristic named the coronavirus optimization algorithm, inspired by the propagation patterns of the SARS-Cov-2 virus. Focusing on a 4-hour prediction horizon, their optimal model achieved notably low prediction errors, surpassing the performance of existing state-of-the-art methods.

Slowik and Urban [24] aimed at developing a universal forecasting tool for energy consumption, intended for enabling end-use consumers participating in the smart grid energy market. They developed an LSTM model for short-term energy demand prediction using data from a manufacturing plant. Their proposed model had a simple architecture of 1 LSTM layer and 128 units, making it suitable for computers with standard processing capabilities. This design effectively balanced accuracy with computational demands, resulting in low prediction errors and demonstrating a practical trade-off between precision and processing requirements.

Shao and Kim [33] proposed a novel deep LSTM model based on a multi-channel architecture, enabling their model to effectively process several factors in parallel. Their model, named Multi-Channel LSTM with Time Location (TL-MCLSTM), extracted information from power consumption, time location, and customer behavior to predict electricity demand multiple steps ahead. Utilizing two electric company datasets from Pennsylvania, New Jersey, and Maryland, they developed their model, which demonstrated prediction accuracy surpassing that of state-of-the-art models.

Hwang et al. [34] employed various machine learning methods, including LSTM, to predict electricity demand for 28 commercial buildings. Their approach integrated a data-driven methodology with the physical characteristics of the energy consumption of the buildings. This combination enhanced the predictive capabilities of their models.

Rafi et al. [31] created a method for short-term electricity demand prediction by integrating a convolutional neural network (CNN) with an LSTM network. While CNNs are often used for image recognition, they have shown promising results in time series analysis as well. Thus, [31] states that combining CNN and LSTM is a strategic choice for minimizing forecast errors. Their model, developed and validated using data from the Bangladesh power system, achieved higher prediction accuracy than other commonly used models, including a standard LSTM model.

Chapter 3

Case: Oslo Airport Gardermoen

Building on the work of Kvisberg [9], this thesis contributes to the NeX2G project by exploring machine learning algorithms aimed at accurately predicting peak electricity demand at OSL. While the LSTM models from [9] captured general demand patterns, they struggled with accurate peak predictions. This work aims at refining the LSTM models for better peak forecasting by employing two main strategies: training models with four different loss functions to assess their impact on performance, and implementing a comprehensive optimization method to identify the best model parameters. Since an extensive data exploration regarding the same case is already provided by [9], this part is kept brief in this thesis. Hence, this chapter concisely presents the relevant patterns and characteristics of the data used in this work.

3.1 Oslo Airport Gardermoen

OSL is the largest airport in Norway, normally serving more than 28 million passengers each year [35]. The location of OSL is in the municipalities Ullensaker and Nannestad in Viken county. Avinor owns and runs the airport, and supplies the data used in this thesis.

OSL, which includes two runways and a terminal building, also features a range of operational and administrative structures. The airport requires electricity for multiple purposes, including lighting, transportation, heating cables, air conditioning, and other technical facilities and appliances [9]. In this thesis, high-resolution electricity load measurements have been aggregated into hourly values. The data utilized here does not represent the entire airport but is derived from one of the main measurement points at OSL.

Avinor pays both electricity prices and grid fees for their electricity usage, where the latter includes consumption taxes, electric certificates, surcharges, and a power component [36]. The power component is determined on a monthly basis by the highest hourly power usage each month for the total electricity consumption of the entire facility at OSL [9]. Reducing the highest hourly power usage in a month can lead to cost savings.

3.2 Electricity Consumption Data

As the aim of the thesis is to predict the electricity demand, the patterns and characteristics of the measurements need to be examined. The data is a time series spanning 12431 hours of measurements, beginning on March 1, 2022, and concluding on July 31, 2023. Older data is available, but this start date is chosen to minimize disruptions caused by the Covid-19 pandemic. The following sections describe the relevant patterns seen across the data. Additional plots and statistical information regarding the electricity consumption time series can be found in Appendix A.

Figure 3.1 presents the entire electricity consumption time series from beginning to end as daily average values. Observing the series as a whole, clear seasonal patterns in electricity consumption are evident. The consumption reaches its peak in the winter, gradually decreases until summer, and starts rising again during autumn.

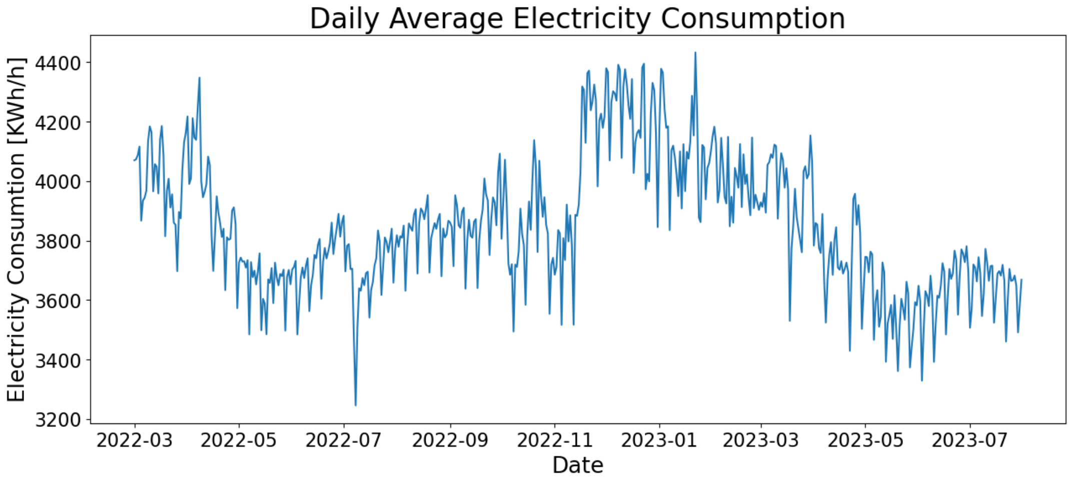

FIGURE 3.1: The daily average electricity consumption for the full time series.

Observing figure 3.2 reveals the average weekly pattern in the data set. Weekdays, i.e. Monday to Friday, exhibit similar consumption levels with two distinct daily peaks. Weekends, however, show a different pattern. Saturdays typically record the lowest overall consumption, while Sundays start similarly but experience a significant increase in consumption during the afternoon.

Figure 3.3 illustrates the average weekly electricity consumption for each season. Here, spring is defined as March until May, summer is June until August, autumn is September until November, and winter is December until February. This figure highlights seasonal variations, with winter weeks showing generally higher consumption levels and more pronounced peaks. The other seasons appear to be similar in consumption level, with autumn displaying the most pronounced peaks among them.

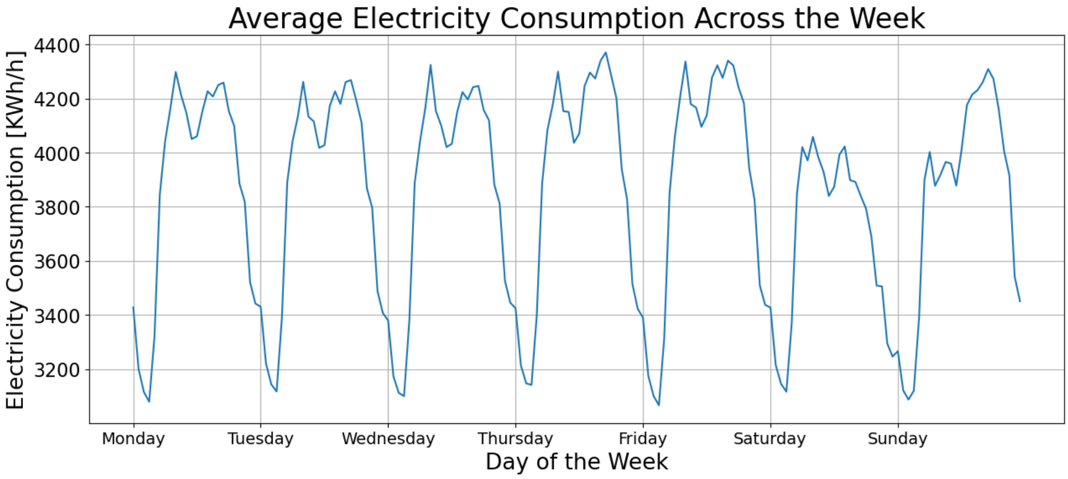

FIGURE 3.2: The average weekly electricity consumption pattern.

Figure 3.4 presents the average daily pattern of electricity consumption throughout the time series. It reveals a distinct morning peak at 8 a.m., while the timing of the second peak is more diffuse. Decomposing the average day into seasonal averages, as shown in figure 3.5, explains why that is. The first peak is consistent at 8 a.m. across all seasons, but the timing of the second peak changes. In spring and summer, it occurs at 2 p.m., while in autumn and winter, it shifts to 5 p.m. Conclusively, there are clear seasonal, weekly, and daily patterns in this data. In order to robustly predict the electricity demand, these components need to be captured successfully.

3.3 Other Variables

In this work, alongside electricity consumption data, several exogenous variables are utilized to provide additional information. Exogenous variables are external factors not directly related to the predicted variable. In this case, these include electricity price, outdoor temperature, and airport passenger number, all provided by Avinor. However, as detailed in section 4.1.1, the temperature data is substituted with data from the Norwegian Meteorological Institute [37] at the same location. Table 3.1 lists the variables used in this study. As indicated in the table, all variables have hourly time sampling, except for the passenger numbers, which are weekly sums. Chapter 4 describes how this disparity in time sampling is managed. While [9] used similar variables for their LSTM models, the inclusion of electricity price is new to this work. However, a significant issue was identified regarding this variable. It was discovered to actually be the electricity cost, which is a product of the electricity consumption. This error was unfortunately discovered too late for correction, and has inadvertently allowed the model to access information that would be unavailable in a practical application. The implications of this misstep are discussed in 5.3.3.

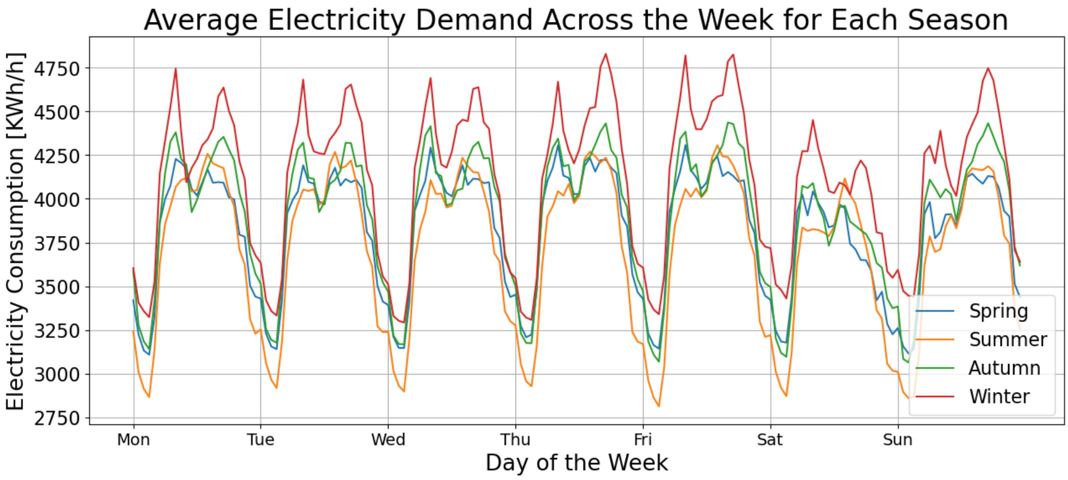

FIGURE 3.3: Seasonal decomposition of the average weekly electricity consumption pattern.

TABLE 3.1: The table shows the variables in the dataset and their time sampling.

Variable name	Time sampling
Electricity consumption [kWh/h]	Per hour
Electricity price [NOK/kWh]	Per hour
Passenger number	Weekly sum
Outdoor air temperature [°C]	Hourly average

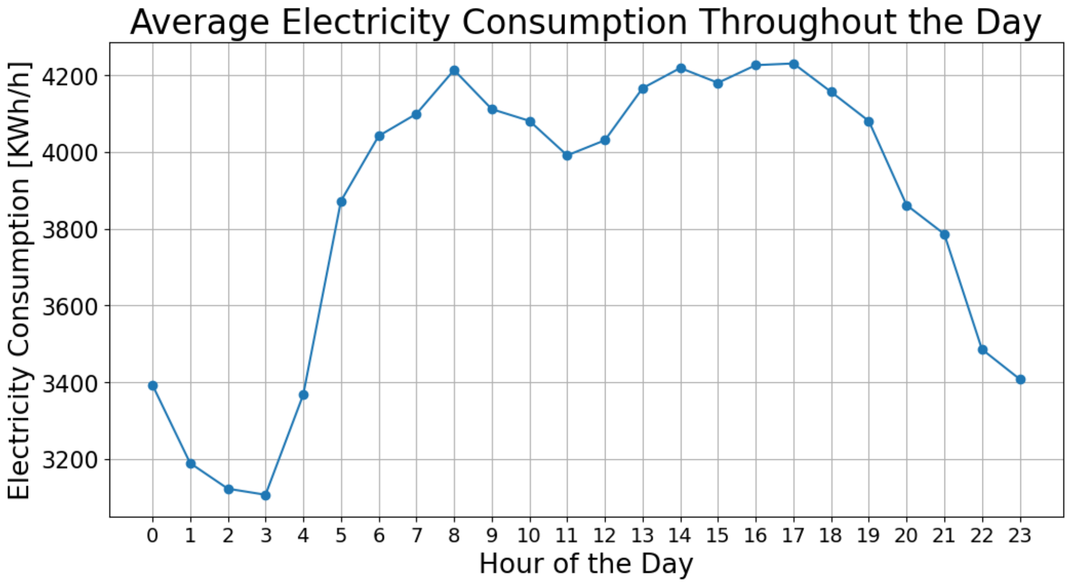

FIGURE 3.4: The average daily electricity consumption pattern.

