Predicting next day's production in run-of-river hydropower plants using machine learning

Kamilla K. Wergeland

A Master's Thesis in Energy

Geophysical Institute

University of Bergen

Supervisor: Asgeir Sorteberg

June 1, 2023

Abstract

This master thesis focuses on time series forecasting using machine learning to predict hourly production in run-of-river hydropower plants. As electrical energy cannot be easily stored, it must be produced and consumed at the same time. To balance production and consumption, power suppliers in Norway must report to Statnett their expected energy production for the following day. Run-of-river hydropower plants rely on the natural water flow in rivers and do not involve any storage. Therefore, the production must be estimated based on the prevailing weather and hydrological conditions. Accurate predictions are crucial to minimize imbalance fees and ensure grid stability. The main objective of this study is to enhance the accuracy of predictions compared to Småkraft AS's current method. The study aims to answer three research questions related to the most significant time intervals for weather variables that affect production, the most important features for making predictions, and the model's performance in various weather situations. Two of Småkraft's power plants, Bjørgum power plant and Furegardane power plant, provide the data for testing three machine learning models: a random forest regressor, a multilayer perception neural network, and a long short-term memory neural network. Input data for the models include weather forecast and observational-based variables, along with engineered features such as accumulated rainfall and snowmelt. The study concludes that the long short-term memory neural network is the best model and outperforms the current method used by Småkraft at both locations. The findings suggest that machine learning models can significantly improve the accuracy of hydropower production forecasting, which can benefit hydropower plant operators as well as the stability of the electricity grid. One limitation of the model is that it requires years of historical data and consequently will not be suitable for newly established power plants. Future work could focus on using data from power plants with comparable location, production patterns, and climate to predict the production in the power plant of interest.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Background

Nearly 200 countries have committed to The Glasgow Climate Pact, which seeks to limit global warming to 1.5 degrees Celsius and achieve net-zero greenhouse gas emissions by 2050. The plan involves transitioning to clean energy and increasing investments in renewable energy [*United Nations* (2022)]. As power generation increasingly comes from variable renewable energy sources, the flexibility of power systems will decrease [*Impram et al.* (2020)]. Therefore, accurately predicting production will be crucial to ensure a reliable energy supply and maintain grid stability. Additionally, in order to accelerate the development of new renewable energy sources it is important that the technologies are profitable and have minimal negative impact on the local environment. Småkraft AS is a company that have these aspects in the core of their business model. Småkraft supports landowners in Norway to realise small-scale hydropower projects by contributing with capital and competence. At the time of writing, Småkraft owns and operates 176 small-scale hydropower plants in Norway and 45 small-scale hydropower plants in Sweden, generating over 2TWh renewable energy in a normal year [*Småkraft* (2023)].

1.1.1 A Brief History of Hydropower

For thousands of years, people have harnessed the power of water to produce work, from grinding flour to generating electricity. Evidence of hydropower dates back to the first millennium BC, and the coupling of water wheels and generators for electric power began in the late 19th century. The first mills and water wheels were simple, but the introduction of iron during the industrial revolution led to the development of more advanced turbines. James Francis developed the first modern water turbine in 1849, which is still widely used today [*Breeze* (2018)].

The Francis turbine, also known as a mixed flow turbine, is designed to extract the maximum energy from water by carefully shaping its blades based on the water head and flow volume. It is optimally used for medium water heads and large flow volumes and can capture up to 95% of the energy in the water [*Breeze* (2019)]. In the 1870s, American inventor Lester Allen Pelton developed the Pelton wheel, which is an impulse turbine used for medium to high water heads and lower flow rates [*Breeze* (2019)]. Another important development is the Kaplan turbine which was developed by Viktor Kaplan at the beginning of the 20th century with the aim to improve the efficiency of the Francis turbine at low heads and variable flow rates. It is still a reaction turbine driven by the pressure and velocity of the water, but its impeller shape and blade number differ from the Francis turbine [*Polák* (2021)].

Hydroelectric power plants have been in operation since the late 19th century, with rapid innovation and optimisation of efficiency occurring throughout the 20th century. Strong post-war economic and population growth led to significant hydropower development in Western Europe, the Soviet Union, North America, and Japan [*iha* (2022)]. Today, hydropower remains the largest renewable source of electricity, generating more than any other renewable technology. To reach the goal of net zero emissions by 2050, a 3% average hydropower capacity growth rate is required, which has not been achieved in the last five years [*Bojek* (2022)]. Large-scale hydropower involve damming of rivers and creating large reservoirs, which can have negative environmental impacts. These impacts include the displacement of communities, loss of natural habitats, disruption of aquatic ecosystems, alteration of river flow, and greenhouse gas emissions from decaying vegetation in flooded areas. As a result, attention has shifted to smaller-scaled, run-of-river hydropower schemes, which are considered less harmful to the environment [*Anderson et al.* (2015)].

1.1.2 Run-of-River Hydropower Plants

Run-of-river hydropower plants utilise the natural water flow in rivers to produce energy. Small scale hydropower is defined as hydropower plants with capacities up to 10MW [*NVE* (2015)]. The key concepts of a hydropower plant will be explained in Chapter 2: Theory.

The Norwegian Water Resources and Energy Directorate (NVE) has been responsible for managing Norway's water and energy resources since 1921. NVE's mandate is to ensure that the development of Norwegian hydropower is environmentally friendly and beneficial to society [*NVE* (2016)]. All small-scale hydropower plants with an installed capacity from 1 MW to 10 MW, has to apply to NVE for a licence (concession). When granted a licence to build and operate a small-scale hydropower plant, NVE sets requirements for, amongst other things, the amount of water the power plant is allowed to use through maximum and minimum intake capacity, and requirements for environmental water flow.

In this thesis data from two of Småkraft's power plants, Bjørgum power plant and Furegardane power plant, will be used. Bjørgum power plant is located in Agder county, while Furegardane power plant is located in Vestland county, as illustrates in Figure 1.1. A description of the power plant's catchment areas and the climate at their locations is presented in Chapter 4: Methods. Bjørgum power plant has been in operation since 2004, and has an average yearly production of 17GWh. The power plant has a permit of using a maximum of $2.3 m^3/s$, resulting in a capacity of 5.5MW. Furegardane has been in operation since 2014, and has had an average yearly production of 11GWh. Furegardane has a permit to use $2.4 m^3/s$, resulting in a capacity of 5.35MW. Both power plants use one pelton turbine.

Figure 1.1: Map displaying the location for Bjørgum power plant and Furegardane power plant.

1.1.3 The Norwegian power market

Norway is part of a joint Nordic power market with Sweden, Denmark and Finland, and this is in turn integrated into the wider European power market. The power market is composed of multiple markets where bids are presented and prices are decided, which include the day-ahead market, continuous intraday market, and balancing market. Trading for day-ahead and intraday occurs on the Nord Pool exchange, while the balancing market is operated by Statnett. The day-ahead market is the main power trading market in the Nordic region where participants bid and offer contracts for physical power delivery hour-by-hour the next day. Trading takes place on the Nord Pool exchange between 08:00 and 12:00 each day, and prices are determined based on all purchase and sell orders received, as well as transmission capacity available [*Norwegian Ministry of Petroleum and Energy* (2022)]. In this thesis, the deadline for submitting bids for the next day is set to 11:30, aligning with the current practice of Småkraft.

Electricity cannot be easily stored, and there must therefore always be an exact balance between generation and consumption. To address this, Statnett procures balancing power from the balancing power market. Imbalances arise from uncertainties in plans and from failures in generation, consumption, and the grid. Thus, an imbalance settlement is essential for a commercial electricity market [*eSett* (2023)]. For each imbalance settlement period, the imbalance volumes are priced based on the regulation power market prices. The price depends on the dominating direction from the regulation power market and can be either up or down regulation price.

To illustrate this, consider two hypothetical scenarios where in both scenarios a

heavy rainfall event is forecasted and a run of river hydropower plant is expected to produce at maximum capacity the next day. However, the rainfall event does not occur as expected, and the power plant is unable to generate as much power as planned. In the first scenario, the consumption in the area is already surpassing the generation and there would be an up regulation price. Since the power plant is contributing to the imbalance, the regulation price for the power plant would be negative. In the second scenario, the generation in the area is surpassing the consumption and there would be a down regulation price. In this case, the power plant would be helping to reduce the imbalance, and the regulation price would be positive.

To summarise, the regulation price vary both in direction and magnitude, making it an unpredictable cost for the power supplier. The only way to reduce the risk of getting high imbalance costs is to reduce the imbalance volume by having accurate estimates of the production plan for the next day. In addition to the regulation price, there is a fixed imbalance fee. At the time of writing, this fee is at 1.15EUR/MWh which equals 11.58NOK at the time of writing [*Norges Bank* (2023)].

1.2 Objectives

The main objective of this thesis is to build a machine learning model that is able to predict next day's production hour by hour in a run of river hydropower plant. Data from two of Småkraft's power plants will be used, and the goal will be to improve the prediction accuracy compared to the solution Småkraft uses today. In relation to this, the following research questions will be answered:

- Which time intervals for weather variables affect the production?
- Which features are most important when making predictions?
- How does the final model perform in various weather situations?

The structure of the thesis is as follows: Chapter 2 covers the theory including key concepts in hydrology, the process of energy production in a hydropower plant, data preparation techniques, and an introduction to machine learning, including model descriptions and evaluation techniques. Chapter 3 outlines the data and methods used. Chapter 4 presents and discusses the modelling results and addresses the research questions. Finally, Chapter 5 provides a summary and conclusion.

Chapter 2

Theory

2.1 Hydrology

The field of hydrology focuses on the movement and distribution of fresh water across the earth's surface, underground, and atmosphere [*Davie* (2019)]. The volume and flow rate of water is what controls the power generation in a run of river hydropower plant. In this section some fundamental hydrological concepts will be introduced and explained.

In hydrology the most common spatial unit is the **catchment** which, by *Davie* (2019), is defined as "the area of land from which water flows towards a river and then in that river to the sea". This means that wherever water falls in the catchment it will end up in the same place, unless it evaporates. Water moving in a channelised form is referred to as **discharge** (m^2/s). The size of a catchment can range from hectares (0.01km^2) to millions of square kilometres, and all catchments can bee divided into a set of sub-catchments. The catchment boundary is determined by the topography through the assumption that all water falling on the surface flows downhill. This assumption may not always be correct as the underlying geology can make it possible for the water to flow as groundwater to another catchment. This means that the surface water catchment and the groundwater catchment are not necessarily the same [*Davie* (2019)].

The **hydrological cycle** explains how water moves between the earth and the atmosphere in different states. The cycle consist of liquid water that evaporates into water vapor and is moved around the atmosphere. At some point the water vapor condenses into liquid again and falls to the surface as precipitation. The deviation between evaporation and precipitation in the terrestrial zone is the **runoff** which completes the hydrological cycle. *Davie* (2019) defines the runoff as "water moving over or under the surface towards the ocean". The concept of the hydrological cycle can be represented in terms of an equation, the **water balance equation**:

$$P \pm E \pm \Delta S \pm Q = 0$$

where P is precipitation, E is evaporation, ΔS is change in storage and Q is runoff. The $\pm$ terminology is used as each term can be either positive (gain) or negative (loss) depending on whether the equation is viewed in terms of the earth or the atmosphere. The water balance equation can be re-arranged to quantify water flow in the river:

$$Q = P - E - \Delta S$$

In this equation the change in storage (ΔS) can either be positive (water released) or negative (water absorbed). The storage term includes soil moisture, deep groundwater, water from lakes, glaciers and seasonal snow cover [*Davie* (2019)].

The process of water moving below the surface includes infiltration into the soil and the movement of water either in the unsaturated zone (throughflow) or the in the saturated zone (groundwater flow). The amount of water that enters the soil over a specific time interval is referred to as **infiltration rate**, and relies on the water content in the soil and the soil's ability to transmit water. Land-cover and antecedent moisture condition are important factors controlling the infiltration rate. The antecedent moisture condition indicates the initial level of soil moisture before the start of a rainfall event. The infiltration rate will be faster in a drier soil than in a wetter soil.

2.2 Hydropower: water to energy

A run of river hydropower plant utilise the kinetic energy of running water in a river to produce electrical energy. The power, often measured in mega watts (MW), is the rate at which energy is generated. The amount of energy generated during one hour is usually quantified in megawatt-hours (MWh). The power output largely depends on the quantity of available water and the head, which refers to the vertical difference between the water intake and the turbine. A higher head leads to a more substantial gravitational force, resulting in a higher pressure, which translates to more power. Furthermore, the efficiency of the power plant determines the amount of energy produced relative to the available energy in the water. This can be summarised in the following equation:

$$P_{prod} = \rho_w \cdot Q_{prod} \cdot g \cdot \eta_{plant} \cdot H_{gross}$$

where P_{prod} is the power output [W], ρ_w is the density of water [1000kg/m^3], Q_{prod} is the water flow to the turbine [m^3/s], g is the gravitational force [m/s^2], η_{plant} is the efficiency of the power plant, and H_{gross} is the vertical difference between the intake and the turbine [m], as illustrated in Figure 2.1.