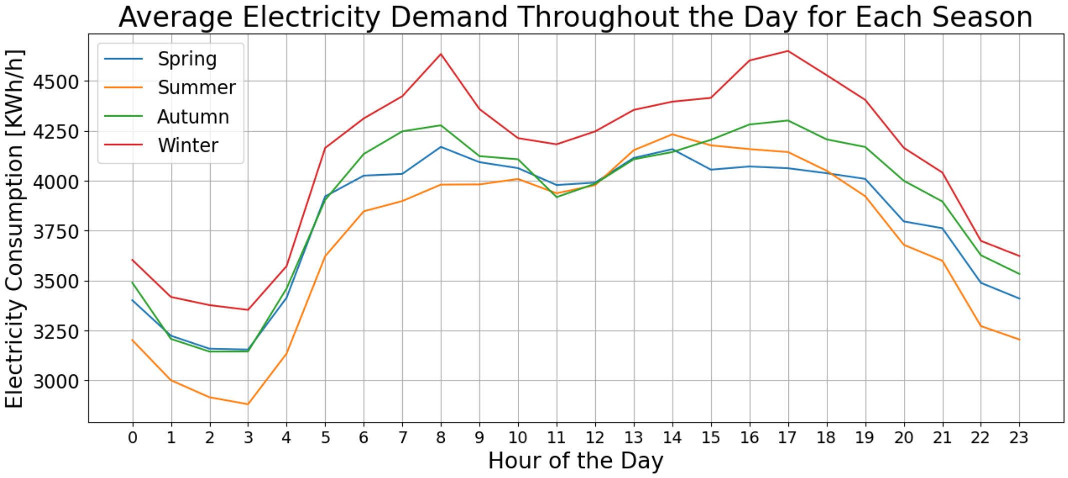

FIGURE 3.5: Seasonal decomposition of the average daily electricity consumption pattern.

Chapter 4

Methods

This chapter describes the methods used to create the LSTM models for predicting electricity demand at OSL. The process involves cleaning the data, manipulating variables, and preparing the data for the LSTM algorithm. To set a benchmark, a baseline model is established. A grid search is conducted for four different loss functions to identify the optimal model hyperparameters. This approach results in four distinct yet optimized models, which are then evaluated both empirically and through error metrics.

The data processing and machine learning programming implemented for this thesis use Python. For building the LSTM models, the ML library Tensorflow [38] is used with the interface Keras [39]. As described in Chapter 2, grid searches are computationally demanding. The Orion High Performance Computing Center at NMBU provides the necessary computational resources for these intensive tasks. For less demanding tasks, a Microsoft Surface Pro 7 with an Intel(R) Core(TM) i7-1065G7 CPU (1.30GHz base, 1.50 GHz max speed), 16.0 GB RAM, and Windows 10 Home has been used. The Python script employed for developing the models is made available through GitHub [40].

4.1 Data Preparation

As presented in Chapter 3, the data set is a multivariate time series consisting of hourly measurements of electricity consumption, electricity price, air temperature, and weekly passenger numbers. To address the difference in time sampling, each hour within a week is assigned the same corresponding weekly passenger number. Prior to using this data in the LSTM models, it requires cleaning and preparation. The subsequent section elaborates on the techniques and methods used for this process.

4.1.1 Data cleaning

Data cleaning is an important step in the ML process, as it enables the algorithm to learn the underlying patterns of the data without being disturbed by measurement errors and discrepancies. In the air temperature data, there are multiple constant-value intervals suggesting some kind of faulty measurement. To deal with this, the entire air temperature time series from Avinor is replaced with measurements from the Norwegian Meteorological Institute at the same location [37]. The new temperature series is devoid of damaged intervals, except for just a total of 5 non-consecutive missing values that are addressed using forward filling. The rest of the measurements in the data set are considered to be generally highly reliable, and no further cleaning is required.

4.1.2 Variable Manipulation

When preparing a data set for ML, variables can be transformed or created to provide the algorithm with additional information. A key method employed is the deconstruction of the date and time for each sample into new categorical variables. This is done to give the model a sense of time. These variables include the hour of the day, month number, and weekday number, which are then incorporated into the data set as one-hot encoded vectors. For example, Sunday is represented by the index value 6 and thus encoded as [0, 0, 0, 0, 0, 0, 1]. This one-hot encoding approach is critical to ensure the algorithm does not misinterpret the numerical values of the days, avoiding the incorrect assumption that a higher index value implies a greater day.

Additionally, following the approach used by Kvisberg [9], incorporating previous electricity consumption as an input variable is considered beneficial for the model. Consequently, each data sample is supplemented with information on electricity consumption from 24 hours prior. This decision is supported by the auto-correlation plot in figure A.5, which indicates that electricity consumption data from the same time on the previous day could be valuable for the predictions of the model. However, due to a coding error discovered late in the process, this time shift actually ended up being 48 hours, not 24. The potential implications of this discrepancy are discussed in section 5.3.3.

The electricity consumption for the current hour is extracted from the data set to serve as the target variable. This process results in a total of 78 features, with 74 related to one-hot encoded categorical variables. The remaining four features are the electricity price, air temperature, passenger number, and the electricity consumption 24 hours prior. This set of features, being a combination of the previous consumption, temporal variables, and exogenous variables are commonly used in state-of-the-art ML models seen in the literature [6].

4.1.3 Preprocessing for LSTM

The LSTM algorithm requires the data to be formatted in a specific way, necessitating some preprocessing. Firstly, the data is scaled using the MinMaxScaler [41] from the scikit-learn ML library. This process transforms all variables to values between 0 and 1, with 0 representing the minimum and 1 the maximum value of each variable. This scaling ensures that all variables are on an equal scale, allowing for their uniform interpretation by the LSTM algorithm.

The input data for the LSTM model must be shaped as [samples, time steps, features]. In this case, time steps denote the number of time points received by the algorithm in a single input. The data is reshaped to allow the LSTM to process the time series in 24-time-step windows, advancing the window by one time step for each input. This model follows a many-to-one approach, where 24 time steps of training features predict the target value at the final time step. Therefore, to forecast 24 hours of electricity demand, the model analyzes 24 separate samples, each containing a 24-hour feature sequence, and produces a prediction for each of these sequences. This is possible due to the day-ahead availability of both the electricity price and the outdoor air temperature, which are used as input in the model. The choice of a 24-hour sequence in the input is designed to capture the daily pattern in electricity demand, and similar decisions regarding this option are common in the literature [32] [33].

4.2 The models

4.2.1 Loss Functions

Four distinct loss functions are utilized for developing the LSTM models. Firstly, MSE is chosen for its common use in such models. Secondly, MAPE is selected, which may influence the training differently due to its error being percentage-based. Thirdly, NLL is used for its capability to predict both the mean and standard deviation of a Gaussian distribution. This adds a probabilistic dimension to the predictions, potentially increasing the practical applicability of the models. Finally, a modified loss function is proposed to specifically target the peaks. This function is designed as a weighted MSE (WMSE), and is given by

$$\mathrm{WMSE} = \frac{1}{N} \sum_{i=1}^{N} y_i \times (\hat{y}_i - y_i)^2. \tag{4.1}$$

Here, the squared difference between the true value y_i and the predicted value $\hat{y}_i$ is multiplied with y_i, thus increasing the error for large values of y_i. This function will therefore penalize the algorithm more heavily for incorrect predictions of high values, potentially increasing the prediction accuracy for the peaks. As the values of y_i are scaled and lie between 0 and 1, this function will automatically give more weight to hours with higher peak values.

4.2.2 Grid Search and Cross-validation

To determine the optimal hyperparameters for constructing the LSTM models, a comprehensive grid search is conducted. The hyperparameters under consideration are the number of hidden layers n, the number of units in each hidden layer u, and the dropout probability following each hidden layer d. Table 4.1 displays the values for each of the hyperparameters that are explored in the grid search. All 120 combinations of these values are tested for each of the four loss functions, resulting in a total of 480 model variants. Each specific variant is referred to as a configuration.

TABLE 4.1: Hyperparameter values tested in the grid search. These include the number of hidden layers n, the number of units per hidden layer u, and the dropout probability d following each hidden layer.

Hyperparameter	**Values**
n	$\{1, 2, 3, 4, 5\}$
u	$\{32, 64, 128, 256\}$
d	$\{0, 0.1, 0.2, 0.3, 0.4, 0.5\}$

To ensure statistical validity and to account for seasonal variations in the data set, a cross-validation method is employed. The data is partitioned into 10 folds, enabling each of the 480 model configurations to be tested on subsets across the entire data series. The choice of 10 folds is, as described in section 2.2.5, based on empirical evidence from [30]. In each fold, 80% of the dataset is designated for training, 10% for validation, and the remaining 10% for testing. The validation and test sets are consecutive 20% segments of the time series, progressively moving from the beginning to the end over the 10 folds. This method of splitting the data is based on common practices found in related literature.

The performance metrics used are RMSE, MSE, MAE, and MAPE, which are standard in related literature and allows for comparison between different research projects [32] [6]. Additionally, WMSE is used for model evaluation. Although it is not a real error metric, it can be useful to compare its value between different models. Given that WMSE provides values on a cubed scale, the cubic root WMSE (R3WMSE) is used for a more intuitive comparison between models, which is given by

$$\text{R3WMSE} = \sqrt[3]{\frac{1}{N}\sum_{i=1}^{N} y_i \times (\hat{y}_i - y_i)^2}. \tag{4.2}$$

Each of the 480 models is therefore tested on the 10 folds across the time series, and the final performance score of a model configuration is calculated as the average of its performance across all the folds. For each model, the average scores for all error metrics, along with the corresponding standard deviations, are calculated. Ultimately, the best-performing model configurations for each loss function are identified and evaluated.

Additionally, a naive baseline model is included for comparison. This baseline model predicts the electricity consumption throughout a day as the average consumption of the previous day, providing a basic benchmark for error comparison with LSTM models.

4.2.3 Other Decisions and Parameters

Besides the hyperparameters listed in table 4.1, there are several additional parameters that influence the LSTM model. These include the learning rate, which is managed by the Keras implementation of the Adam optimizer [42]. The optimizer coordinates the training algorithm of the model, and Adam is a standard choice. The learning rate starts at the default Adam value of 0.001 and is adjusted during training. Further, as described in Chapter 2, the number of epochs is crucial for preventing both overfitting and underfitting. The epochs are handled through early stopping, where the training process is stopped if the validation loss does not improve. In this case, if the validation loss does not improve for 100 epochs the training will stop and the model will revert back to its best performing point. Finally, the batch size is set to 64. The batch size determines the number of input samples that are passed to the model at once during training, and its value can have a significant impact on the training dynamics. As it is seen in the literature as a good starting point for batch size, 64 is chosen and kept static through this work. The parameters discussed in this section could also be managed as hyperparameters and optimized through a grid search. However, examining all of them in such detail is beyond the scope of this thesis.

Chapter 5

Results and discussion

In this chapter, the results of this work are presented and discussed. The performance of the models is evaluated using various error metrics, and the best-scoring models are tested on a subset of the data. Additionally, the results are compared to similar works in the literature. The significance and potential applications of the findings are also explored. Finally, the limitations of this thesis are acknowledged and discussed.

5.1 Grid Search Results

To determine the best model configuration for each loss function, a grid search with cross-validation is performed as described in section 4.2.2. The best-scoring model for each loss function in each metric is shown in table 5.1. In addition, the scores of the baseline model are included for reference.

TABLE 5.1: Table showing results from grid search and cross-validation for model optimization. Columns represent various loss functions used during training, while rows correspond to different error metrics. Each cell displays the average score and standard deviation for the best-performing model under each combination of loss function and error metric. Additionally, the number of hidden layers n, number of units u, and dropout probability d are provided for these models. Models achieving the lowest score for each error metric are highlighted in bold.

	Model-NLL	Model-MAPE	Model-MSE	Model-WMSE	Baseline
RMSE	248.43 ± 80.08 n=1, u=64, d=0.0	394.50 ± 156.48 n=1, u=128, d=0.3	**212.17 ± 38.51** **n=1, u=128, d=0.5**	213.95 ± 42.76 n=1, u=64, d=0.0	442.57
MSE	67488 ± 45147 n=1, u=64, d=0.0	170801 ± 67498 n=5, u=64, d=0.4	**46350 ± 15789** **n=1, u=128, d=0.5**	47420 ± 18939 n=1, u=64, d=0.0	195872
MAE	201.19 ± 65.22 n=1, u=64, d=0.0	299.64 ± 71.15 n=5, u=64, d=0.4	**171.95 ± 32.16** **n=1, u=128, d=0.5**	173.78 ± 36.59 n=1, u=64, d=0.0	365.95
MAPE	5.42 ± 1.98 n=1, u=64, d=0.0	8.13 ± 1.76 n=5, u=64, d=0.4	**4.53 ± 0.77** **n=1, u=128, d=0.5**	4.59 ± 0.76 n=1, u=64, d=0.0	10.10
R3WMSE	608.65 ± 123.99 n=1, u=64, d=0.0	830.86 ± 210.77 n=1, u=128, d=0.4	**555.38 ± 76.83** **n=1, u=128, d=0.5**	**557.86** ± 88.34 n=1, u=64, d=0.0	888.86

As seen in the table, all of the models outperform the baseline reference model, meaning that they at least perform better than using the average consumption of the previous day as the prediction. They all reproduce a sinusoidal pattern corresponding to the day-to-day electricity consumption cycle, as shown by figures 5.1, 5.2, and 5.3. Moreover, the models appear to recover the fundamental aspects of the data described in Chapter 3, such as the dual peak shape seen across a typical day and the characteristic patterns seen on weekend days. These observations indicate that the models generate sensible predictions of the electricity demand.

As all of the models are tested on 10 folds, meaning 10 iterations with the same starting conditions and covering the entire data set, the models giving the lowest average scores are the ones that perform best on the time series as a whole. This method ensures that the best-scoring models are the ones that best handle temporal variation and seasonality, and hence are the most generalized and robust. The results shown in 5.1 display that the MSE model with 1 hidden layer, 128 units, and 0.5 dropout probability achieves the lowest averages in all error metrics. For the WMSE and NLL loss functions, the best models both have 1 hidden layer, 64 units, and 0.0 dropout probability. The best MAPE model varies between three different configurations depending on the error metric. From this point, the models representing each of the loss functions will be referred to as Model-MSE, Model-WMSE, Model-NLL, and Model-MAPE.

These results have a statistical nature and must be analyzed accordingly. For one model to be better than another, there must be a statistically significant difference. For instance, the error scores of Model-WMSE are consistently higher than those of Model-MSE across all metrics. However, since the average scores of Model-WMSE fall well within one standard deviation of the mean scores of Model-MSE, this difference is not significant, and their performances can be considered equivalent. This is based on the reasoning that the standard deviations represent the uncertainty in calculating the mean values. Therefore, when the average scores of Model-WMSE are within the uncertainty intervals of the Model-MSE scores, there is not enough statistical evidence to claim that one model is better than the other. Comparing Model-MSE to Model-NLL, the average error scores of the latter are higher but still within one standard deviation of the Model-MSE averages for all error metrics, except for MSE and MAPE. This suggests some evidence that Model-NLL might perform worse than Model-MSE and Model-WMSE, although their performances could still be considered equivalent. In the case of Model-MAPE, however, the error averages lie outside one standard deviation of the mean values of Model-MSE for all metrics. Consequently, the error metric performance of Model-MAPE is considered significantly worse than the other models.

The three models Model-MSE, Model-WMSE, and Model-NLL are thus statistically similar in their error metric performance, and they are similar in their architecture as well. They all have 1 hidden layer, and either 64 or 128 units. The dropout probabilities are either 0.0 or 0.5, but these values affect the model training and not the actual structure. Considering the possible combinations from table 4.1, these architectures are relatively simple. Given the thoroughness of the grid search and cross-validation used to evaluate the models, this observation suggests that for this specific problem, a simpler model tends to perform better.

To evaluate the performance of these models relative to all models included in the grid search, the distribution statistics of the grid search results are presented in table 5.2. A comparison of these statistics with the results in table 5.1 provides context for the best models within the entire grid search. Comparing the lowest scores with the bottom 25% of the distribution indicates that a significant portion of the models perform close to the best ones. For instance, looking at the RMSE statistics, 25% of the models have errors lower than 269.85, and 50% have errors lower than 310.03. Given that the lowest RMSE score, achieved by Model-MSE, is 212.17 with a standard deviation of 38.51, it is likely that a significant amount of models have scores that are similar.

TABLE 5.2: Table presenting the distribution statistics for all models evaluated during the grid search, categorized by each error metric.

	RMSE	MSE	MAE	MAPE	R3WMSE
Median	310.03	105595	257.54	6.81	706.34
Minimum	212.17	46351	171.95	4.53	555.38
Maximum	1188.30	6019255	987.47	25.35	1735.00
25%	269.85	79507	221.79	5.90	644.05
75%	428.27	205188	332.32	8.78	873.34

To reflect on why the best-performing models have simple architectures, the descriptions of model complexity in section 2.2.5 are relevant. The optimal complexity of a machine learning model for a specific task is one that neither overfits nor underfits the data. In this work, a comprehensive grid search and cross-validation have been employed to optimize model complexity for predicting electricity demand at OSL. If these techniques are successful, it suggests that the optimal model complexity for this task is relatively simple, indicating that the patterns in the data are not overly complex. Furthermore, a simpler model might be more effective for generalized predictions across various seasonal and temporal variations compared to a more complex model. A comprehensive analysis of how model complexity affects performance is possible, but it is considered to be outside the scope of this thesis.

As already noted, the optimal configuration of Model-MAPE shifts between three variants depending on the error metric. This is different from the other three models, which all have a single configuration performing best on all metrics. This observation, along with the significantly worse performance compared to the others, implies that Model-MAPE is unable to learn the patterns of the data in a stable manner. Moreover, when considering the uncertainty intervals represented by the standard deviations of the error metrics, the performance of Model-MAPE does not show a significant improvement over the baseline model, which has a very simple and naive design. This comparison highlights the poor performance by Model-MAPE.

5.2 Model Testing

In this section, the four model configurations selected from table 5.1 are tested and evaluated. Using the first 80% of the data set for training, the next 10% for validation, and the final 10% for testing, the four models are trained as described in Chapter 4. The test set contains seven full weeks of electricity consumption during the summer of 2023. To display the different behaviors of the models, the predictions on week 7 of the test period are displayed. This is not necessarily the best nor worst week for any of the models, but it is chosen as the target week as it displays the trends seen across the full period.

In the following sections, the predictions of Model-MSE, Model-WMSE, and Model-NLL on week 7 of the test period are displayed. The predictions of Model-MAPE are left out as its performance is highly underwhelming compared to the other models. Upon observing the prediction plots, it is clear that this model is unable to learn the complexities of the data. Unlike the other models, its prediction pattern is softly shaped with no sharp corners and is thus completely missing the dual-peak characteristic of the daily consumption pattern. The Model-MAPE prediction plot for the target week can be found in Appendix B.

5.2.1 Model-MSE

The predictions of Model-MSE on the targeted week 7 are shown in figure 5.1. As seen here, the predictions are in general following the actual consumption closely, but are for several days unable to fully reach the large afternoon peak. This observation is largely true for the entire test period, but the peaks are generally underestimating more than what is seen in week 7. Model-MSE appears to capture the overall patterns more accurately than the other models, but often fails to reach the large peaks. Moreover, the peaks are rarely over-estimated. The low-points of the electricity consumption are generally not predicted correctly, but as these are not the focal point of this thesis they will not be studied as closely as the peaks.

5.2.2 Model-WMSE

Figure 5.2 shows the predictions from Model-WMSE on the target week. Compared to Model-MSE, this model clearly predicts larger values for the peaks. This observation indicates that the WMSE loss function can indeed affect the peak prediction of a model. The results do, however, show that

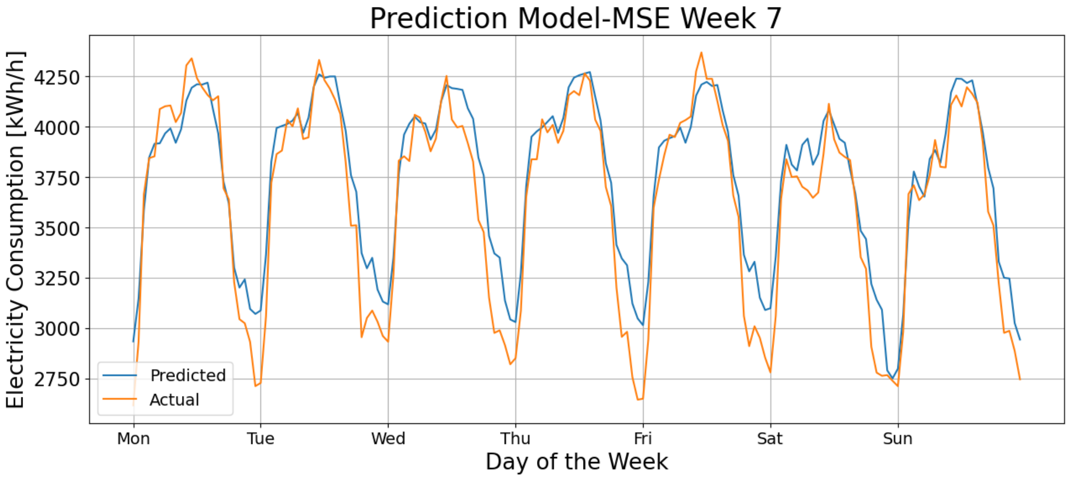

FIGURE 5.1: Predictions for week 7 in the test set using Model-MSE.

the predictions of Model-WMSE have a tendency to be overestimated. The peaks of the actual values in 5.2 are generally overlapped by the model forecast, but the predictions are higher than the actual values on multiple occasions. Although this is not consistent throughout the entire seven-week test period, the general trend shows that Model-WMSE tends to predict larger values and simpler patterns, which are more likely to overestimate the demand compared to Model-MSE. The patterns are less accurate, but the predicted peaks are closer to the peaks of the actual values. To determine which of the two models provides the best predictions, it is necessary to consider whether overestimating or underestimating the demand is more detrimental.

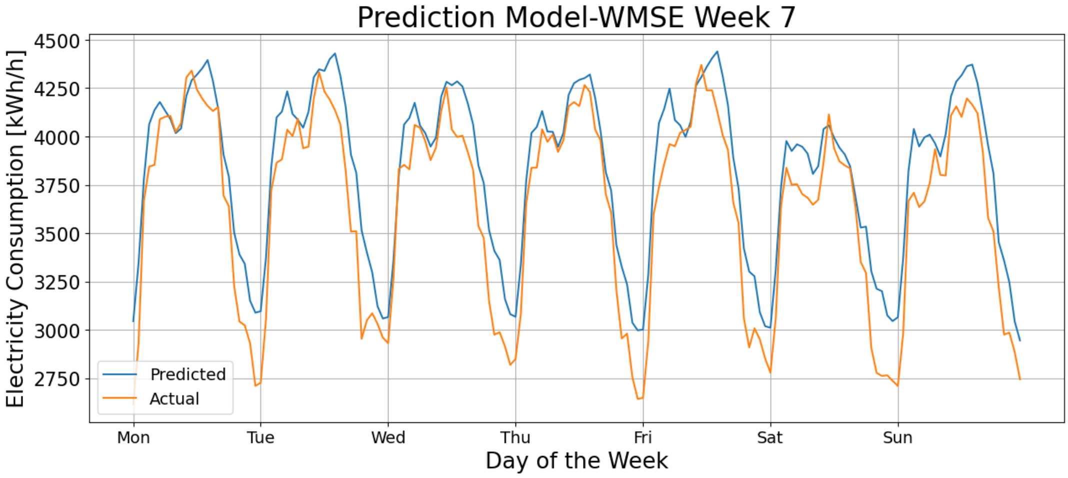

FIGURE 5.2: Predictions for week 7 in the test set using Model-WMSE.

5.2.3 Model-NLL

In figure 5.3, the week 7 forecast from Model-NLL is shown. As the NLL loss function enables the model to predict the mean and standard deviation of a Gaussian distribution, the prediction of Model-NLL contains more information than the other models. In figure 5.3, this prediction is illustrated by three graphs. One line for the mean value, and two lines indicating one full standard deviation above and below the mean. In a Gaussian distribution, about 68% of values lie within one standard deviation of the mean [43]. Consequently, Model-NLL predicts there is a 68% probability that the actual value falls within the dotted lines in figure 5.3, with the most likely value being the mean. Thus, the model can indicate the uncertainty of its own prediction, where the size of the standard deviation reflects the level of uncertainty. Observations from week 7 show that the actual values are mostly within one standard deviation of the mean, but the predicted mean values often underestimate the actual electricity consumption. Similar observations are noted throughout the rest of the test period. Therefore, although the predictions of Model-NLL seem to be less accurate compared to Model-MSE and Model-WMSE, they provide additional information that could be useful in practical applications.

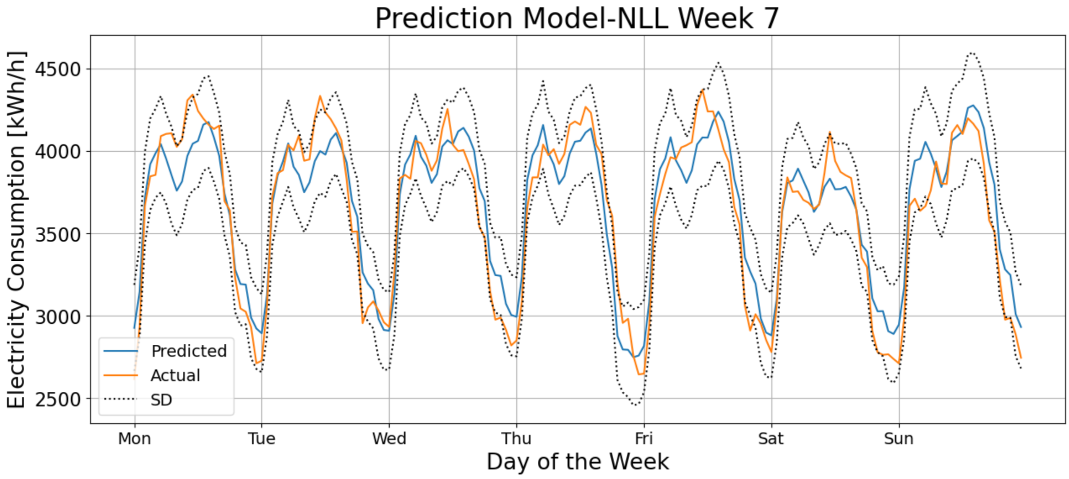

FIGURE 5.3: Predictions for week 7 in the test set using Model-NLL.