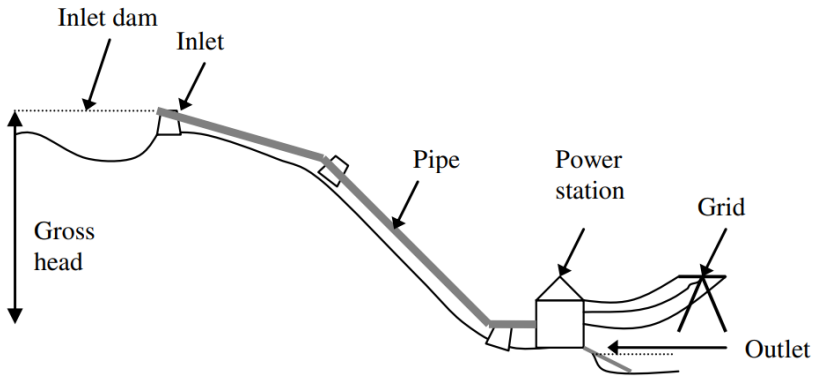


Figure 2.1: Illustration of a small-scale hydropower plant [Bøckman et al. (2008)].

When calculating the water flow to the turbine, Q_{prod}, loss due to environmental flow and over flow needs to be considered. The environmental flow is the amount of

water that should be passing the intake at all times when operating. The overflow is the amount of water lost when the total discharge minus the environmental flow exceeds the maximum intake capacity. The water flow to the turbine can be defined by the following equation:

$$Q_{prod} = Q_{total} - Q_{overflow} - Q_{environmental}$$

The efficiency of the power plant (η_{plant}) can be divided into the efficiencies of the units (η_{unit}), which includes the turbine, generator and transformer, and the head loss ($\eta_{headloss}$) due to friction in the penstock:

$$\eta_{unit} = \eta_{turb} \cdot \eta_{gen} \cdot \eta_{trans}$$

$$\eta_{headloss} = \frac{H_{gross} - H_{loss}}{H_{gross}}$$

Head loss due to friction can be calculated using Darcy-Weisbach's equation for head loss:

$$H_{loss} = f_d \cdot \frac{(\frac{Q}{\pi {r_{pipe}}^2})^2}{2g} \cdot \frac{L_{pipe}}{2r_{pipe}}$$

where Q is the water flow to the turbine $[m^3/s]$, L_{pipe} is the length of the penstock $[m]$, r_{pipe} is the radius of the penstock $[m]$, and f_d is the head loss coefficient of the penstock. Different types of penstocks have their own head loss coefficient, where a higher head loss coefficient leads to a greater loss.

2.3 Data Management

Machine learning algorithms require large amounts of data to train, validate and test the models. Consequently, data management is a large part of the machine learning process.

2.3.1 Exploratory Data Analysis

Exploratory data analysis (EDA) involves summarising the main characteristics of a dataset and presenting them through visualisations. This approach is useful for gaining a better understanding of the data. In this thesis moving average has been used to visualise the seasonality in the variables. A moving average is a time series constructed by taking the averages of several sequential values of another time series. If the original

time series is represented by $y_1, ..., y_n$, then a two-sided moving average of the times series can be expressed as [*Hyndman* (2011)]:

$$z_t = \frac{1}{2k+1} \sum_{j=-k}^{k} y_{t+j}, \qquad t = k+1, k+2, ..., n-k$$

Furthermore, to explore the linear relationship between variables, correlation has been calculated. The correlation between two variables, x and y, can be expressed as:

$$r_{xy} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

The correlation coefficient, r_{xy}, varies from -1 to 1. A correlation of 1 implies a perfectly positive relationship, meaning that when one variable increases the other increases by the same amount. A correlation of -1 indicates a perfectly negative relationship, meaning that when one variable decreases the other decreases by the same amount. If two variables have no linear relationship, the correlation coefficient between them would be 0 [*James et al.* (2013)].

2.3.2 Prepossessing Techniques

Preprocessing refers to the steps taken to prepare the data before feeding it into a machine learning model. These steps may include creating new features, handling missing values, and scaling of variables. The approach used for handling missing data depends on how the data is missing. Missing data is often categorises as data missing not at random (MNAR), missing at random (MAR), and missing completely as random (MCAR). Data missing not at random means that the mechanism for the missing data is related to the unobserved value itself. On the other hand, data are missing at random if the reason for the omission is not related to the unobserved value. Lastly, data are missing completely at random if the absence is unrelated to the values of both the observed and unobserved data. The MCAR assumption is stronger than the MAR assumption, and most imputation methods require MCAR to be valid. Assuming that the data are MCAR, methods for handling the missing data could be to discard any missing values, rely on the learning algorithm to deal with the missing values in its training phase, or use an impute-technique for filling any missing values. Methods for filling missing values may include using the variable mean, forward-fill, or if the variables have some dependency it is possible to estimate the missing values with a predictive model [*Hastie et al.* (2009)].

Another important part of preprocessing is scaling of data. While tree-based algorithms are insensitive to the scale of the data, other models, such as neural networks, rely heavily on scaling to operate optimally. Scaling ensures that each features have

equal importance. The two most commonly used scaling techniques are max-min normalisation and standardisation. Normalisation scales feature values to a range between 0 and 1 by subtracting the minimum value (x_{min}) from each value and dividing by the range (x_{max}):

$$x_i' = \frac{x_i - x_{min}}{x_{max}}, \qquad i = 1, ..., n$$

When standardising the data, all variables are given a mean of zero and a standard deviation of one:

$$x_i' = \frac{x_i - \bar{x}}{\sigma}, \qquad i = 1, ..., n$$

where $\bar{x}$ is the mean value of the variable and σ is the standard deviation of the variable.

2.4 Machine learning

2.4.1 Introduction to Machine Learning

In this section, a brief introduction to machine learning will be given together with explanations of the concepts used in this thesis. Machine learning is a branch of artificial intelligence, and uses concepts and results from many different fields including statistics, information theory, cognitive science and mathematics [*Mitchell et al.* (2007)]. The main difference between statistics and machine learning is that machine learning algorithms learn and improve from experience without explicit instructions from the user. The machine learning algorithm is given a large dataset and is able to learn patterns and relationships within the observations. The dataset is usually dived into three subset: training data, validation data, and test data. The model is fit using the training data, optimised using the validation data, and finally an unbiased performance measure is calculated using the unseen test data.

Machine learning problems can be divided into two categories: supervised and unsupervised. In a supervised learning problem, each predictor measurement x_i is associated with a response measurement y_i, and the goal is to create a model that predicts future responses accurately. This thesis focuses on such a problem. In contrast, an unsupervised problem occurs when there are no associated responses to the observations. The goal in this situation is to describe the associations and patterns among the input variables [*James et al.* (2013)].

The variables in a machine learning problem can be either quantitative or qualitative. Quantitative variables take on numerical values, while qualitative variables take on values in one of K classes. Machine learning problems where the response variable

is quantitative is referred to as regression problems, while problems involving a qualitative response variable is referred to as classification problems. The predictors in each problem can be either quantitative or qualitative. In this thesis the aim is to predict the power output in a hydropower plant, hence the response variable is quantitative and regression algorithms will be used [*James et al.* (2013)].

2.4.2 Machine Learning Models

In this thesis, three different regression algorithms will be used to predict the response variable. These models include a tree-based algorithm, specifically a Random Forest Regressor, and two neural networks: a Multilayer Perceptron Neural Network and a Long Short-Term Memory Neural Network.

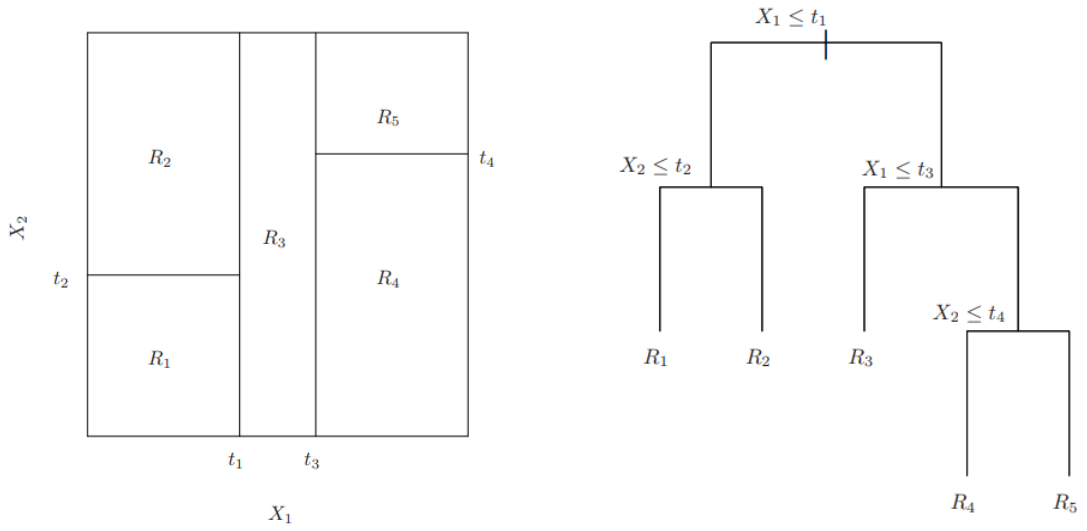
Random Forest Regressor

Tree-based methods for regression and classification involve dividing the predictor space into simpler regions and making predictions based on the mean or mode response of training observations in each region. These approaches are simple and interpretive, but less accurate than the best supervised learning methods. To address this issue, methods have been developed to combine a large number of trees, with random forest being one of them.

In order to understand the random forest algorithm, a simple decision tree needs to be introduced. A decision tree is built by dividing the predictor space into J distinct and non-overlapping regions. This is done using a top-down, greedy approach referred to as recursive binary splitting. The approach begins at the top of the tree with all observations in one region, and then successively splits the predictor space where each split is indicated by two new branches. The approach is referred to as greedy because the best split is considered at the current step without looking ahead. The best split is usually defined as the split leading to the greatest reduction in residual sum of squares (RRS), given by:

$$RSS = \sum_{j=1}^{J} \sum_{i \in R} (y_i - y_{\hat{R}_j})^2$$

where $y_{\hat{R}_j}$ is the mean response for the training observations in the jth region [*James et al.* (2013)]. However, other criteria for splitting can be chosen by the user. This process is continued, but instead of splitting the entire predictor space, the two previously identified regions are considered. The process continues until a stopping criterion is reached, such as a maximum depth of the tree or a maximum number of observations in each region. Figure 2.2 illustrate the output of a recursive binary splitting using two predictors, X_1 and X_2, resulting in five regions, R_1,...,R_5.



(a) Partition *(b) Tree*

Figure 2.2: The output of recursive binary splitting with two predictors illustrated with the partition (a) and the corresponding tree (b) [James et al. (2013)].

The decision tree described suffers from high variance, meaning that if the training data was randomly split in two and a decision tree was fit on each half the resulting trees are likely to be very different. Bagging is a method that addresses this issue by taking B bootstrap samples of size n from the initial training set and creating B different decision trees. This means that for each tree, the n observations are selected from the total training set with replacement such that some observations are likely to be repeated while others a left out. The trees are deep, thus they have high variance and low bias. The resulting predictions from each tree are then averaged, reducing the variance.

Random forest is an improvement over bagged trees by decorrelating the trees. Suppose the dataset consist of a very strong predictor alongside several moderately strong predictors. When using the begging method most or all of the trees will choose this strong predictor as the first split, resulting in similar looking trees. The random forest algorithm avoids this problem of correlated tress by only allowing the trees to consider a random sample of predictors at each split [*James et al.* (2013)].

Multilayer Precepton Neural Network

The other two models used in this thesis are both neural networks. Neural networks are the cornerstone of deep learning, and are based on the structure of the human brain. Feed-forward neural networks (FNN), also called single- and multilayer preceptons (SLP/MLP), are the most basic types of neural networks where the information flows in one direction without any feedback. The MLP takes an input vector of p predictors $\mathrm{X} = (X_1, X_2, ..., X_p)$ and builds a non-linear function $f(X)$ to predict the response Y. The structure of the model is what distinguishes the neural network from other non-linear models such as the tree-based algorithms introduced above. The predictors

$X_1, X_2, ..., X_p$ make up the input layer, and each of the predictors feed into each of the units in the following hidden layer. This is illustrated in Figure 2.3 for a SLP. The MLP has the same structure, but with two or more hidden layers [*James et al.* (2013)].