5.2.4 Model-MSE vs. Model-WMSE

In this section, the performance of Model-MSE and Model-WMSE are compared. These two models are chosen since they have the lowest average error scores according to table 5.1, and because they can potentially display the effects of introducing a weighted loss function. As described in Chapter 4, the WMSE loss function is a modification of MSE. This modification is studied by comparing the two models. Figure 5.4 shows the average predicted days for both Model-MSE and Model-WMSE in comparison to the true average day across the entire 7-week test period.

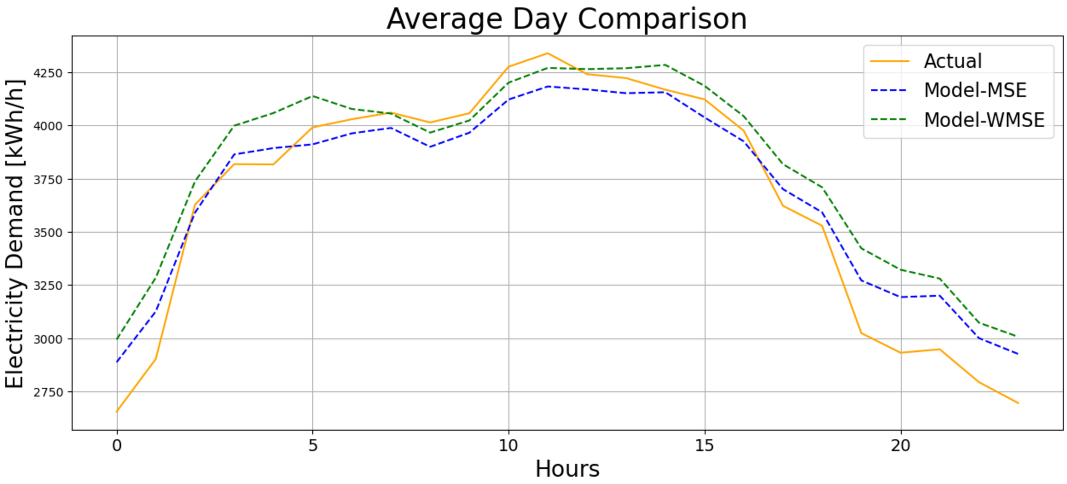

FIGURE 5.4: Comparison of the average predicted days for Model-MSE and Model-WMSE for the full test set

Figure 5.4 shows that, on average, Model-WMSE predicts higher values than Model-MSE, particularly at the peaks. However, as these graphs represent averages over the entire test period, they do not capture the complete picture. The objective is to predict the electricity demand on a day-to-day basis, not as an average over an extended period. Nevertheless, Figure 5.4 does suggest that introducing a weighted loss function can have a substantial impact on the prediction patterns

Figure 5.5 presents the best and worst prediction days for Model-MSE and Model-WMSE, employing a method similar to that of [32] in their evaluation of an LSTM model. The predictions for the entire test period are divided into individual days, with MAE calculated to determine the most and least accurate day predictions. On their best days, both models achieve accurate predictions that closely match the actual values. However, Model-WMSE more accurately predicts peak values than Model-MSE. On the worst days, the differences between the models are more pronounced. Model-MSE fails to capture the actual pattern, significantly underestimating peak values. By contrast, Model-WMSE overestimates substantially but more effectively captures the demand pattern, especially at peak times. It is important to consider which type of error is more critical in practical applications.

5.3 General Discussion

5.3.1 Comparison to Similar Literature

In this section, the results from this work are compared to similar research in the literature. The comparison is done using the MAPE metric when

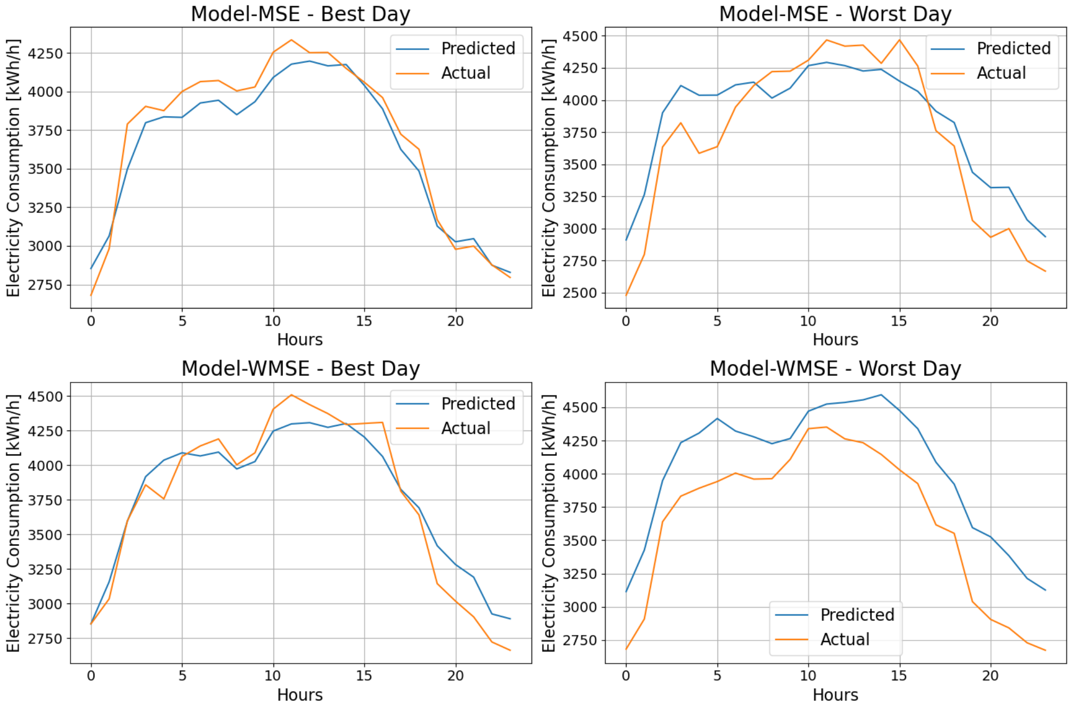

FIGURE 5.5: The best and worst predicted days for Model-MSE and Model-WMSE, as measured the MAE metric.

available, as it provides a scale-independent metric. Relevant differences in methodology are discussed.

Torres et al. [32] achieved MAPE scores under 1.5% for their LSTM models, notably lower than the 4.53% by Model-MSE in this study, as shown in table 5.1. A key advantage for Torres et al. lies in their data quantity: 9 years at 10-minute intervals, compared to the approximately 1.5 years at hourly intervals used here. This substantial difference suggests that more extensive training data can significantly enhance model performance. The finer resolution of their data might also contribute to the improved pattern recognition of the model. Their data, covering the entire electricity consumption of Spain, is also on a much larger scale, potentially reducing the impact of random noise on the model. Furthermore, Torres et al. employ a more intricate hyperparameter search technique, allowing different numbers of units and dropout rates for each hidden layer and including learning rate in their search. Despite using a random search rather than a full grid search, this approach enables them to explore more complex model architectures, potentially leading to a more finely tuned model. Their model makes predictions using only the previous consumption as input, suggesting that exogenous variables are not necessarily required for accurate forecasting. Their model used 168 time steps in the input sequences, corresponding to 28 hours, which is 4 hours more than in this work.

Shao and Kim [33] predicted the electricity demand 12 hours ahead for regions in Pennsylvania, New Jersey, and Maryland. Their TL-MCLSTM model, a variant of an LSTM model, achieved an average MAPE score of 3.13%, which is better than the results of this work. Notably, their model only utilized one LSTM layer with 20 units, and no hyperparameter optimization was performed. These high-performing results along with the simple LSTM architecture suggest that the multi-channel structure of their method allowed the model to efficiently learn several aspects of the data in parallel in order to make accurate predictions. In this way, their model was able to efficiently utilize power consumption data, time location, and consumer behavior. This could indicate that there are more efficient ways of handling exogenous variables than what is done in this thesis. Further, they use two data sets covering 7 and 14 years, which is considerably longer than in this thesis. The time sampling of the data used was hourly measurements, similar to this work, but as [32] they are on a much larger scale. Like this work, they also utilized 24-hour input sequences.

Rafi et al. [31] employed a CNN-LSTM fusion model to predict the electricity demand for the Bangladesh power grid. Predicting 24-hour intervals, their model achieved an average MAPE score of 3.22%, which is 1.31% better than the best result in this work. Their dataset spans nearly 6 years with a half-hour sampling interval. Instead of conducting a hyperparameter optimization routine, they focused on evaluating the impact of integrating a CNN into the model. Their findings demonstrate that the inclusion of a CNN enhances performance compared to a standard LSTM model, which achieved a MAPE of 7.55%. The specifics of this LSTM architecture are not detailed, but it is still outperformed by the models in this thesis, indicating the effectiveness of the methods used here in improving predictive accuracy. Unlike in this study, Rafi et al. did not use any exogenous variables in their model.

Slowik and Urban [24] predicted the electricity demand for a manufacturing plant with a 4-hour prediction horizon. They used a simple LSTM with 1 layer and 128 units, and achieved an MAE score of 0.0464 W. This error value shows a very high prediction accuracy considering that the median value in the consumption data is 82668.418 W. There are, however, major differences between their approach and the work in this thesis. For instance, their data set contains electricity consumption measurements over 24 hours with a time sampling of 10 seconds, adding up to 8640 data points. This is a much higher resolution than what is used in this work, which likely is beneficial for more accurate predictions. In addition, the nature of the data is different than OSL, as they are taken from a factory which exhibits different consumption patterns. They also use no exogenous variables, unlike this work. Three different network architectures were tested, and the LSTM with 1 layer and 128 units performed best.

5.3.2 Implications and Applications

To assess the implications of the results in this work, it is important to consider the specific application within the NeX2G project. The models developed are intended for predicting significant future peaks in electricity demand at OSL, enabling the activation of flexible energy resources to flatten these peaks, as noted in [8]. The primary goal for the consumer in this context is to reduce costs by flattening consumption peaks, making peak detection a critical aspect of the model. In this scenario, underestimating a peak is more harmful than overestimating. Therefore, Model-WMSE might be better suited for this particular challenge due to its tendency to predict higher peak values.

Successfully predicting peaks is the most crucial function of the model, but accurately forecasting general consumption patterns is also highly valuable. Flexible resources like water heaters and V2G systems require careful planning for smooth operation without compromising usage and comfort. Therefore, precise estimates of overall consumption patterns are still essential for effectively shifting electrical loads over time without impacting user experience.

The results from Model-NLL highlight the potential for predicting the mean and standard deviation of a Gaussian distribution. As previously mentioned, this feature allows the model to estimate its own prediction uncertainty, enhancing trust in practical applications and aiding consumers in making better-informed decisions. For example, high uncertainty in peak predictions could lead to a more conservative approach, such as reserving additional energy capacity for larger-than-expected peaks. Conversely, a peak prediction with low uncertainty is likely to be more reliable. This additional information could be highly beneficial for efficiently managing flexible resources.

The use of probabilistic loss functions like NLL in LSTM models, while not common in literature, shows potential. Wang et al. [44] developed a probabilistic load forecasting model utilizing a pinball loss-guided LSTM. This approach demonstrated high performance in forecasting for both residential and commercial buildings, indicating the effectiveness of probabilistic models in load forecasting scenarios. The use of NLL in this study is inspired by the work of Trebbien [25], who employed this loss function in an LSTM model for predicting electricity prices and achieved high-performance results. The probabilistic dimension introduced by using NLL as a loss function was found to be highly beneficial for the practical applications of their model.

The results from Model-WMSE and Model-NLL suggest that using a weighted loss function and a loss function that predicts the mean and standard deviation of a Gaussian distribution could both be advantageous for predicting electricity demand peaks. Model-WMSE more effectively captures peaks compared to other models, while Model-NLL provides additional information useful for managing peaks. Combining these two concepts into one loss function could potentially offer the benefits of both in a single model. Exploring this new combined loss function, however, falls outside the scope of this thesis and is recommended for future research.

Since the electricity cost paid by customers in Norway is influenced by the largest power peak, as described in section 2.1.3, consumers have an economic incentive to identify large future peaks and apply strategies like load shifting for peak reduction. Robust and accurate predictions are essential for identifying these peaks. Therefore, the use of predictive models, like those developed in this work, could be highly beneficial for consumers looking to reduce their electricity costs.

The LSTM models developed in this work have potential applications beyond just OSL. Any energy-consuming system looking to flatten its electricity consumption peaks and offer implicit flexibility to the grid, through the use of flexible energy resources, would benefit from an accurate demand forecast that successfully predicts peaks. Since these models are developed using data from a large commercial building like OSL, the methodology is most relevant to similar settings, including other large commercial buildings such as shopping malls, universities, and other airports. As highlighted in Chapter 2, buildings hold significant flexibility potential, and robust electricity demand predictions using ML models are central for unlocking this potential.

The main strength of the methodology in this work lies in its applicability to a wide range of systems similar in purpose and characteristics to OSL. It ensures that the hyperparameters of the LSTM model are tuned to each specific system, which might require different model structures than those used for OSL. Furthermore, the cross-validation technique ensures that the models are robustly evaluated across seasonal and temporal variations, favoring those with the lowest average error scores. This approach verifies that the models can handle diverse temporal conditions. Consequently, this methodology offers a versatile and robust framework for building LSTM models to predict electricity demand for various purposes.

Building on this, the methods presented in this thesis have demonstrated their effectiveness in systems like OSL, characterized by distinct cyclical consumption patterns. While there are variations between weekdays and seasons, most days at OSL follow regular patterns. However, in scenarios where consumption patterns are irregular and non-stationary, such as in hotels, a different approach may be required. This is illustrated in the work of Chen et al. [45], where they developed a clustering-based hybrid method combining fuzzy c-means (FCM) and support vector regression (SVR) for predicting hotel electricity demand. These consumption patterns are more complex and irregular, necessitating methods equipped to handle such variability.

A relevant consideration is the accessibility of the methods used in this study. The comprehensive grid search model optimization routine carried out here requires substantial computing resources. These were provided by the Orion High Performance Computing Center at the Norwegian University of Life

Sciences, as employing such intensive techniques on standard computing devices is not viable. However, the need for high-performance computing diminishes once the models are optimized and trained. A trained LSTM model for an electric system can be effectively operated on standard devices without the need for extensive computing power. The need to re-train the model arises only if there are significant changes in the physical conditions of the system that lead to a notable change in the patterns of electricity consumption. If the consumption patterns of the system are cyclical and stable, the LSTM model would not require frequent re-training.

5.3.3 Limitations

In the development of LSTM models for predicting electricity demand, it is important to recognize and address several limitations to improve their robustness and accuracy. The results from this work show that each model has weaknesses that could be addressed through further optimization efforts.

A major limitation in this work is the confusion between electricity price and cost. The models are intended to use the hourly electricity price for the next day, a readily available metric, as a feature. However, they actually use electricity cost, which is dependent on consumption and cannot be known in advance. The specific details of how this variable is calculated are unknown to the author, as they are based on a confidential electricity deal regarding Avinor, but it is known to include consumption data. This means the models inadvertently have access to information about the demand they are trying to predict, potentially influencing their accuracy. The exact impact of this error on the results remains unclear, but it raises concerns about the predictive integrity of the models.

Another limitation is that due to a coding error, the models currently use electricity consumption data from 48 hours ago, instead of the intended 24 hours. The impact of this error depends on whether the electricity consumption at a given time is more correlated with its value from 24 or 48 hours earlier. This can be assessed using the auto-correlation function, as depicted in figure A.5. The figure indicates that electricity consumption is highly correlated with its value at the same time each day, but this correlation diminishes over several days. This suggests that the data from 24 hours earlier might be more beneficial for the model than data from 48 hours ago.

As identified in section 5.3.1, this study uses considerably less data than similar research, representing a notable limitation. Despite this, the performance achieved here, while not as high as in these other studies, is still comparable. This suggests that the methodology used in this work holds promising potential, especially if applied to larger data sets similar to those in other research.

The quality of the input data is critical to the performance of ML models, which warrants greater emphasis in this work. Data cleaning, noise reduction, and outlier detection are vital preprocessing steps. Given the limited size of the dataset used here, these steps become even more important. Notable outliers in the electricity consumption data, as seen in figure A.1, can disrupt model training and performance. Addressing these outliers using interpolation techniques can help maintain data integrity. Techniques like wavelet transforms, as utilized in related studies by [46] and [47], have proven effective in reducing data noise. Implementing such methods could significantly enhance data quality, thereby potentially improving the predictive abilities of the models.

The structural formulation of the dataset also heavily impacts the performance of the model. The size of the lookback window, which determines the number of past observations considered for predicting future values, directly influences the number of weights in the LSTM model. This is due to the fact that each time step in the network is associated with its own hidden layers and weights, as described in Chapter 2. Thus, the lookback window size is a key factor for the model complexity and must be carefully considered when developing an LSTM model. Furthermore, the size of the prediction window, or how many hours ahead the model forecasts from a single input sequence, also influences the performance of the model. The size of the prediction window determines the size of the model output, which influences how the LSTM algorithm responds to the input data. Throughout this work, a 24-hour rolling lookback window with hourly predictions has been used, but further variation testing is recommended for optimal model development.

Furthermore, batch size, learning rate, and the number of epochs are critical parameters that should ideally be optimized. As detailed in Chapter 4, this work employs a constant batch size of 64, utilizes the Adam optimizer to set the learning rate, and implements early stopping to manage the epochs. Optimizing these parameters as hyperparameters, through a grid search like the one conducted for hidden layers, units, and dropout probability, could potentially result in more finely tuned models.

Another potential limitation arises from using multiple loss functions under the same conditions, which might not be equally effective across all functions. For instance, the underperformance of the Model-MAPE compared to others suggests that a single setting may not be universally optimal. Different loss functions may require different settings for batch size, learning rate, and epochs to function most effectively.

Concerning the cross-validation, the approach used in this work deviates from the standard preference for time series analysis. Typically, training data for time series is kept as consecutive measurements to maintain the temporal sequence. However, due to data constraints, all available data is used for training in each fold. As a result, when validating on an earlier part of the series, the model has already been exposed to subsequent data, which can potentially introduce look-ahead bias and is not standard practice. In this work, maintaining a sufficient amount of training data for all folds was seen as more important than avoiding look-ahead bias.

Furthermore, while the hyperparameter values listed in table 4.1 have been tested, it is possible to explore a more comprehensive range of values. Doing so may lead to hyperparameters that are more optimally tuned than those presented in this work.

Although a full grid search is thorough, it may not always be necessary due to its computational intensity. A random search could be a more efficient alternative, offering the potential to achieve comparable results while conserving resources. This approach could allow the exploration of more complex model architectures, as demonstrated in [32]. This is not a limitation of this work, but provides a guideline to reduce the training time in the future.

In this work, categorical variables are represented as one-hot encoded vectors, leading to a high-dimensional feature space. These vectors are sparse, typically filled with zeros and a single one. This 'curse of dimensionality' can hinder neural network performance, as noted by [26]. Embedding layers offer a solution by condensing information into a lower-dimensional space. However, they introduce additional parameters that require optimization, which was outside the scope of this thesis.

In the data, passenger number data is sampled weekly, limiting the amount of information it can contribute. In contrast, other variables are recorded at an hourly rate. For optimal model performance and to align with the temporal resolution of other data, it would be ideal for passenger numbers to also be available at an hourly resolution.

Addressing these limitations in future work could lead to more refined models and more accurate predictions of electricity demand.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

This thesis aimed to investigate the potential for predicting electricity demand at Oslo Airport Gardermoen (OSL), particularly focusing on accurate peak prediction. Following the work of a previous master thesis at NMBU [9], LSTM algorithms were used to develop ML models to predict the power demand 24 hours ahead. The models were trained using electricity consumption data from OSL for the years 2022 and 2023. Besides electricity consumption measurements, exogenous variables like electricity price, outdoor air temperature, and passenger numbers at the airport were used by the models. The results demonstrated that these models could provide accurate forecasts for future electricity demand, showing promising improvement in peak prediction.

To improve peak prediction capabilities specifically, two main strategies were employed. First, the models were trained using four different loss functions: MSE, MAPE, NLL, and a new proposed weighted MSE loss function. Second, a comprehensive grid search was performed to determine the optimal hyperparameters for the models. Cross-validation was used to robustly evaluate model performance across the entire time series. The models with the lowest average error scores from the cross-validation were deemed the best for each loss function.

The best-performing model, trained using the MSE loss function, achieved a MAPE score of 4.53%. Models trained with WMSE and NLL produced comparable scores of 4.59% and 5.42%, respectively. However, the model trained with MAPE performed significantly worse, with a score of 8.13%.

The results displayed promising results regarding the use of NLL and WMSE as loss functions. WMSE, designed to focus more on peak hours, predicted larger peak values compared to MSE, resulting in more reliable peak predictions, at the cost of slightly reduced accuracy. This resulted in a more reliable prediction of the large peaks at the expense of slightly less accurate predictions. Moreover, NLL added a probabilistic dimension to the predictions. While less accurate than the models trained with MSE and WMSE, the NLL model demonstrated the ability to estimate its own prediction uncertainty, providing additional practical value.

The grid search optimization revealed that the best models all had relatively simple architectures, with 1 hidden layer and either 64 or 128 units. This suggested that simpler models could provide the most generalizable predictions across seasonal and temporal variations. This observation implied that the patterns of the data are not overly complex, and a simple model is more fitting for this particular problem.

A main strength of this thesis lies in the methodology, which offers a robust and versatile framework for developing ML models to predict electricity demand. This approach can be applied to a variety of energy systems similar to OSL, enabling the development of optimized LSTM prediction models made for each particular system. These models are designed to perform well across various temporal and seasonal variations.

The models developed in this thesis show promising potential for predicting electricity demand patterns and peaks. Such models could assist various consumers like OSL in identifying future power peaks, thereby enabling efficient management of flexible resources such as electric vehicles or heating systems. By employing peak-reducing strategies, consumers can lower their largest power peaks and consequently reduce electricity costs, as costs are influenced by peak power usage.

For electric power systems as a whole, implementing such strategies could reduce peak demand. Since the power grid must handle peak demand, these strategies could enable more efficient use of existing grid resources.

6.2 Future work

While the LSTM models developed in this thesis produced promising results, they were not performing as well as some other models seen in the literature. Therefore, there are aspects of this work that can be improved to further enhance the applicability of the models in predicting electricity demands for facilities such as OSL.

A notable limiting factor of this work when compared to the literature is the limited amount of data used. While other research articles use multiple years of measurements, this work utilized about 1.5 years of hourly measurements, primarily to avoid disruptions arising from the Covid-19 pandemic. Utilizing a more extensive dataset could likely enhance the quality of the model training.

In future work, more emphasis could be placed on data cleaning and noise reduction. Enhancing the quality of training data could significantly improve the performance of the models. For instance, techniques such as wavelet transforms have been reported in the literature as effective for reducing noise in measurements.

The concept of utilizing weighted loss functions can be more extensively investigated in future work. The WMSE loss function utilized in this thesis is a simple design to explore the concept. More sophisticated variations can be developed.

Furthermore, using loss functions such as NLL which enables a probabilistic prediction is a promising concept to improve the practical applicability of the model. However, further investigation is needed to enhance prediction accuracy. Combining the concepts of WMSE and NLL to create a model that provides probabilistic outputs focused on peak predictions could be an interesting area for future research.

Investigating other types of algorithms expanding on the LSTM algorithm is also a possibility for future studies. For instance, models that fuse CNN with LSTM have been reported in the literature to produce high-performance results. Additionally, the use of LSTM models with multi-channel architectures to better handle exogenous variables is seen as a promising concept.

Besides LSTM algorithms, the literature also reports high-performance results from other machine learning methods, such as Gated Recurrent Units (GRUs) and standard feedforward neural networks. Investigating these alternative techniques is a possible direction for future work.

Appendix A

Attachments for Chapter 3

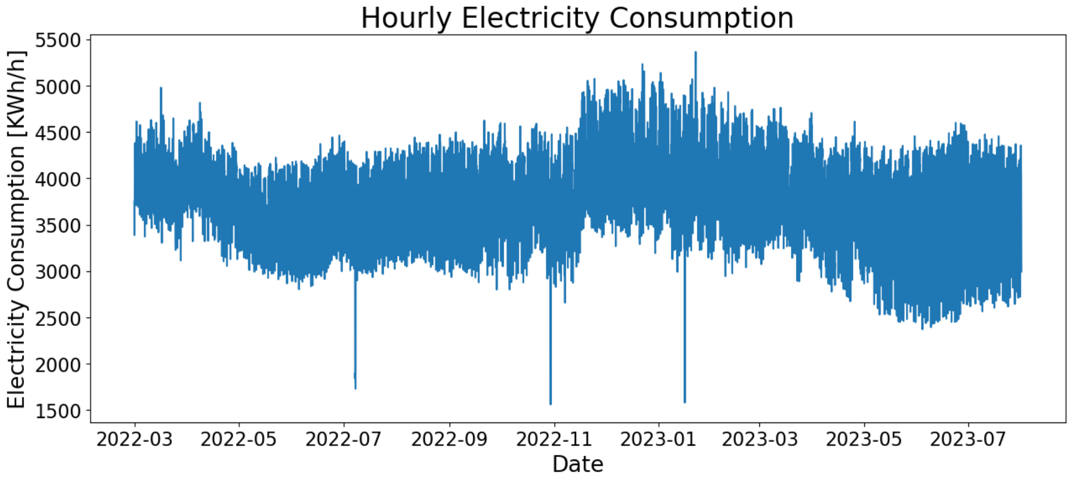

FIGURE A.1: Hourly electricity consumption measurements for the entire time series.