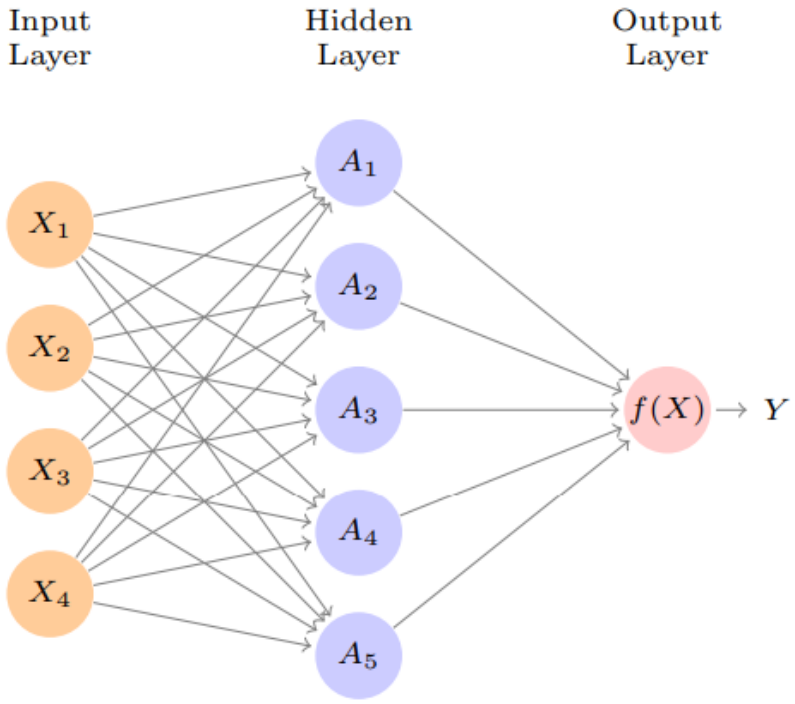


Figure 2.3: Neural network with a single hidden layer [James et al. (2013)].

The neural network model has the form:

$$f(X) = \beta_0 + \sum_{k=1}^{K} \beta_k g(w_{k0} + \sum_{j=1}^{p} w_{jk} X_j) = \beta_0 + \sum_{k=1}^{K} \beta_k A_k$$

where all parameters $\beta_0, ..., \beta_K$ and $w_1, ..., w_K$ need to be estimated from data. The first hidden layer is built up of K activations (also known as neurons) $A_1, ..., A_K$ which are non-linear functions that transform the input $X_1, X_2, ..., X_p$. These activations are then fed into the next hidden layer which treats these as input and computes new activations. The activations from the final hidden layer are fed to the output layer which is a linear model that uses the activations as inputs, resulting in a function $f(X)$ that outputs the predicted value Y. The number of hidden layers and activations, as well as activation functions need to be specified. The most commonly used activation function in modern neural networks is the ReLU (rectified linear unit) activation function, which takes the form:

$$g(z) = (z)_+ = \begin{cases} 0, & \text{if } z < 0 \\ z, & \text{otherwise} \end{cases} \tag{2.1}$$

This means that the ReLU function returns the input value if it is positive or zero, and returns zero if the input is negative. This can be viewed as the units being activated if the input value is above 0, and remaining inactive if the value is below 0 [*James et al.* (2013)].

A widely used procedure for training neural network is back-propagation, first introduces by *Rumelhart et al.* (1986). He defined the procedure as "repeatedly adjusting the weights of the connections in the network so as to minimize a measure of the difference between the actual output vector of the net and the desired output vector". This is done by first forward pass the input through the network until an output is produced, and the error between the prediction and actual value can be calculated. The error is then propagated from the output layer to the input layer, allowing the weights of the network to be adjusted in a way that minimizes the error. This is achieved through the use of the chain rule in calculus, which allows the gradient of the error with respect to the weights to be calculated. The updated weights are then used in the next iteration of forward propagation, and the process is repeated until the error is minimized [*Rumelhart et al.* (1986)].

Long Short Term Memory Neural Network

The long short term memory neural network (LSTM) is a type of recurrent neural network (RNN). RNNs are specifically designed for sequential data, making it ideal for tasks such as forecasting. Unlike MLP, which processes input data in a feedforward manner, RNN is designed to accommodate and take advantage of the sequential nature of the input and processes the data by maintaining a memory of previous inputs. Figure 2.4 displays the structure of a simple recurrent neural network. On the left side of the equal sign there is a compact depiction of the network which is unrolled into a more detailed version on the right side.

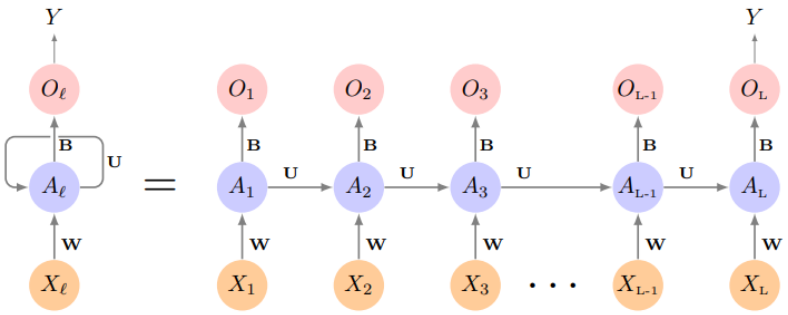


Figure 2.4: Schematic of a simple recurrent neural network [James et al. (2013)].

The input to the neural network is a sequence of vectors $X = \{X_1, X_2, ..., X_L\}$, and the network is built up of a hidden-layer sequence $\{A_l\}_L^1 = \{A_1, A_2, ..., A_L\}$. As the sequence is processed, each X_l feeds into the hidden layer which also takes the previous activation vector A_{l-1} as input, and the current activation vector A_l is produced. Each

A_l feeds into the output layer which produces a prediction O_l, but typically only the last of these, O_L, is of relevance. The same collection of weights **W**, **U** and **B** are used at each step, and needs to be learned by the model.

The main limitation of a simple RNN is that it struggles to learn long-term dependencies, which is often referred to as the vanishing gradient problem. This issue can be addressed using more advanced neural networks such as the LSTM. The LSTM was first indtroduced by *Hochreiter and Schmidhuber* (1997), and has since had a significant impact on the development of recurrent neural network architectures. The LSTM has two tracks of hidden layers, one containing the activations seen in the simple RNN storing the short-term information and an additional layer called the cell state storing the long-term information. This way, when the activation A_l is computed it gets input from hidden units both further back in time, and closer in time [*James et al.* (2013)]. Each cell has three gates: an input gate, an output gate, and a forget gate. The input gate controls the flow of information from the current input and the previous hidden state into the memory cell, and decides which information is important and should be stored and which should be ignored. The output gate controls the flow of information from the memory cell to the output at the current time step, and decides which information should be output and which should be suppressed. The forget gate controls the flow of information that is forgotten from the memory cell over time, and decides which information is no longer relevant and should be removed from the cell [*Gers et al.* (2000)].

Hyperparameter Tuning

As seen for the models above, there are some parameters that are determined based on the data during the training process, while other parameters must be specified by the user. These parameters are called hyperparameters and are used to control the behavior and learning of the algorithm. For the random forest algorithm hyperparameters that can be optimised include, amongst others, the number of trees used, the maximum depth of the tress, and the criterion used to measure the quality of the split. For the MLP neural network the number of hidden layers, the number of neurons, and the activation function are some of the hyperparameters that can be tuned. These hyperparamters are also relevant for the LSTM. In this thesis, the number of epochs and the batch size have been specified for the LSTM algorithm. The number of epochs controls how many complete iterations of the dataset that should be run and needs to be high enough to allow the model to converge to a stable solution, but not so high that it overfits the training data. The batch size defines the number of input sequences processed together in a single forward/backward pass during training, which impacts the convergence rate and ultimately affects the training time and generalization [*Keras*]. Typically, the models have default values for hyperparameters that come into effect when not specified by the user.

In order to find the best hyperparameters, various methods can be employed. In this thesis, the hyperparameters have been tuned using an iterative process called "grid search". The method involves selecting the hyperparameters to be tuned and specifying

a range of values for each one. The algorithm then runs through all possible combinations of hyperparameters to identify the optimal one. For each combination, the machine learning model is trained, and an evaluation metric is computed on the validation data. The combination of hyperparameters that results in the best value for the evaluation metric is chosen.

2.4.3 Evaluation Techniques

The aim of a machine learning model is to be able to accurately predict the response value of unseen data. The model should be able to generalise, meaning that it does not overfit or undefit the training data. When the complexity of a model increase, it tends to adapt itself more closely to the training data and will not generalise well. However, if the model is not complex enough it tends to underfit and fail to identify patterns in the data, resulting in poor generalisation. This balance between over- and underfitting is known as the bias-variance tradeoff. The bias represents the models ability to accurately capture the relationship between the input variables and the response variable. The variance referrers to the models ability to generalise, meaning how much the model will change if trained on various datasets. A complex model has low training bias and high variance, while a simple model has high bias and low variance. To achieve low prediction error on unseen data, the model must strike a balance between bias and variance, as shown in Figure 2.5 [*Hastie et al.* (2009)].

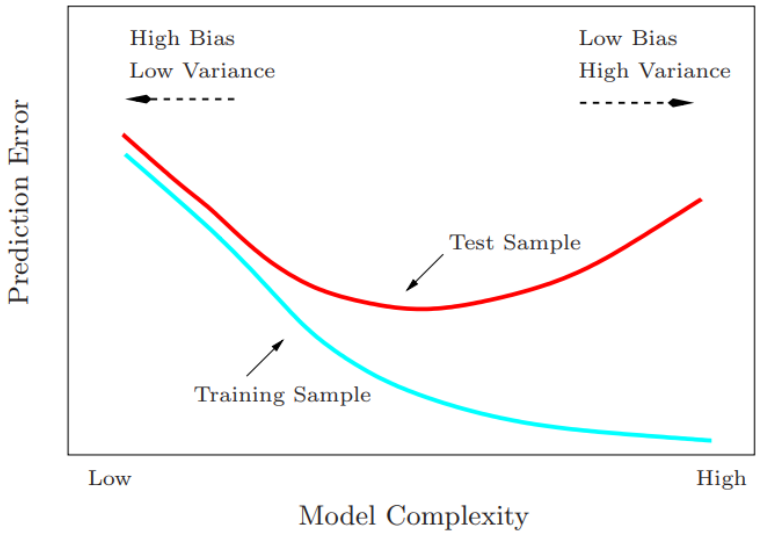


Figure 2.5: Test and training error as a function of model complexity [Hastie et al. (2009)].

To get an unbiased estimate of the prediction error, the unseen test dataset is used. The prediction error can be calculated using various metrics depending on the nature of the problem. In this thesis mean squared error (MSE), mean absolute error (MAE), accuracy-score and f-score have been used to evaluate the prediction error. The MSE

and the MAE are measures used in regression problems, while accuracy-score and f1-score are metrics used for classification problems. The accuracy- and f1-score have been used to get an understanding of the model's performance on different intervals of power output. In this case each interval is considered a class.

The MSE measures the average squared error between the predicted and actual value, and is given by:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{f}(x_i))^2$$

where $\hat{f}(x_i)$ is predicted response value for the ith class and y_i is the actual value. By squaring the error, the MSE is sensitive to outliers.

The MAE measures the average absolute error between the predicted and actual value. The formula for MAE is given by:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{f}(x_i)|^2$$

where $\hat{f}(x_i)$ is predicted response value for the ith class and y_i is the actual value. The MAE is indicate the average magnitude of the error, and are less sensitive to outliers compared to the MSE. Calculating both the MSE and MAE can give an indication on whether there are many outliers in the data.

The accuracy-score simply measures the percentage or ratio of correctly classified observations out of the total number of observations in the dataset. This gives a measure of the overall performance of the model, however, this can be misleading in an imbalanced dataset where the model performs well on the over-represented classes, but are not able to correctly classify the under-represented classes. To address this issue, the f1-score can be used which is defined as:

$$f_1 = \frac{2PR}{(P+R)}$$

where P is precision and R is recall. Precision is defined as the proportion of correctly identified positive cases among all predicted positive cases, including true positives (TP) and false positives (FP). This can be formulated as:

$$P = \frac{TP}{TP+FP}$$

Recall is defined as the proportion of correctly identified positive cases among all actual positive cases, i.e. all true postives (TP) and false negatives (FN). This can be formulated as:

$$R = \frac{TP}{TP + FN}$$

When dealing with multiple classes, the f1-score for each class can be calculated and then their average can be taken. This approach is referred to as macro-averaged f1-score and has been employed in this thesis.