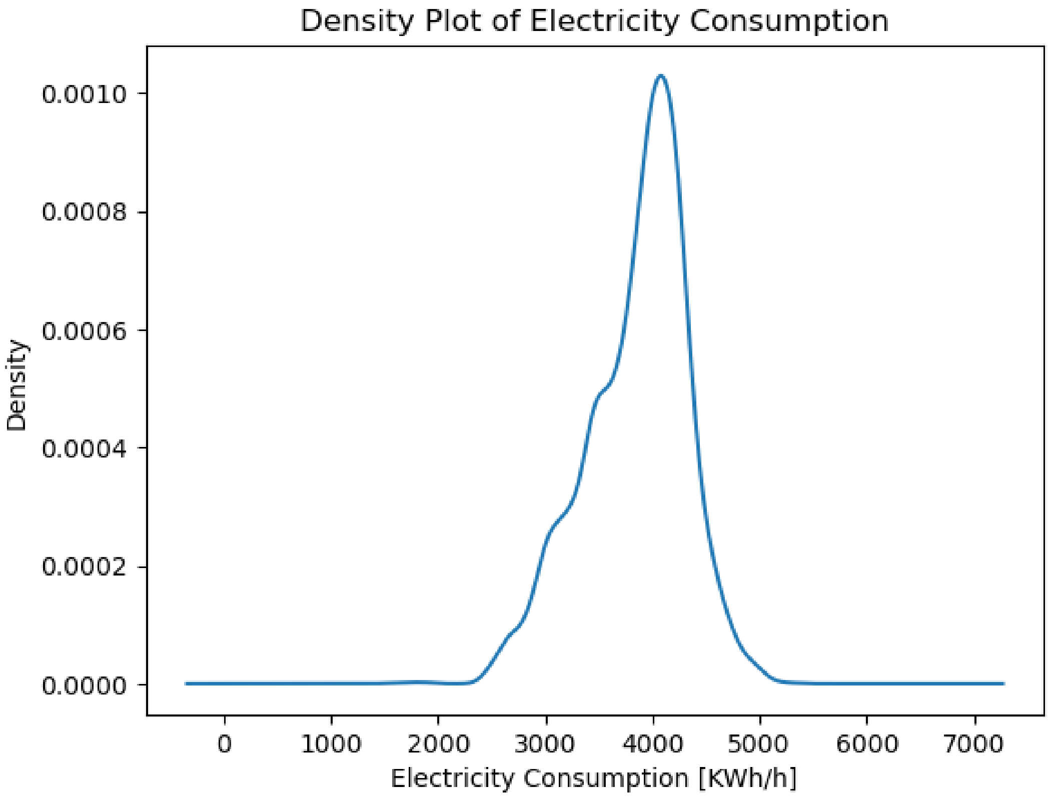

FIGURE A.2: Density plot of electricity consumption data, showing the distribution of values across the entire time series.

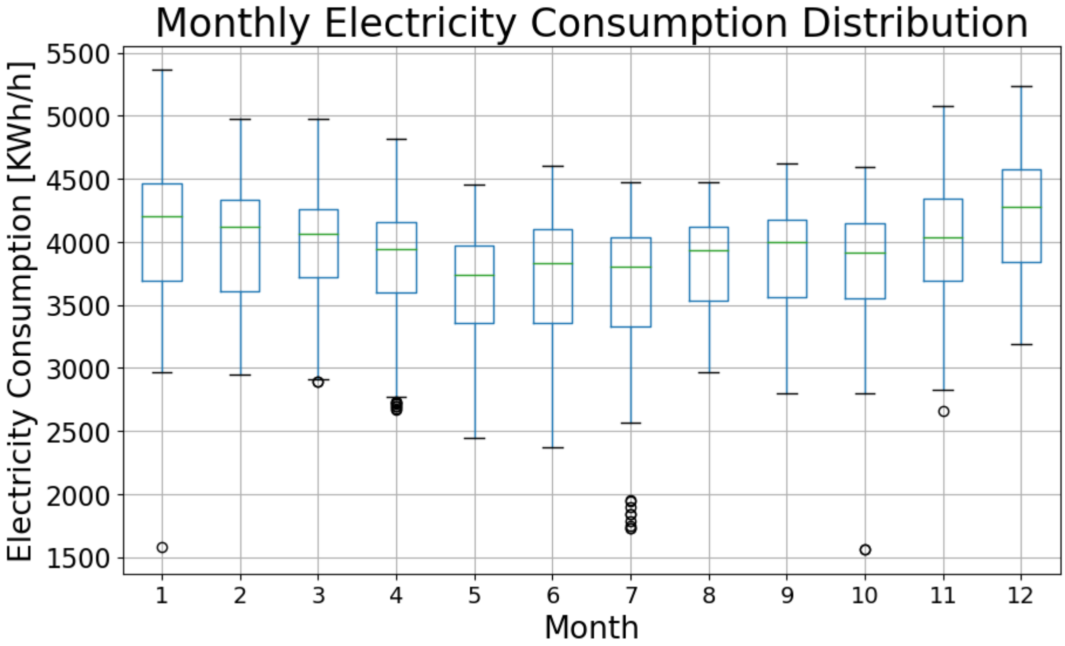

FIGURE A.3: Box plot for each month, showing the monthly value distribution of the electricity consumption.

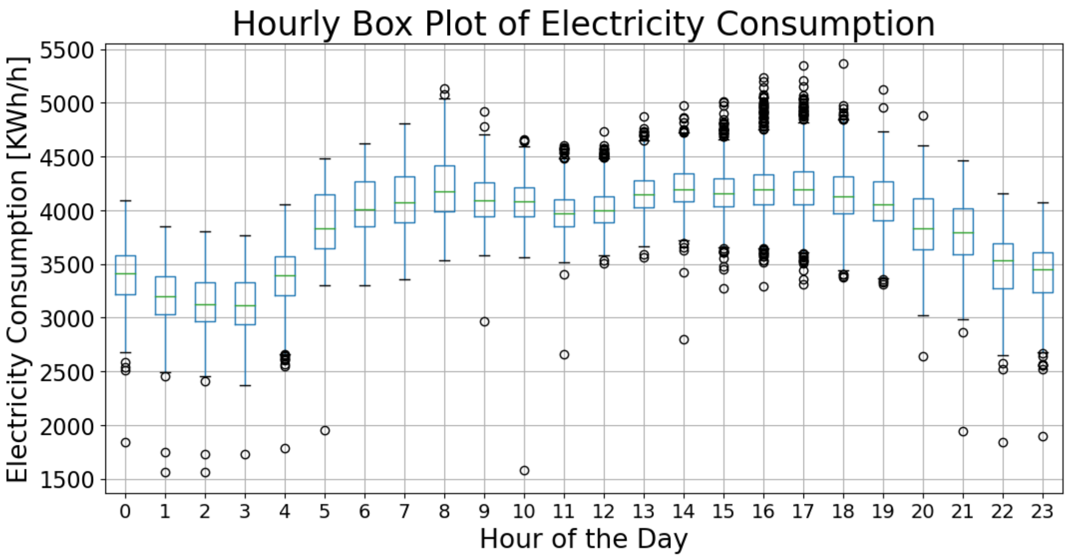

FIGURE A.4: Box plot for every hour, showing the hourly value distribution of the electricity consumption.

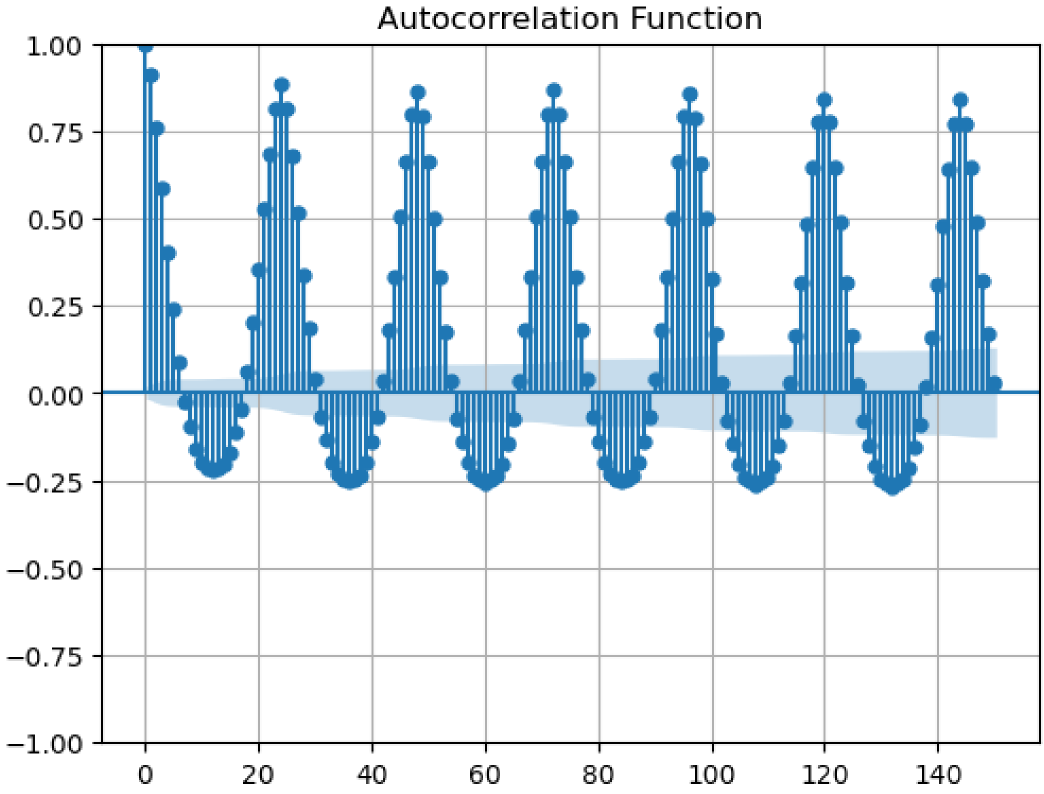

FIGURE A.5: Auto-correlation function applied to electricity consumtion data, showing how the time series correlates with itself over time

Appendix B

Attachments for Chapter 5

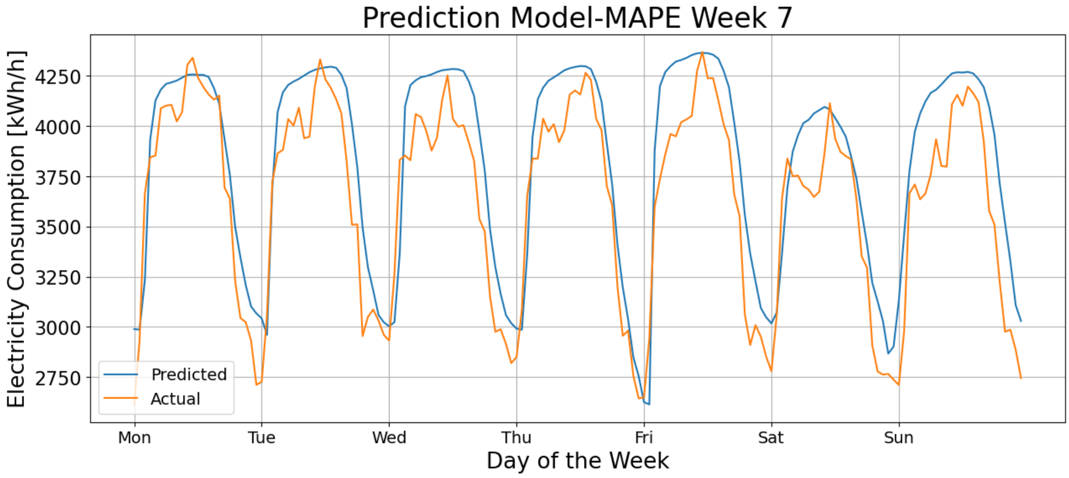

FIGURE B.1: Predictions for week 7 in the test set using Model-MAPE.

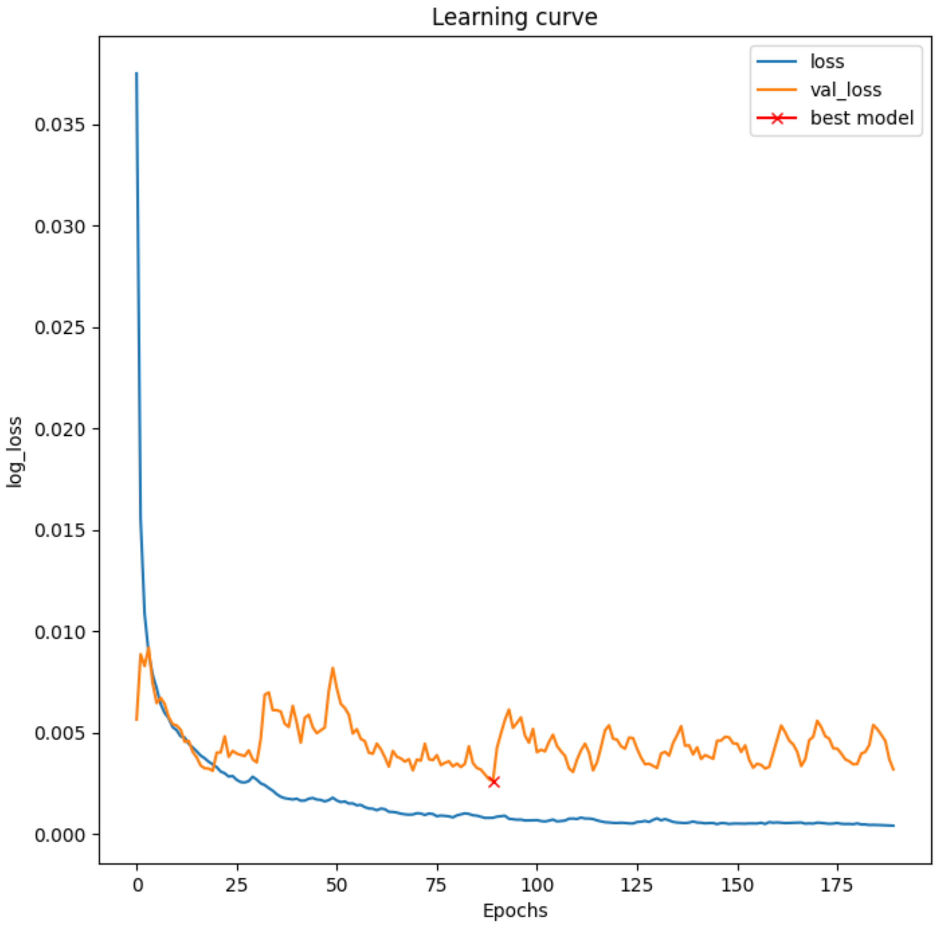

FIGURE B.2: Training and validation loss curves for Model-MSE.