To display information on the actual and predicted classifications, a confusion matrix can be used. The confusion matrix consists of two dimensions, where one is indexed by the actual class of an object, and the other is indexed by the predicted class by the model. Figure 2.6 illustrates the fundamental format of a confusion matrix for a multi-class classification task, where the classes are denoted by $A_1, A_2, ..., A_n$. In the confusion matrix, N_{ij} represnts the number of samples that belong to class A_i but were classified as class A_j [*Deng et al.* (2016)].

		Predicted			
		A_1	... A_j ...		A_n
Actual	A_1	N_{11}	N_{1j}		N_{1n}
	⋮ A_i ⋮	N_{i1}	... N_{ij} ...		N_{in}
	A_n	N_{n1}	N_{nj}		N_{nn}

Figure 2.6: Confusion matrix [Deng et al. (2016)].

To increase the comprehensibility of the confusion matrix, the values for N_{ij} can be normalised by dividing each value with the number of observations in class A_i. This results in each value representing the proportion of accurate or inaccurate predictions within the given class. This normalised form of the confusion matrix has been used in this thesis.

Chapter 3

Methods

This chapter outlines the methodology of the thesis. It starts by detailing the data used in the project and the preprocessing techniques applied. Then, it covers feature engineering, training methods, feature selection and evaluation techniques. Results from the preprocessing and training process are presented in this chapter, while results directly related to the research questions are presented in Chapter 4.

3.1 Data

For this project a combination of weather forecast data and observational based weather data has been used as input in the machine learning models, and production data has been used as the response. Hourly data has been gathered from 2017 to 2022 for two different small scaled hydropower plants. In this section the data will be explained in detail.

3.1.1 Locations

The selection of the two power plants was based on their location and production patterns. Bjørgum power plant is located in Agder, the southernmost county in Norway. The climate in Agder varies from the coastal areas in the south to the mountainous inland areas. According to data obtained from Valle meteorological station situated close to Bjørgum power plant, the average annual temperature over the past ten years has been 5.6°C, and the average total precipitation received over a year has been 1160mm [*Klimaservicesenter* (2023)]. The catchment area, shown in Figure 3.1, is extensive at 38.5km^2 and comprises of 68% forest [*NVE* (2022)]. This leads to a relatively slow runoff, resulting in stable production. The intake is situated at 560 masl, which leads to the temperature dropping below freezing during the winter months. However, the power plant is often able to operate almost throughout the winter. This is due to the size of the catchment area and type of ground. As the catchment area consists mostly of forest as well as mire, precipitation is absorbed during autumn and stored under the snow during winter, maintaining a continuous baseflow to the river.

Furegardane power plant is situated in Vestland county on the west coast of Norway. Vestland county exhibits a diverse climate, ranging from a mild and humid climate near the coast to a drier climate inland by the fjords and mountains. Data collected from Bulken meteorological station, which is the closes station to Furegardane power plant, show that the average annual temperature over the past 10 years has been 6°C, and the average total precipitation over a year has been 1965mm [*Klimaservicesenter* (2023)].

Figure 3.1: Catchment area for Bjørgum power plant.

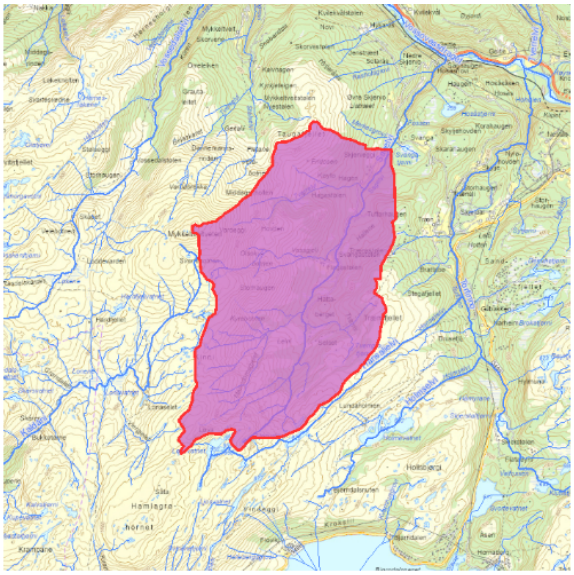


Figure 3.2: Catchment area for Furegardane power plant.

The catchment area, shown in Figure 3.2, is nearly one-third the size of Bjørgum power plant's catchment area and is made up of 49% bare rock. The intake is placed at an altitude of 310 masl, however, 50% of the catchment area is above 683 masl, with a maximum altitude of 1129 masl [*NVE* (2022)]. This leads to faster runoff, resulting in a more unstable production pattern with less operating hours than Bjørum power plant.

3.1.2 Production data

The production data is an hourly time series that represents the average power output [MW] of the power plant during each hour, which corresponds to the amount of electrical energy [MWh] produced during that time. There are available measurements of power output both on the generator inside the power plant and on the grid outside the power plant. The efficiency of the transformer and grid loss will lead to variations between these values, and the extent of the difference will depend on the placement of the measuring point. For this project, the grid measurements have been used as the imbalance costs are calculated based on these values. The data is sourced from Elhub through Småkraft.

As previously stated, the selection of the two power plants was based on their differing production patterns. The distribution of power output for Bjørgum and Furegardane are illustrated in Figure 3.3 and Figure 3.4 respectively. As depicted in Figure 3.3, more than half of the values for Bjørgum are less than 2MW and nearly 15% of the values exceed 5MW. The density plot for Furegardane (Figure 3.4) shows that a significant number of values are equal to zero. Calculations reveal that 43% of the values for Furegardane are equal to zero, while 10% of the values exceed 5MW. Hence, Bjørgum has more operating hours during a year than Furegardane with more of the power output in the mid-range.

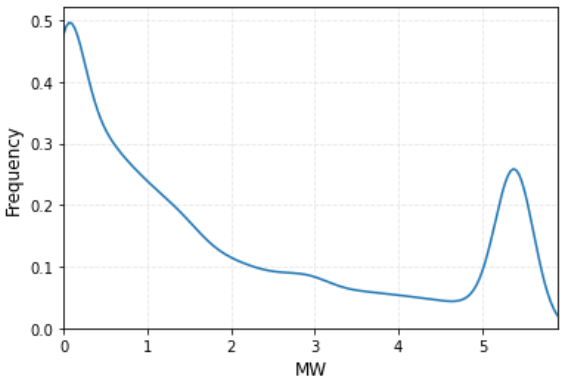


Figure 3.3: Density plot for the power output at Bjørgum power plant.

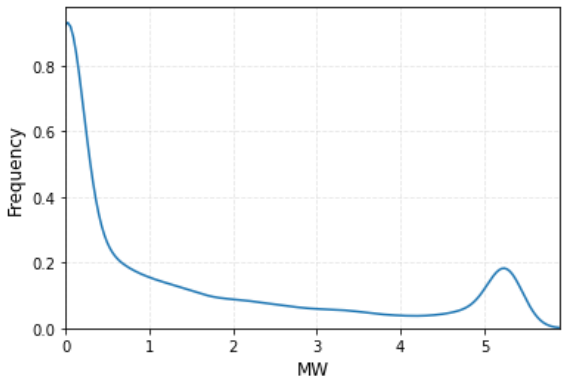


Figure 3.4: Density plot for the power output at Furegardane power plant.

Figure 3.5 and Figure 3.6 show the seasonal variations in production for Bjørgum and Furegardane respectively. The plots are created using a moving average of power output with a window of two months. Day 0 in the plot refers to the 1st of January. According to the plot for Bjørgum, production tends to be low in the beginning of the year, but rises during spring. The production then proceeds to drop in the summer months, before rising again in the fall. The plot for Furegardane reveals that the power plant follows a similar cycle as Bjørgum, but with more variation across the years. What remains consistent for Furegardane is that there is a clear peak in production around May and June.

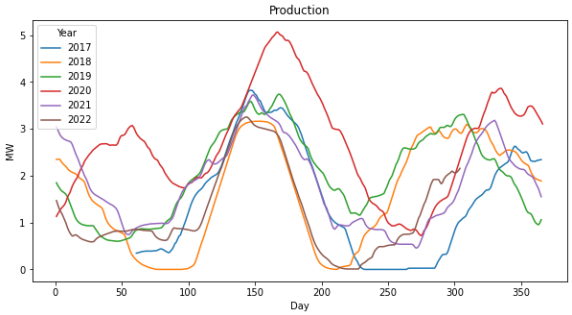


Figure 3.5: Plot showing seasonal variations of production at Bjørgum power plant.

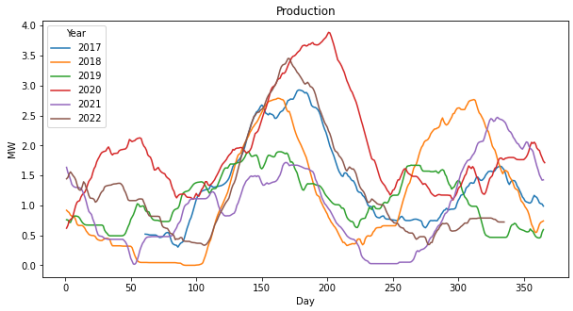


Figure 3.6: Plot showing seasonal variations of production at Furegardane power plant.

3.1.3 Forecast data

The weather forecast data was obtained from the Meterological Cooperation on Operational Numeric Weather Prediction (MetCoOp), a collaboration between the Finnish Meteorological Institute (FMI), the Meteorological Institute of Norway (MET), and the Swedish Meteorological and Hydrological Institute (SMHI). The forecast is generated by the MetCoOp Ensemble Prediction System (MEPS), which is an atmospheric ensemble model that covers Scandinavia and the Nordic seas. The model produces a 66-hour forecast at 00, 06, 12, and 18 UTC, with a 2.5km grid spacing [*Frogner et al.* (2019)]. To meet the requirement of power suppliers reporting their expected volumes for the next day at 1130 UTC, the forecast produced by MEPS at 06 UTC has been used in this thesis. Using 06 as the reference hour (hour 0), the forecast for hours 18 through 41 corresponds to the reporting period. To get a representative forecast for the entire catchment area, an average value of all grid boxes within a chosen radius from the center of the catchment area has been used.

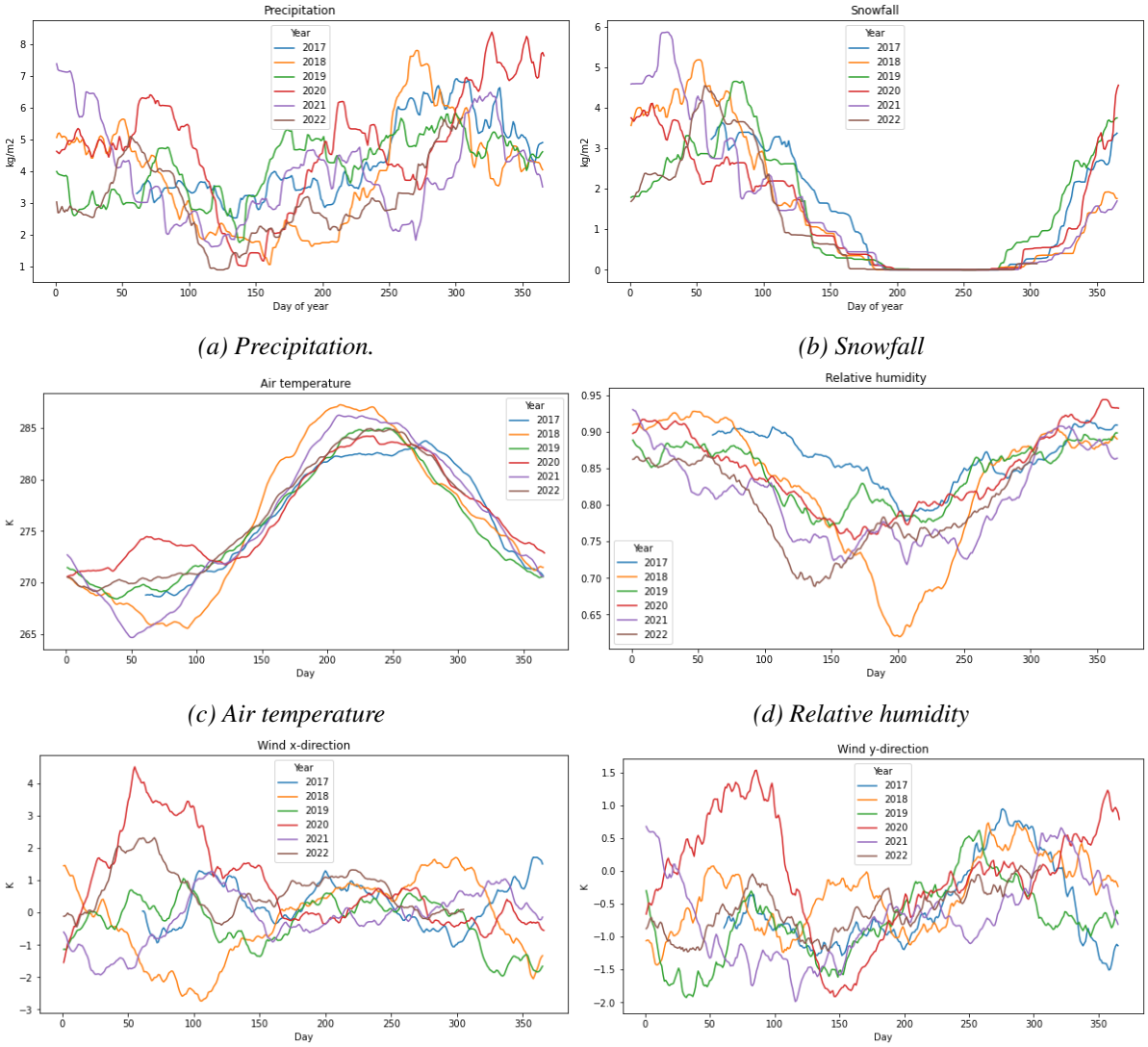
On February 4th, 2020, there was a structural change in the model output of MEPS. When selecting variables to use in the models, the change had to be taken into account as new variables were added while others were removed. The variables listed in Table 3.1 were chosen from the variables that were present both before and after the update. For the variables with the attribute *_acc* the values are accumulated from the start of the forecast, i.e. for each hour after 06 and until the next forecast at 06.

Table 3.1: Weather forecast variables used in this project.

Standard name	Long name	Unit
precipitation_amount_acc	Accumulated total precipitation	kg/m^2
snowfall_amount_acc	Total accumulated solid precipitation	kg/m^2
air_temperature_2m	Screen level temperature	K
relative_humidity_2m	Screen level relative humidity	1
x_wind	Zonal 10 metre wind	m/s
y_wind	Meridional 10 metre wind	m/s

The seasonal variations of the selected weather forecast variables for Bjørgum and Furegardane from 2017 to 2022 are illustrated in Figure 3.7 and 3.8, respectively, using a two-month moving average. Comparing the two figures, the seasonal cycles appear to be similar. The precipitation plot for Furegardane shows a more distinct cycle than that of Bjørgum, but both plots indicate a seasonal cycle with less precipitation in spring and more precipitation in the winter and autumn months. The two figures also reveal a notable seasonal cycle in snowfall and air temperature, with greater snowfall during winter and no snowfall in summer, and low air temperatures at the beginning of the year, a peak in summer, and a drop in fall. The relative humidity plots show a

weaker seasonal cycle with a slight reduction in the summer months, as supported by the Clausius - Clapeyron relationship [*Hartmann* (2015)]. The wind variables in both x and y-directions do not have a clear seasonal cycle, but there seem to be less variation across the years during summer months.



(a) Precipitation.

(b) Snowfall

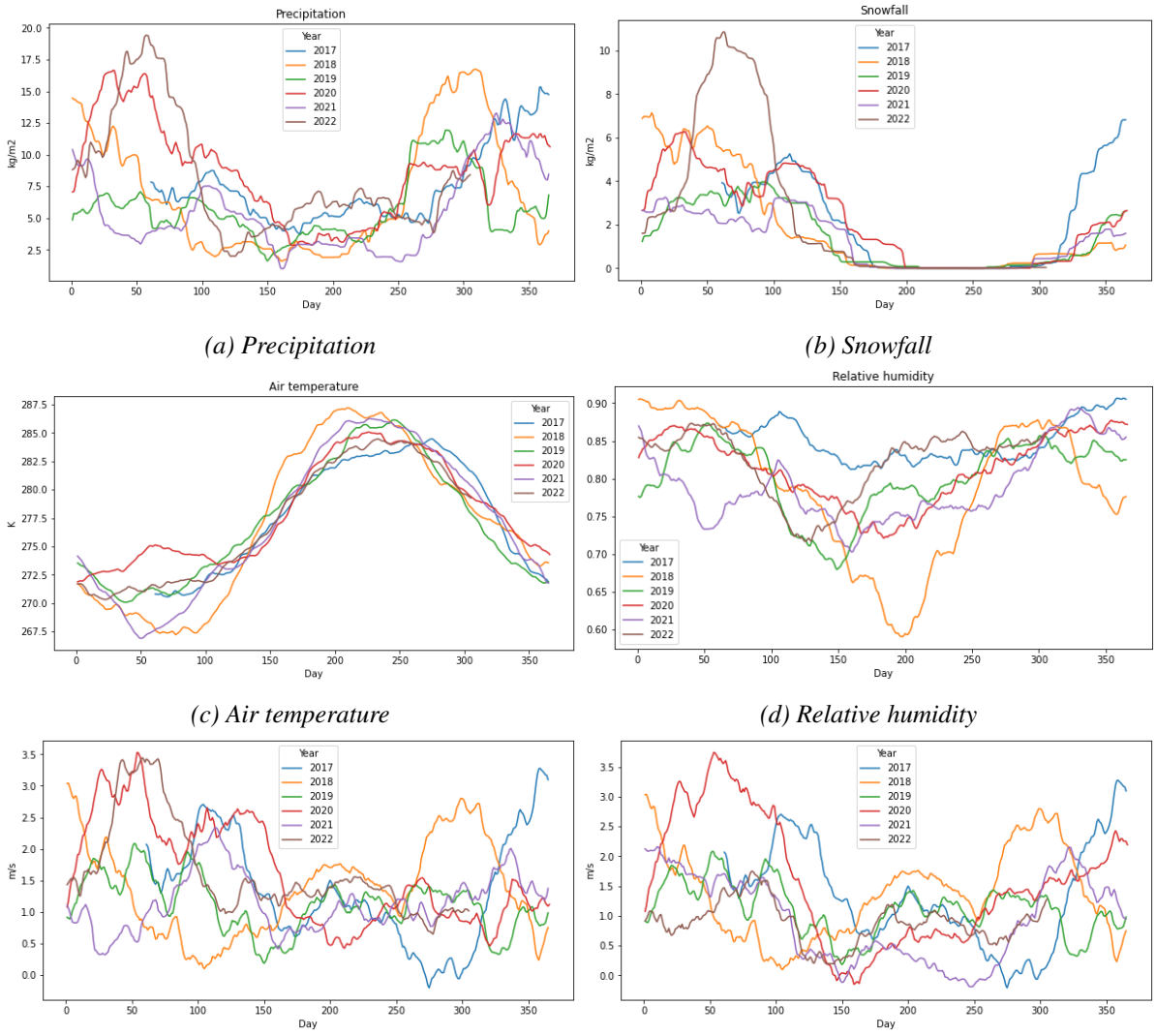
(c) Air temperature

(d) Relative humidity

(e) Wind x-direction

(f) Wind y-direction

Figure 3.7: Seasonal variations of weather forecast variables for Bjørgum power plant.



(a) Precipitation

(b) Snowfall

(c) Air temperature

(d) Relative humidity

(e) Wind x-direction

(f) Wind y-direction

Figure 3.8: Seasonal variations of weather forecast variables for Furegardane power plant.

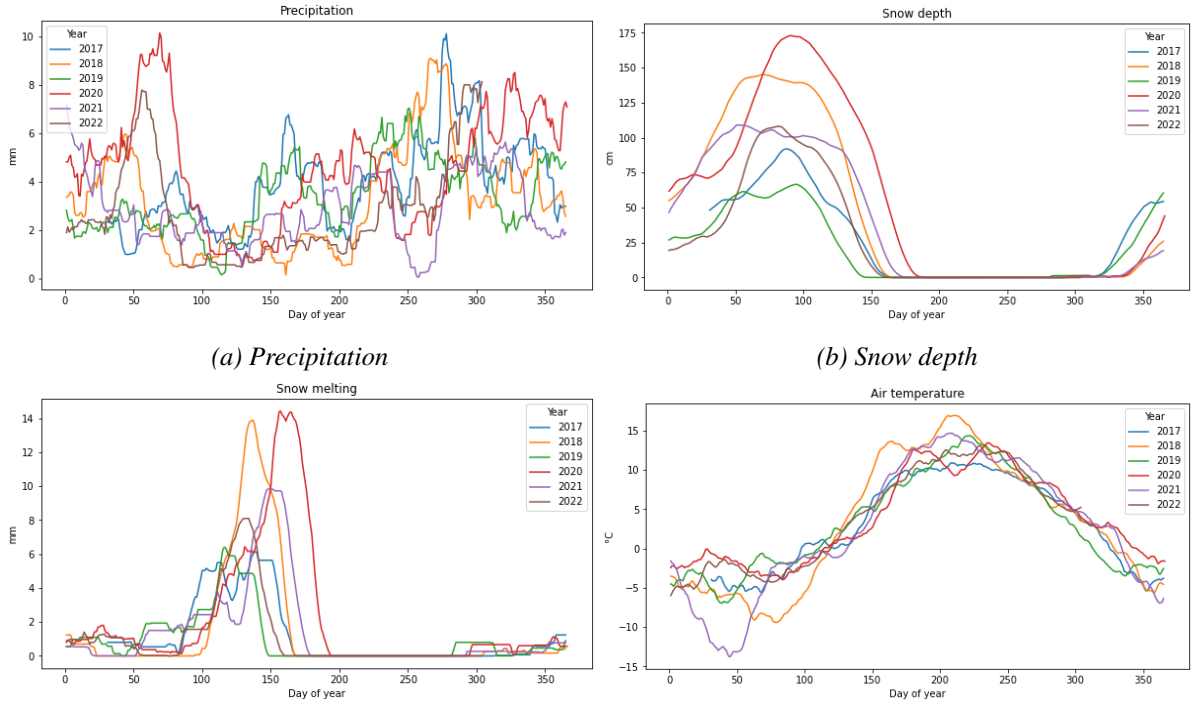
3.1.4 Observational based and modelled data

Observational based data for precipitation and temperature, as well as modelled data for snow depth and snowmelt have been included in the model. This data is sourced from Varsom SeNorge, a collaboration between NVE, MET, Statens Vegvesen, and Kartverket [*Varsom SeNorge*]. The observations for precipitation and temperature are interpolated to achieve a spatial resolution of 1km^2 using data from measuring stations throughout the country. The data is provided on a daily basis, with the temperature indicating the daily average and the precipitation representing the total amount received over the course of a day [*Lussana et al.* (2019)]. The estimation of snow data is based on the HBV-model that takes into account temperature and precipitation. Essentially, the model differentiates between snow and rain for temperatures below and above 0.5°C, and it estimates melting if temperatures rise above 0°C [*Saloranta* (2012)].

Table 3.2: Observational based and modelled variables used in this project.

Variable name	Description	Unit
rr	Total daily precipitation	mm
sd	Snow depth	cm
tm	Average daily temperature	$^{\circ}C$
qsw	Daily melted snow	mm