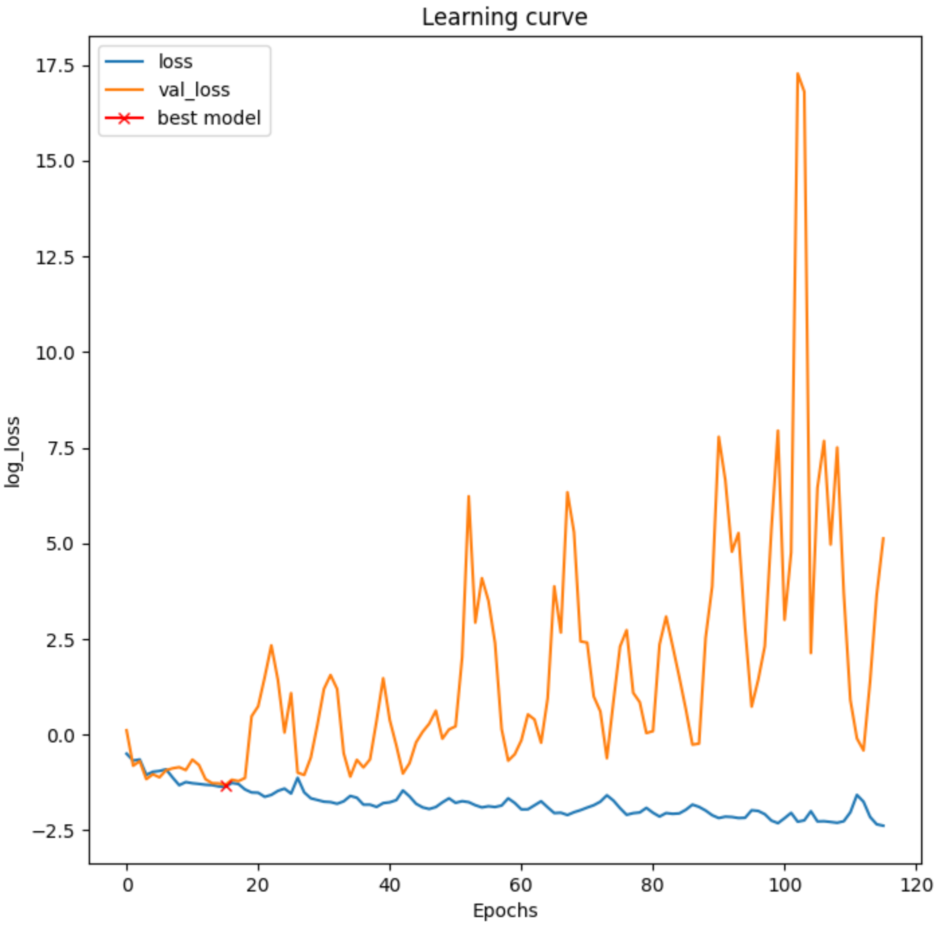

FIGURE B.3: Training and validation loss curves for Model-NLL.

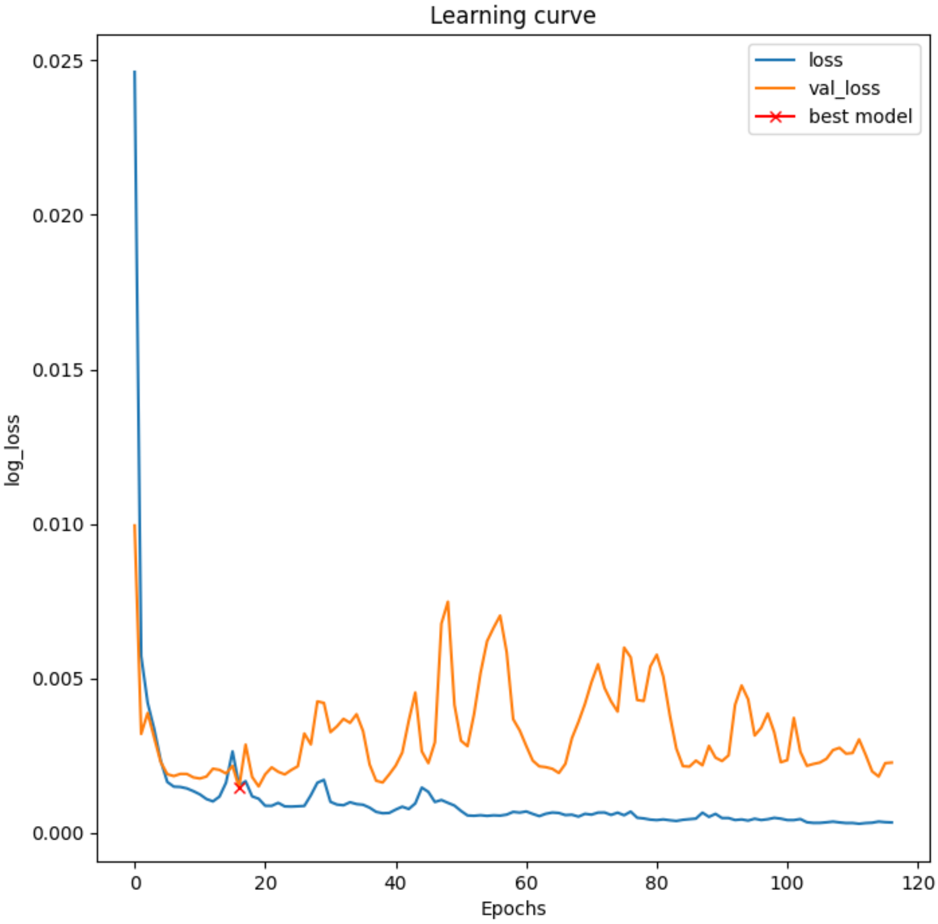

FIGURE B.4: Training and validation loss curves for Model-WMSE.

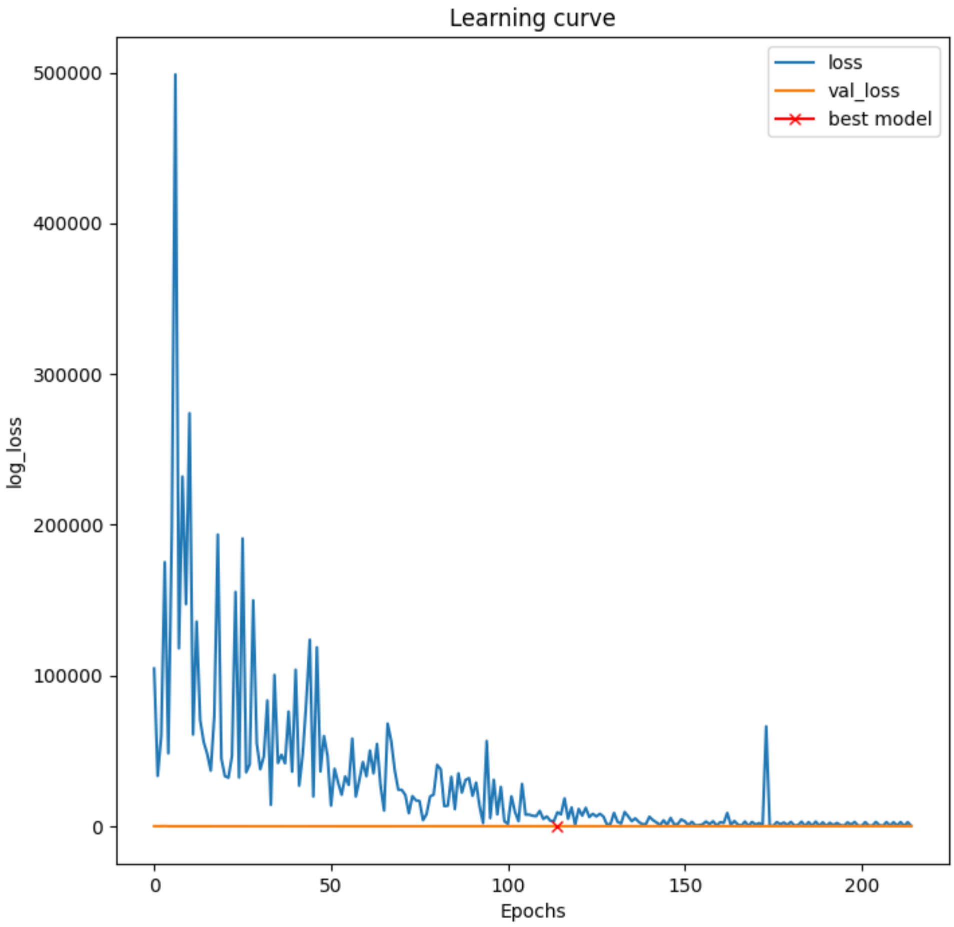

FIGURE B.5: Training and validation loss curves for Model-MAPE. Note that the training and validation losses are on different scales. The author was unfortunately unable to retrieve a plot with separate axes for the two losses.

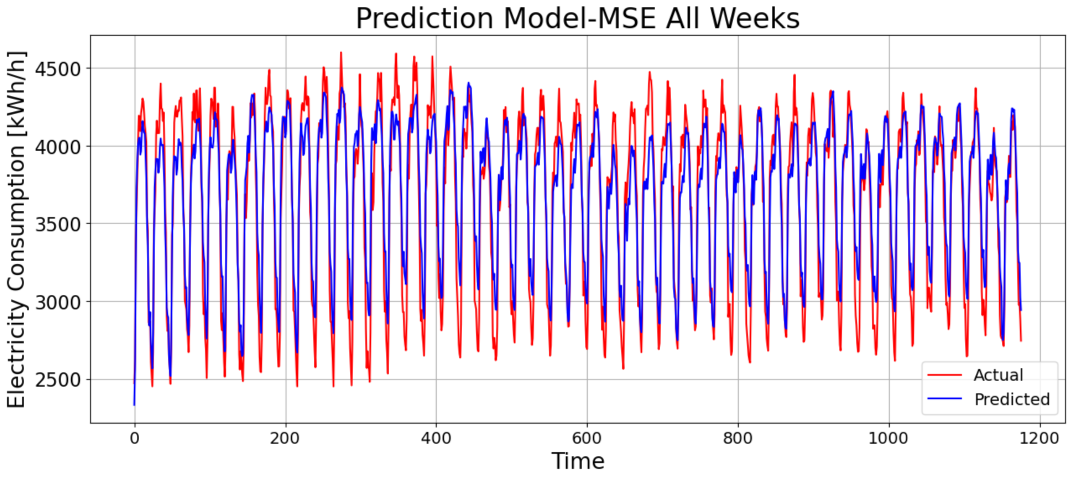

FIGURE B.6: Predictions for the entire test set using Model-MSE.

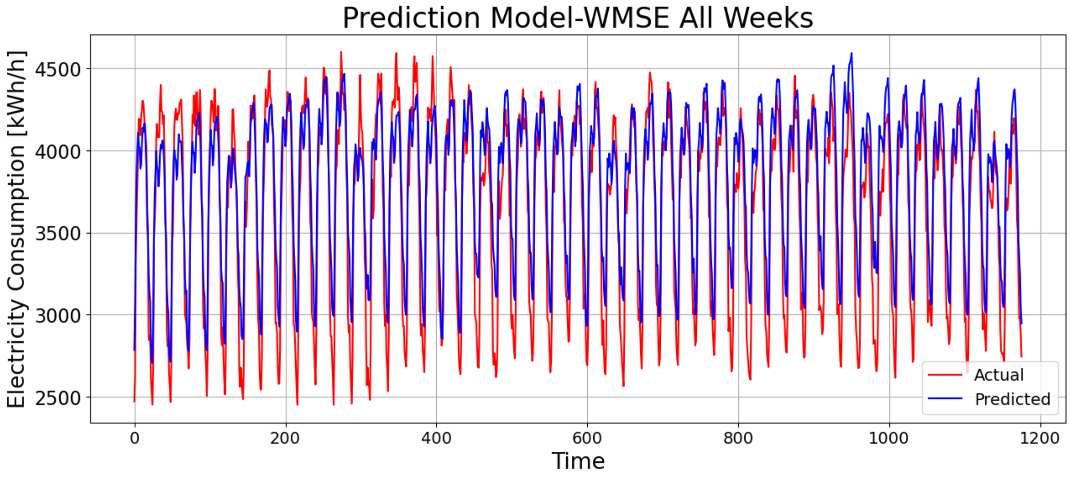

FIGURE B.7: Predictions for the entire test set using Model-WMSE.