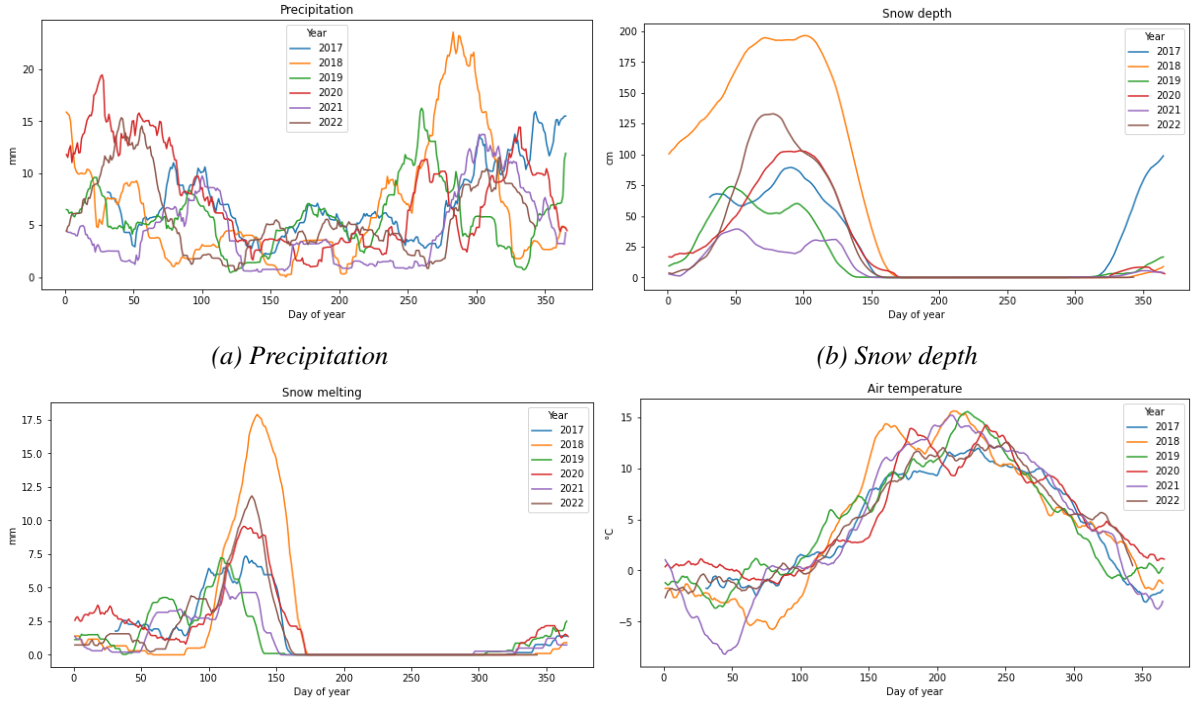
Figure 3.9 and 3.10 show the seasonal variations for the chosen observational based and modelled variables from SeNorge in the period 2017 to 2022 for Bjørgum and Furegardane, respectively. The plots are made using a moving average with a window of one month. As for the forecast variables, the seasonal cycle appear to be similar at both locations. Comparing the precipitation plots, Bjørgum experiences a shorter drop in precipitation during spring, while Furegardane's drop lasts longer into the summer. The amount of precipitation for Bjørgum ranges from 0mm to 10.2mm, while for Furegardane it ranges from 0mm to 23.6mm. The snow depth is at its highest during winter and non-existent in the summer months. The figures indicate that the amount of snow varies considerably from year to year. For Bjørgum, 2019 had the lowest snow depth during winter, with approximately 50cm, and 2020 had the largest snow depth, with a maximum of 175cm. For Furegardane, the lowest snow depth was in 2021, with about 25cm during the winter months, and 2018 had the largest snow depth, with around 200cm. The snow melting plots show that both locations have a peak in spring. Comparing the two plots, the timing of the peak varies more from year to year for Bjørgum than for Furegardane. Lastly, the air temperature at each location exhibits a clear seasonal cycle, with a peak during summer and a drop in the winter months.



(a) Precipitation *(b) Snow depth*

(c) Snow melting *(d) Air temperature*

Figure 3.9: Seasonal variations for observational based and modelled weather variables at Bjørgum power plant.



(a) Precipitation *(b) Snow depth*

(c) Snow melting *(d) Air temperature*

Figure 3.10: Seasonal variations for observational based and modelled weather variables at Furegardane power plant.

3.1.5 Missing data

Both the weather forecast and production datasets have some missing values. Additionally, missing values are created during engineering of accumulated features. The data is assumed to be missing completely at random, meaning that the reason for its absence is unrelated to both the values of the observed and unobserved data. To handle this, any rows containing NaN values were eliminated from the dataset. Another form of missing data occurs when there are hours with sufficient runoff for the power plant to operate, but for various reasons, no electricity is produced. This is classified as technical downtime, which could be caused by grid complications, operational restrictions, or technical problems with the power plant. To avoid inaccurate input to the models, these values have been removed. The values were identified by estimating the runoff for each hour, and removed if the runoff estimate exceeded a certain value. For Bjørgum power plant the sum of daily snowmelt and 72 hours of accumulated rainfall were used as a runoff estimate, and the threshold value was set to 25mm. This resulted in a reduction in the dataset of 1,477 rows. For Furegardane power plant the sum of daily snowmelt and 24 hours of accumulated rainfall were used as a runoff estimate, and the threshold value was set to 10mm. This resulted in a reduction in the dataset of 1,241 rows. The process of selecting threshold values involved plotting both the runoff and production data and examining which levels of runoff were associated with production.

As a result, Bjørgum's dataset decreased from 50,664 to 47,715 rows, and Furegardane's dataset decreased from 50,664 to 46,749 rows. Despite the reduction in size, the resulting datasets still have a sufficient number of samples.

3.2 Feature Engineering

The specifics of the catchment area play a role in determining how quickly and how much of the precipitation and melted snow reaches the power plant's intake. To account for this lagged effect, accumulated features have been created. In this section, the feature engineering will be explained.

3.2.1 Accumulated rainfall

The MET forecast variables provide hourly accumulated precipitation and snowfall during a day. To determine the hourly accumulated rainfall, the snowfall variable has been subtracted from the precipitation variable for each hour. Similarly, SeNorge provides observational based total daily precipitation without differentiating between snow and rain. Using the daily average temperature from SeNorge and a threshold value of 273.15K, variables for daily rainfall and snowfall amounts have been created.

Furthermore, an hourly rainfall variable has been created by subtracting the previ-

ous value, with the exception of the first value of the forecast which remains unchanged ($rain\_h = X_h - X_{h-1}$). From the hourly rainfall data, various intervals of accumulated rainfall have been created. This way, the model will have information on the total rainfall in the catchment area during these intervals leading up to the hour to be predicted. These features have been created as the rainfall events will have a delayed effect on the runoff. Thus, it is expected that the specifics of the catchment area will determine whether the model selects shorter or longer intervals of accumulated rainfall. To determine the most significant interval of accumulated rainfall for predicting production, the correlation between accumulated precipitation over n hours and production has been calculated. The results for are presented and discussed in Chapter 4: Results and Discussion.

Finally, a feature has been created to determine the accumulated precipitation over the course of a month based on the observational based data from SeNorge. This feature was created to reflect the degree of saturation of the ground at any given time. After a period of drought, a greater amount of precipitation will be required to produce the same level of runoff as compared to after a prolonged period of rainfall.

3.2.2 Accumulated snowmelting

As previously mentioned, seNorge provides estimated daily values for snowmelt in millimeters runoff. These values are not interpolated between measurements, but calculated based on precipitation and temperature. Since the predicted production will be given on an hourly basis while the snowmelt values are given on a daily resolution, the values have not been differentiated throughout the day. Instead, the value for the previous day is repeated for each hour of the next day. This allows the model to know how much snow melted the day before.

From the daily snowmelt variable, various intervals of accumulated melted snow have been created. As for the accumulated rainfall, the specifics of the catchment areas for the different power plants will determine whether the model selects shorter or longer intervals. Thus, the correlation between accumulated snowmelt over n days and production was calculated. When calculating the correlation, periods with no snowmelting were excluded from the analysis. This equaled 83% of the time for Bjørgum power plant and 79% of the time for Furegardane power plant. Since these instances dominate the data, their inclusion would significantly affect the results. The results for Bjørgum and Furegardane are presented and discussed in Chapter 4: Results and Discussion.

3.2.3 Time indicators

As seen in Figure 3.5 and Figure 3.6, the production data exhibits seasonal variations following the natural runoff. To account for this, features have been created to indicate the day of the year (*doy*) and month of the year (*moy*). The day of the year is inherently

represented by a range of 1 to 365, while the month of the year spans from 1 to 12. To achieve cyclicality, such that January 1st is next to December 31st, the features have been split into two parts using sine and cosine functions. The functions are as follows:

$$day\_sin = sin(2\pi * doy/365) \quad month\_sin = sin(2\pi * moy/12)$$
$$day\_cos = cos(2\pi * doy/365) \quad month\_cos = cos(2\pi * moy/12)$$

3.2.4 Production trend

During periods of consistent weather conditions or in large catchment areas, there may be small changes in the production level from day to day. To provide the model with an estimate of the production level, a feature has been created that represents the average production during the six hours leading up to the reporting time. Specifically, the average production from 06 to 11 has been used:

$$prod\_trend = \frac{1}{6}\sum_{n=6}^{11} h_n$$

3.2.5 Final dataset

Figure 3.11 displays a timeline indicating when the input variables are availible relative to the reporting time and forecast period. The observational based and modelled variables obtained from SeNorge are calculated based on measurements taken between 07:00 on day 1 and 07:00 on day 2. The most recent version of the MEPS forecast for day 3 becomes available at 06:00 on day 2, five and a half hours before the reporting time, which is displayed at 11:30 on day 2. The forecast period is indicated on day 3.

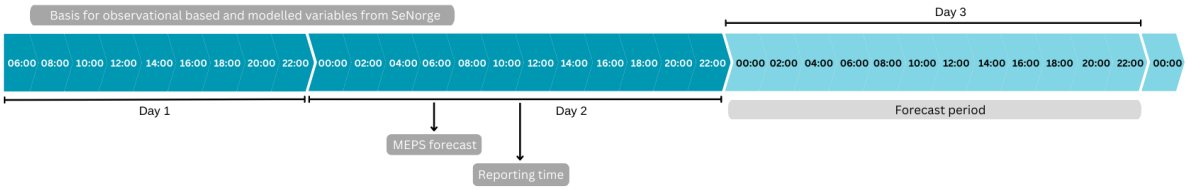


Figure 3.11: Timeline showing when the input variables are available relative to the reporting time and the forecast period.

Table 3.3 summarise all features in the final dataset.

Table 3.3: Complete list of features with units and source.

Feature	Unit	Source	Feature	Unit	Source
prod_trend	MWh	Elhub (measured)	rainfall_acc_24h	mm	MEPS forecast
precipitation_amount_acc	kg/m^2	MEPS forecast	rainfall_acc_30h	kg/m^2	MEPS forecast
snowfall_amount_acc	kg/m^2	MEPS forecast	rainfall_acc_48h	kg/m^2	MEPS forecast
air_temperature_2m	$°C$	MEPS forecast	rainfall_acc_60h	kg/m^2	MEPS forecast
relative_humidity_2m	1	MEPS forecast	rainfall_acc_72h	kg/m^2	MEPS forecast
x_wind_10m 1	m/s	MEPS forecast	rainfall_acc_84h	kg/m^2	MEPS forecast
y_wind_10m	m/s	MEPS forecast	rainfall_acc_96h	kg/m^2	MEPS forecast
month_sin			rr(mm)	mm	SeNorge observational based
month_cos			tm(Celcius)	$°C$	SeNorge observational based
day_sin			sd(cm)	cm	SeNorge modelled
day_cos			qsw(mm)	mm	SeNorge modelled
rainfall_amount_acc	kg/m^2	MEPS forecast	snow_obs	mm	SeNorge observational based
precipitation_amount_inst	kg/m^2	MEPS forecast	rain_obs	mm	SeNorge observational based
snowfall_amount_inst	kg/m^2	MEPS forecast	rr_acc_month	mm	SeNorge observational based
rainfall_amount_inst	kg/m^2	MEPS forecast	snowmelt_2day	mm	SeNorge modelled
rainfall_acc_week	kg/m^2	MEPS forecast	snowmelt_3day	mm	SeNorge modelled
rainfall_acc_3h	kg/m^2	MEPS forecast	snowmelt_4day	mm	SeNorge modelled
rainfall_acc_6h	kg/m^2	MEPS forecast	snowmelt_5day	mm	SeNorge modelled
rainfall_acc_12h	kg/m^2	MEPS forecast	snowmelt_6day	mm	SeNorge modelled
rainfall_acc_18h	kg/m^2	MEPS forecast	snowmelt_7day	mm	SeNorge modelled
rainfall_acc_22h	kg/m^2	MEPS forecast			

3.3 Training process

For this project three different multivariate regression models have been used to predict the next day's production. All models are trained, validated and tested on subsets of data from 2017 to 2022. In this section, the models will be introduced, the general training process used for the models will be explained, and choices made for the specific models will be commented on. Figure 3.12 illustrates the total process from splitting of data to selecting the best machine learning model.

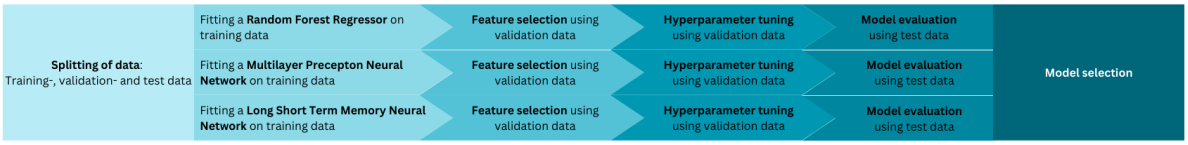


Figure 3.12: Flow chart showing the steps and data used during training, model evaluation and model selection.

3.3.1 Models

The three models used in this project are a Random Forest Regressor (RFR), a Multilayer Precepton neural network (MLP) and a Long Short Term Memory neural network (LSTM). *Sklearn′s* RandomForestRegressor (RFR) was used for the random forest algorithm. The model is built by training several decision trees on different bootstrap samples of the dataset and averaging the output of each tree. To determine the best split at each node only a subset of features is considered. Building trees from subsets of samples and using subsets of features for each split enhances the diversity of the forest and reduces overfitting [*scikit learn* (a)].

Sklean′s MLP Regressor was used to train a multilayer precepton neural network. Given a set of features X and a target y the model can learn a non-linear function approximator. The network consists of at least four layers: an input layer, two or more hidden layers, and an output layer. Neurons in the hidden layers transform values from the previous layer using weighted summation and non-linear activation functions. The output layer receives the last hidden layer's values and transforms them into output values [*scikit learn* (b)].

Finally, the LSTM model has been created using *Keras′* deep learning library in Python. An LSTM is a type of recurrent neural network (RNN) designed to process sequential data such as time series data. It is capable of capturing long-term dependencies in the data and is composed of memory cells that can store information for an extended period. The LSTM architecture is designed to selectively forget or retain information from previous time steps, allowing it to model complex patterns and relationships in the data. Both the LSTM and MLP neural network are sensitive to the scale of the features, thus the whole dataset had to be scaled such that the values for each variable range from 0 to 1. More detailed description of the models are given in Chapter 2: Theory.

3.3.2 Splitting of data

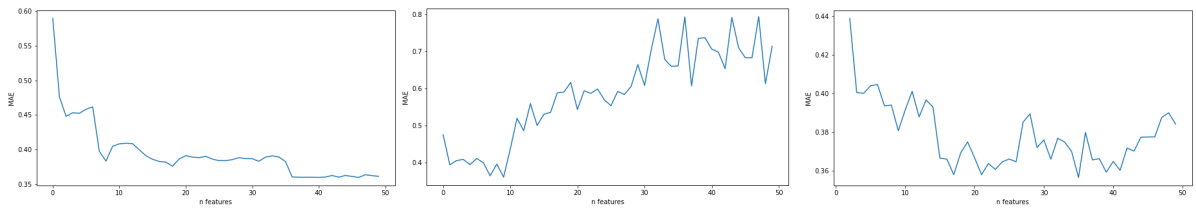
Before the training process, the dataset was divided into three distinct parts: training-, validation-, and test data. The training data consists of data from 2017 to February 2021, while the validation data is from March 2021 to December 2021, and the test data is from January 2022 to November 2022. The sequential splitting was done to prevent autocorrelation issues. If the data was split randomly there is a high chance that the model would be trained on data points nearly identical to those it would be tested on, leading to biased performance metrics that do not accurately represent reality. In March 2021 Småkraft started using a new reporting solution. For comparison purposes the validation data begins in March instead of January. Finally, each dataset was divided into a matrix X containing all feature data and a vector y containing the production values. The models are trained using the X matrix as input and the y vector as the response.

3.3.3 Feature selection

Eliminating less significant features from a model and retaining only the essential ones can enhance the model's efficiency and increase accuracy. The optimal approach to feature selection differs among the models, and various techniques have been experimented with. Initially, default hyperparameters were used to train each model, and feature selection was performed on these default models. Ideally, feature selection and hyperparameter tuning should be performed simultaneously to identify the optimal combinations. However, doing so is likely to result in only a marginal improvement in performance. As a result, for the sake of simplicity, the two procedures have been carried out independently.

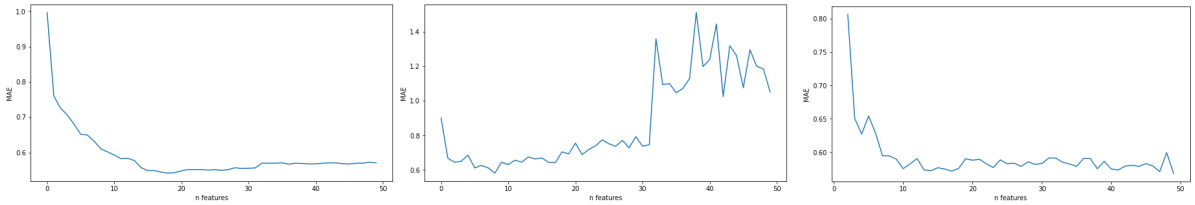
The first feature selection process used was a built-in function in the random forest algorithm. *Sklearn's* random forest module provides an attribute named *feature importance* which evaluates the significance of each feature when predicting the target variable. For each split in the decision trees, the feature leading to the greatest reduction of the criterion (often RSS) is selected. The goal for regression problems when building decision trees is to reduce the variance in the target variable. The average decrease brought by each feature across all trees can be measured and indicate its performance [*scikit learn* (a)]. Once all features had been evaluated, they were sorted in descending order based on their importance, with the most important feature at the top of the list. The result is presented in Chapter 4: Result and Discussion.

This list was used to determine the optimal number of features in the three models. Each model was fit using all features, and the resulting mean absolute error (MAE) was computed using validation data. The feature with the lowest importance was eliminated, and the MAE was recalculated using the reduced set of features. This process was repeated until only one feature remained. The result for each model for Bjørgum and Furegardane are illustrated in Figure 3.13 and Figure 3.14, respectively.



(a) Random forest *(b) MLP neural network* *(c) LSTM neural network*

Figure 3.13: Mean absolute error using n features for Bjørgum power plant.



(a) Random forest *(b) MLP neural network* *(c) LSTM neural network*

Figure 3.14: Mean absolute error using n features for Furegardane power plant.

Another feature selection method used was *Sklearn′s* recursive feature elimination (RFE). This technique progressively reduces the feature set by recursively eliminating a certain number of features. In each iteration, the model is trained on the current feature set, the features are given an importance score, and the least important feature(s) is removed. Thus, this approach acknowledges that a features importance may change as other features are eliminated. Using *Sklearn′s* RFE-model, an estimator that provides information about feature importance needs to be specified. For this reason, the random forest algorithm was used. The model also takes as input the number of features to remove in each iteration. As the algorithm is quite extensive, this number was set to two to increase efficiency. Additionally, the user must input the desired number of features to be selected which was determined by analysing the plots in Figure 3.13 and Figure 3.14. Accordingly, three recursive feature selection processes were performed for each power plant, one for each of the models' optimal number of features.

The last feature selection process was carried out using domain knowledge. The dataset for each power plant was chosen based on knowledge about meteorology and hydrology at the two different locations, and was restricted to the most basic features. Thus, the whole feature selection process resulted in 14 datasets, as summarised in Table 3.4. The MAE for each model using the different datasets were calculated on the validation data, and the values are presented in Chapter 4: Results and Discussion. The different datasets with specified features used in each one are included in the appendix.

Table 3.4: Datasets created after feature selection.

Power plant	Model	Method	n features
Bjørgum	Radom Forest Regressor	*Sklearn's feature importance*	36
Bjørgum	MLP neural network	*Sklearn's feature importance*	9
Bjørgum	LSTM nerual network	*Sklearn's feature importance*	16
Bjørgum	Random Forest Regressor	*Sklearn's recursive feature selection*	36
Bjørgum	MLP neural network	*Sklearn's recursive feature selection*	9
Bjørgum	LSTM nerual network	*Sklearn's recursive feature selection*	16
Bjørgum		Domain knowledge	9
Furegardane	Radom Forest Regressor	*Sklearn's feature importance*	18
Furegardane	MLP neural network	*Sklearn's feature importance*	8
Furegardane	LSTM nerual network	*Sklearn's feature importance*	18
Furegardane	Radom Forest Regressor	*Sklearn's recursive feature selection*	18
Furegardane	MLP neural network	*Sklearn's recursive feature selection*	8
Furegardane	LSTM nerual network	*Sklearn's recursive feature selection*	18
Furegardane		Domain knowledge	9

3.3.4 Hyperparameter tuning

Each machine learning algorithm has its own set of hyperparameters that limit its behavior and learning. Optimal performance and accuracy of the algorithm can be achieved by selecting the appropriate hyperparameters. These parameters have been determined using domain knowledge alongside an iterative process of testing different values for the parameters. Although machine learning algorithms have numerous hyperparameters, only a subset of these have been tuned. The selection of hyperparameters and values to be tested for each one is determined using domain knowledge, and an algorithm is utilised to iterate through each combination of values. Tables 3.5 - 3.7 showcase the hyperparameters that were tuned for each of the three algorithms, along with the corresponding values tested and the selected values for each power plant. For each tested combination, the mean absolute error was calculated using validation data and the set of hyperparameter values resulting in the lowest mean absolute error was returned by the algorithm.

Table 3.5: Tuned hyperparamters for the random forest algorithm.

Hyperparameter	Default value	Values tested	Bjørgum	Furegardane
n_estimators	100	[50, 100, 200, 300, 500]	300	300
max_depth	None	[5, 10, 15, 20, 50, 100]	50	10
criterion	'squared_error'	['squared_error', 'absolute_error', 'friedman_mse', 'poisson']	'friedman_mse'	'poisson'

Table 3.6: Tuned hyperparamters for the MLP neural network.

Hyperparameter	Default value	Values tested	Bjørgum	Furegardane
hidden_layer_sizes	(100,)	[(1,),(5,),(50,),(60,),(70,),(80,)]	(70,)	(50,)
activation	'relu'	['identity', 'logistic', 'tanh', 'relu']	'tanh'	'relu'
solver	'adam'	['lbfgs', 'sgd', 'adam']	'sgd'	'adam

Table 3.7: Tuned hyperparamters for the LSTM neural network.

Hyperparameter	Default value	Values tested	Bjørgum	Furegardane
units	50	[5, 10, 50, 100, 200]	10	150
batch_size	32	[32, 64, 128]	64	64
epochs	10	[10, 50, 100, 200, 500]	100	500

*the default value for units and epochs for LSTM was manually selected.

3.4 Model Selection

So far, the validation dataset has been used to assess the performance of the models. At this point, the final models have been chosen and the test dataset will be used to provide objective estimates of skill. In the process of feature selection and hyperparameter tuning, the MAE was the sole metric employed. However, during the final model selection process, additional criteria such as mean squared error (MSE) and f1-score were taken into account. The difference between MSE and MAE is that MSE gives more weight to larger differences between predicted and true values due to the squaring of the differences. In order to apply the f1-score the predicted values were categorised into 25% intervals of maximum power: {0}, (0, 25%], (25%, 50%], (50%, 75%], and (75%, 100%]. This allows for assessing the model's performance under various scenarios, rather than solely focusing on its overall performance. It is of interest to see if

the models are able to predict each class well and not only the most represented classes as the distribution of values will vary from year to year.

To further investigate the models' performance, a confusion matrix were created together with plots of classification error. A confusion matrix is a matrix of size n x n, where n is the number of classes. The matrix shows the performance of a model by comparing the predicted labels with the actual labels, and summarises the number of correct and incorrect predictions made by the model. If all values are correctly classified, the confusion matrix will have only the diagonal cells filled with values, and all off-diagonal cells will be zero. A further description of the confusion matrix is given in Chapter 2: Theory. The classification error plots represent the same information in a bar plot, where each bar represents the amount of actual values in a class and is divided by colors to display the the share of correctly and wrongly predicted classes. The final model for each location was selected based on the MAE together with an evaluation of its performance on different intervals of power output.

3.5 Model Evaluation

As stated in the introduction, the primary goal of the thesis is to improve the accuracy of forecasting hourly production for the next day compared to Småkraft's current method. Hence, when evaluating the final machine learning model, the model has been compared to the performance of Småkraft's existing solution. Additionally, a simple baseline model was created to serve as a reference point for assessing the performance of both the existing method and the machine learning model. The simple baseline model can also help understand if the benefit is worth the cost. To create the baseline model the actual production was shifted back one day, meaning that the prediction for the next day equals the previous day's production, hour by hour. In order to compare the different methods, the same metrics used in the model selection were used, i.e. MAE, MSE and f1-score. Additionally, scatter plots, confusion matrices and classification error plots were created to further investigate the performance of the three methods.

Following a comparison between the chosen machine learning model and the two other available methods, an evaluation of the machine learning model's performance across various weather scenarios was carried out. This involved plotting the predicted and actual production using the test dataset over the entire time series spanning from January 2022 to November 2022. Subsequently, specific periods from each season were selected for further investigation.

Chapter 4

Results and Discussion

The main objective of the thesis is to enhance the precision of predicting hourly production for the upcoming day compared to the existing method used by Småkraft. The reporting time for the predictions is at 11:30 and covers the time span from 00 to 24 the next day, as illustrated in Figure 3.11. This chapter will investigate the performance of the models and provide answers to the research questions. The structure of the chapter will be as follows: Firstly, an analysis of the most significant time intervals will be presented and discussed. Secondly, the results obtained from the feature selection process will be presented. Lastly, the evaluation of the models will be presented together with an examination of the models' performance in different weather situations.

4.1 Time Interval Analysis

One of the research questions concerns the time intervals for weather variables that affect the runoff and production the most. To investigate this, the correlation between hourly accumulated rainfall and production, as well as the correlation between daily accumulated snowmelting, was calculated. The findings are illustrated in Figures 4.1 through 4.4.