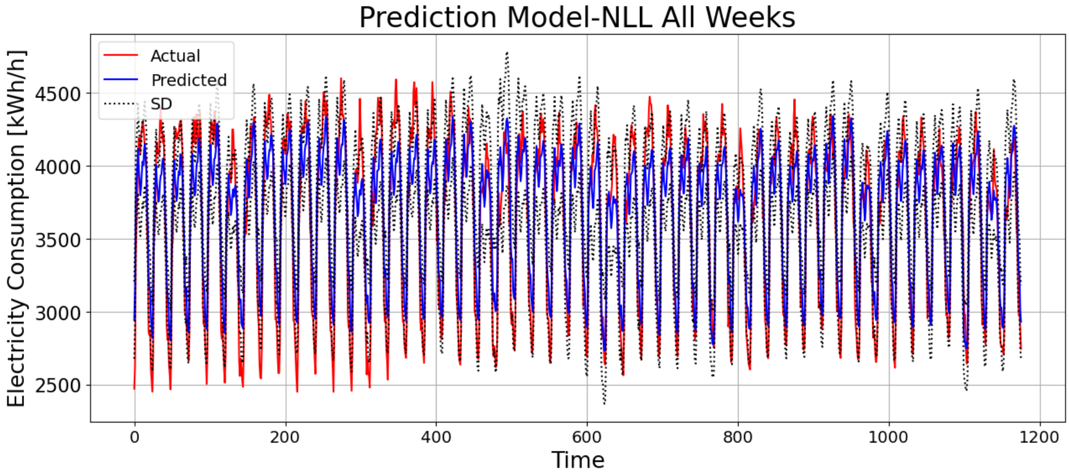

FIGURE B.8: Predictions for the entire test set using Model-NLL.

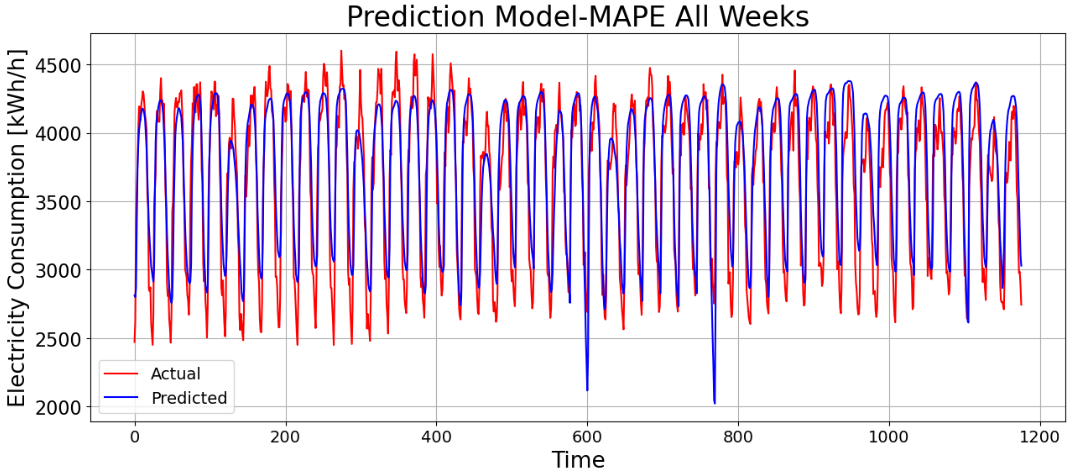

FIGURE B.9: Predictions for the entire test set using Model-MAPE.

Bibliography

[1] H. Birkelund et al. *Langsiktig Kraftmarkedsanalyse 2021 – 2040*. NVE. 2021. URL: https://publikasjoner.nve.no/rapport/2021/rapport2021_29.pdf (visited on 12/28/2023).

[2] J. G. Kirkerud et al. "Langsiktig kraftmarkedsanalyse 2023: energiomstillingen – en balansegang". In: (2023). NVE.

[3] M. Buvik et al. *Norsk og nordisk effektbalanse fram mot 2030*. NVE. 2022. URL: https://publikasjoner.nve.no/rapport/2022/rapport2022_20.pdf (visited on 12/28/2023).

[4] E. Sarker et al. "Progress on the demand side management in smart grid and optimization approaches". en. In: *International Journal of Energy Research* 45.1 (Jan. 2021), pp. 36–64. DOI: 10.1002/er.5631.

[5] A. R. Jordehi. "Optimisation of demand response in electric power systems, a review". In: *Renewable and Sustainable Energy Reviews* 103 (Apr. 2019), pp. 308–319. DOI: 10.1016/j.rser.2018.12.054.

[6] Y. Sun, F. Haghighat, and B. C. M. Fung. "A review of the-state-of-the-art in data-driven approaches for building energy prediction". In: *Energy and Buildings* 221 (Aug. 2020), p. 110022. DOI: 10.1016/j.enbuild.2020.110022.

[7] M. Chang et al. "Aggregated Electric Vehicle Fast-Charging Power Demand Analysis and Forecast Based on LSTM Neural Network". en. In: *Sustainability* 13.24 (Dec. 2021), p. 13783. DOI: 10.3390/su132413783. (Visited on 01/11/2024).

[8] *NeX2G | NMBU*. nb. URL: https://www.nmbu.no/forskning/prosjekter/nex2g (visited on 01/14/2024).

[9] K. R. Kvisberg. *Prediksjon av elektrisitetsbruk i næringsbygg: Casestudie av Oslo Lufthavn Gardermoen*. Master thesis, NMBU. Dec. 2022.

[10] *The electricity grid*. URL: https://energifaktanorge.no/en/norsk-energiforsyning/kraftnett/ (visited on 12/28/2023).

[11] *Security of electricity supply*. URL: https://energifaktanorge.no/en/norsk-energiforsyning/forsyningssikkerhet/ (visited on 12/28/2023).

[12] N. K. Nakstad et al. *Nett i tide – om utvikling av strømnettet*. DDS. June 2022. URL: https://www.regjeringen.no/no/dokumenter/nou-2023-3/id2961311/ (visited on 12/28/2023).

[13] *Energieffektivisering - NVE*. URL: https://www.nve.no/energi/energisystem/energibruk/energieffektivisering/ (visited on 01/10/2024).

[14] L. Sørgard et al. *Mer av alt – raskere*. Jan. 2023. URL: https://www.regjeringen.no/contentassets/5f15fcecae3143d1bf9cade7da6afe6e/no/pdfs/nou202320230003000dddpdfs.pdf (visited on 12/28/2023).

[15] S. Ortega Alba and M. Manana. "Energy Research in Airports: A Review". In: *Energies* 9.5 (May 2016), p. 349. DOI: 10.3390/en9050349.

[16] O. Yildiz, M. Yilmaz, and A. Celik. "Reduction of energy consumption and CO2 emissions of HVAC system in airport terminal buildings". en. In: *Building and Environment* 208 (Jan. 2022), p. 108632. DOI: 10.1016/j.buildenv.2021.108632. URL: https://linkinghub.elsevier.com/retrieve/pii/S0360132321010234 (visited on 01/10/2024).

[17] H. Vefsnmo et al. *Scenarier for fremtidens elektriske distribusjonsnett anno 2030-2040*. Sept. 2020. URL: https://sintef.brage.unit.no/sintef-xmlui/bitstream/handle/11250/2681944/01-2020%2b-%2bCINELDI-rapport.pdf?sequence=2&isAllowed=y (visited on 01/04/2024).

[18] K. W. Høiem et al. *Mulighetsstudie om bruk av fleksibilitetsressurser hos nettselskap*. Energi Norge AS. 2021. URL: https://www.fornybarnorge.no/contentassets/72d407b08a0045b59de36c5545a58069/bruk-av-fleksibilitet-i-nettselskap-2021.pdf (visited on 12/28/2023).

[19] K. Dalen et al. *Fleksibilitet som kilde til verdiskapning og forretningsutvikling*. Statnett. June 2023.

[20] I. I. Lampropoulos. "Energy management of distributed resources in power systems operations". In: (2014). Phd Thesis, Technische Universiteit Eindhoven. DOI: 10.6100/IR771935.

[21] B. K. Sovacool et al. "Actors, business models, and innovation activity systems for vehicle-to-grid (V2G) technology: A comprehensive review". In: *Renewable and Sustainable Energy Reviews* 131 (Oct. 2020), p. 109963. DOI: 10.1016/j.rser.2020.109963.

[22] B. Bibak and H. Tekiner-Moğulkoç. "A comprehensive analysis of Vehicle to Grid (V2G) systems and scholarly literature on the application of such systems". In: *Renewable Energy Focus* 36 (Mar. 2021), pp. 1–20. DOI: 10.1016/j.ref.2020.10.001.

[23] *Nettleie - NVE*. URL: https://www.nve.no/reguleringsmyndigheten/kunde/nett/nettleie/ (visited on 01/10/2024).

[24] M. Slowik and W. Urban. "Machine Learning Short-Term Energy Consumption Forecasting for Microgrids in a Manufacturing Plant". en. In: *Energies* 15.9 (May 2022), p. 3382. DOI: 10.3390/en15093382. URL: https://www.mdpi.com/1996-1073/15/9/3382 (visited on 01/06/2024).

[25] J. Trebbien. *Explainable Artificial Intelligence and Deep Learning for Analysis and Forecasting of Complex Time Series: Applications to Electricity Prices*. Master thesis, University of Cologne. Mar. 2023.

[26] S. Raschka and V. Mirjalili. *Python Machine Learning*. Third Edition. Packt Publishing, 2019.

[27] B. Widrow et al. “An Adaptive "Adaline" Neuron Using Chemical "Memistors"”. In: *Stanford Electron Labs, Stanford University* 1553-2 (1960). Technical Report.

[28] S. Hochreiter and J. Schmidhuber. “Long short-term memory”. In: *MIT Press* 9.8 (1997), pp. 1735–1780. DOI: 10.1162/neco.1997.9.8.1735.

[29] I. Goodfellow et al. *Deep Learning*. MIT Press, 2016. URL: https://www.deeplearningbook.org/ (visited on 12/28/2023).

[30] R. Kohavi. “A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection”. In: vol. 12. Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI). 1995, pp. 1137–1143.

[31] S. H. Rafi et al. “A Short-Term Load Forecasting Method Using Integrated CNN and LSTM Network”. In: *IEEE Access* 9 (2021), pp. 32436–32448. DOI: 10.1109/ACCESS.2021.3060654.

[32] J. F. Torres, F. Martínez-Álvarez, and A. Troncoso. “A deep LSTM network for the Spanish electricity consumption forecasting”. In: *Neural Computing and Applications* 34.13 (July 2022), pp. 10533–10545. DOI: 10.1007/s00521-021-06773-2. URL: https://doi.org/10.1007/s00521-021-06773-2 (visited on 12/28/2023).

[33] X. Shao and C. S. Kim. “Multi-Step Short-Term Power Consumption Forecasting Using Multi-Channel LSTM With Time Location Considering Customer Behavior”. In: *IEEE Access* 8 (2020), pp. 125263–125273. DOI: 10.1109/ACCESS.2020.3007163.

[34] J. Hwang, D. Suh, and M.-O. Otto. “Forecasting Electricity Consumption in Commercial Buildings Using a Machine Learning Approach”. In: *Energies* 13.22 (Nov. 2020), p. 5885. DOI: 10.3390/en13225885.

[35] L. Engerengen, E. Tandberg, and I. S. Kristiansen. *Oslo lufthavn Gardermoen*. Aug. 2023. URL: https://snl.no/Oslo_lufthavn_Gardermoen (visited on 12/28/2023).

[36] R. G. Tveitane. *Fleksibilitet i parkerte elbiler ved næringsbygg : en casestudie av Oslo lufthavn Gardemoen*. Master thesis, NMBU. 2021.

[37] *Meteorologisk Institutt. Norsk Klimaservicesenter*. Aug. 2023. URL: https://seklima.met.no/ (visited on 08/11/2023).

[38] M. Abadi et al. *TensorFlow: Large-scale machine learning on heterogeneous systems*. Software available from tensorflow.org. 2015. URL: https://www.tensorflow.org/.

[39] F. Chollet et al. *Keras*. 2015. URL: https://keras.io (visited on 01/07/2024).

[40] S. Grøtan. *LSTM models*. https://github.com/Siggmeister/LSTM-models/tree/main.

[41] *sklearn.preprocessing.MinMaxScaler*. URL: https://scikit-learn/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html (visited on 01/07/2024).

[42] D. P. Kingma and J. Ba. "Adam: A Method for Stochastic Optimization". In: (2014). DOI: 10.48550/ARXIV.1412.6980.

[43] G. G. Løvås. *Statistikk for universiteter og høgskoler*. 4th edition. Universitetsforlaget, 2018.

[44] Y. Wang et al. "Probabilistic individual load forecasting using pinball loss guided LSTM". en. In: *Applied Energy* 235 (Feb. 2019), pp. 10–20. DOI: 10.1016/j.apenergy.2018.10.078. URL: https://linkinghub.elsevier.com/retrieve/pii/S0306261918316465 (visited on 01/11/2024).

[45] Y. Chen, H. Tan, and U. Berardi. "Day-ahead prediction of hourly electric demand in non-stationary operated commercial buildings: A clustering-based hybrid approach". In: *Energy and Buildings* 148 (Aug. 2017), pp. 228–237. DOI: 10.1016/j.enbuild.2017.05.003.

[46] D. Chi. "Research on electricity consumption forecasting model based on wavelet transform and multi-layer LSTM model". In: *Energy Reports*. 2021 International Conference on New Energy and Power Engineering 8 (July 2022), pp. 220–228. DOI: 10.1016/j.egyr.2022.01.169.

[47] G. Memarzadeh and F. Keynia. "Short-term electricity load and price forecasting by a new optimal LSTM-NN based prediction algorithm". In: *Electric Power Systems Research* 192 (Mar. 2021), p. 106995. DOI: 10.1016/j.epsr.2020.106995.

Norges miljø- og biovitenskapelige universitet
Noregs miljø- og biovitskapelege universitet
Norwegian University of Life Sciences

Postboks 5003
NO-1432 Ås
Norway