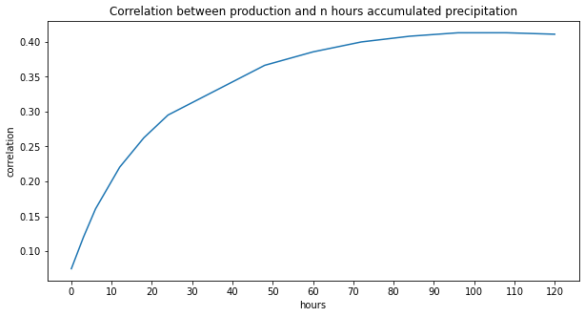


Figure 4.1: Correlation between n hours accumulated rainfall and production for Bjørgum power plant.

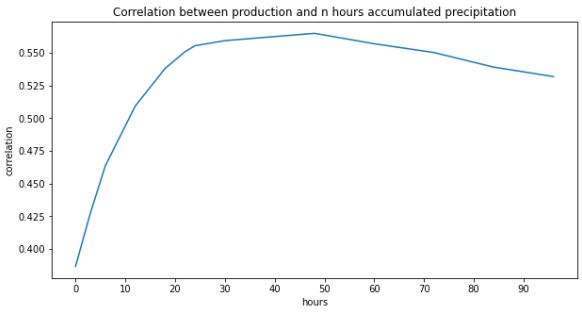


Figure 4.2: Correlation between n hours accumulated rainfall and production for Furegardane power plant.

Figure 4.1 and Figure 4.2 indicate that for Bjørgum a longer interval of accumulated rainfall will be important, while for Furegardane a shorter interval will be more relevant. This aligns with the characteristics of the two catchment areas. With a larger catchment area and more forest, the impact of rainfall on the runoff at Bjørgum is expected to be delayed. In contrast, Furegardane has a smaller catchment area with more bare rock, leading to a quicker response to rainfall. It is noteworthy that the correlation values in the Bjørgum plot range from 0.1 to 0.4, while those in the Furegardane plot range from 0.4 to 0.55. This indicates that more information is available in the accumulated intervals for Bjørgum compared to Furegardane. The plots offer useful

information about the lagged effects of runoff in the catchment area, but do not necessarily reveal the optimal value of n for predicting production. The relationship between precipitation and production may not be entirely linear, and the accumulated values will have a smoother curve leading to higher correlation.

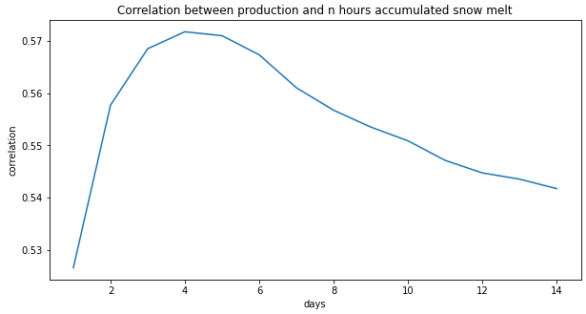


Figure 4.3: Correlation between n days of accumulated snow melt and production for Bjørgum power plant.

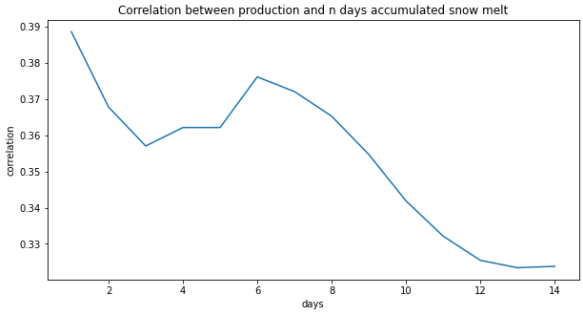


Figure 4.4: Correlation between n days of accumulated snow melt and production for Furegardane power plant.

The correlation between n days of accumulated snowmelt and production is presented in Figure 4.3 and Figure 4.4. In the case of Bjørgum, the strongest correlation of 0.57 is observed after 4 days. For Furegardane, the correlation decreases from day 1 before it increases from day 3 to day 6. However, the variation in correlation during the first six days ranges from 0.36 to 0.39, which is not considered a significant difference. Although the range of correlation values in either plot is small, it can indicate whether or not daily accumulated snowmelt will provide additional information to the model. It is important to note that, similar to accumulated rainfall, correlation may not reflect the best value of n for production forecasting. A further investigation on this will be presented in the section covering feature importance.

Figure 4.5 and Figure 4.6 illustrate the relationship between production and estimated runoff for Bjørgum and Furegardane, respectively. The production is plotted in green with MWh as unit, while the estimated runoff is plotted in blue with mm as unit. The estimated runoff for Bjørgum was computed using a cumulative rainfall interval of 96 hours and a cumulative snowmelt interval of 4 days. For Furegardane, the runoff estimate was calculated using a cumulative rainfall interval of 48 hours and the previous day's snowmelt amount. This estimation, using the optimal values for accumulated variables, appears to provide a satisfactory representation of the production trend. However, based on the plots, there seems to be a benefit in providing the model with additional information.

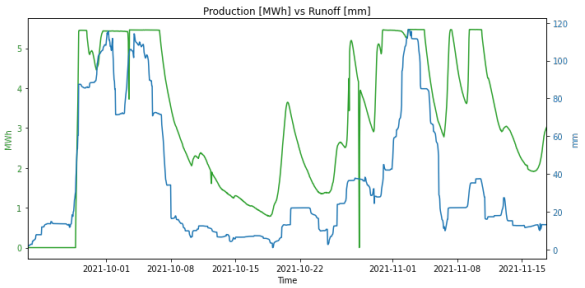


Figure 4.5: Relationship between runoff and production for Bjørgum.

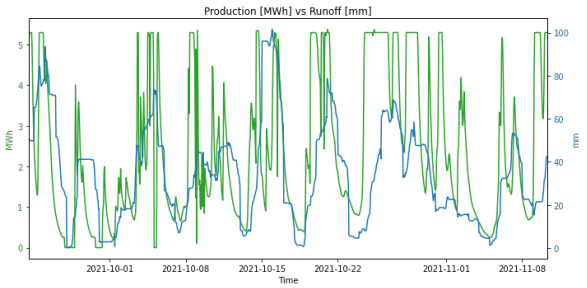


Figure 4.6: Relationship between runoff and production for Furegardane.

4.2 Feature Selection

The second research question presented in the introduction aimed to identify the key features required for predicting the following day's production. To achieve this, three different feature selection techniques were employed for each model: Sklearn's Feature Importance, Sklearn's Recursive Feature Selection, and using domain knowledge. In this section, the outcomes of Sklearn's Feature Importance approach and the final results from the feature selection process will be presented and discussed.

4.2.1 Feature importance

The results using *Sklearn's Feature Importance* can easily be visualised by sorting the features from most to least important, and are presented in Figure 4.7 and Figure 4.8 for Bjørgum and Furegardane, respectivly. As a reminder, a subset for each model was created using the top n important features, where n was chosen based on the plots in Figure 3.13 and Figure 3.14. The same was done for *Sklearn's Recursive Feature Selection*. However, these results are not as easily visualised since the feature ranking varies for each value of n. The subsets created using *Sklearn's Recursive Feature Selection* are presented in the appendix, together with the subsets created using domain knowledge.

Comparing the two plots, the most important feature for both locations is the "average_prod", which represents the average production during the six hours leading up to the reporting time. However, this feature is more significant for Bjørgum than Furegardane. The x-axis for Bjørgum ranges from 0 to 0.8, while for Furegardane it ranges from 0 to 0.4. As a result, other features at Bjørgum become relatively less important, while at Furegardane, several other features have some significance. This difference can be attributed to the characteristics of the catchment areas. Bjørgum has a large catchment area with a substantial forest cover which causes a delay in the runoff, resulting in less variation in production from day to day. Hence, the "average_prod" feature will often be a good representation for the next day's production. Furegardane, on the other hand, has a smaller catchment area with a lot of bare rock, leading to more rapid changes in production. Therefore, the production on one day may not be a good representation of the following day's production.

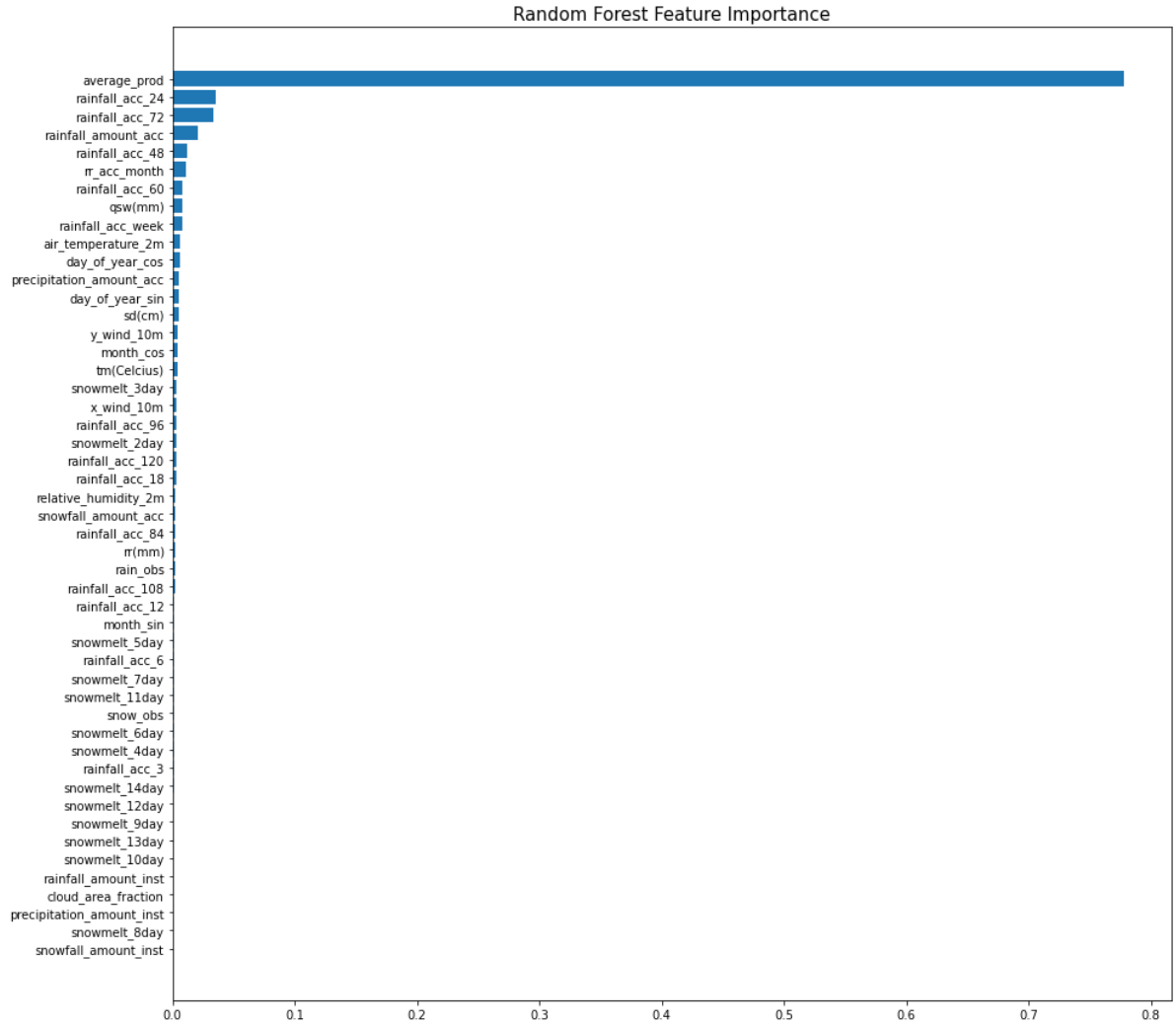


Figure 4.7: Feature importance using the random forest algorithm for Bjørgum.

Both locations consider intervals of accumulated rainfall as the second and third most important feature. Bjørgum places the most importance on 24 and 72 hour intervals, while Furegardane considers 24 and 18 hour intervals to be the most significant. This outcome aligns with the time interval analysis considering that Bjørgum selects longer intervals than Furegardane. For accumulated snowmelt features, both locations select the shortest interval available as the most significant. Additionally, both locations also consider observational-based rainfall during the last month ('rr_acc_month') as one of the most important features. This feature alone does not directly provide the model with information on the next days runoff. Rather, it offers insights into how the runoff will react to other variables. For example, if 'rr_acc_month' is small (indicating unsaturated ground), the runoff's response to rainfall is expected to be lower than if 'rr_acc_month' is large (suggesting saturated ground).

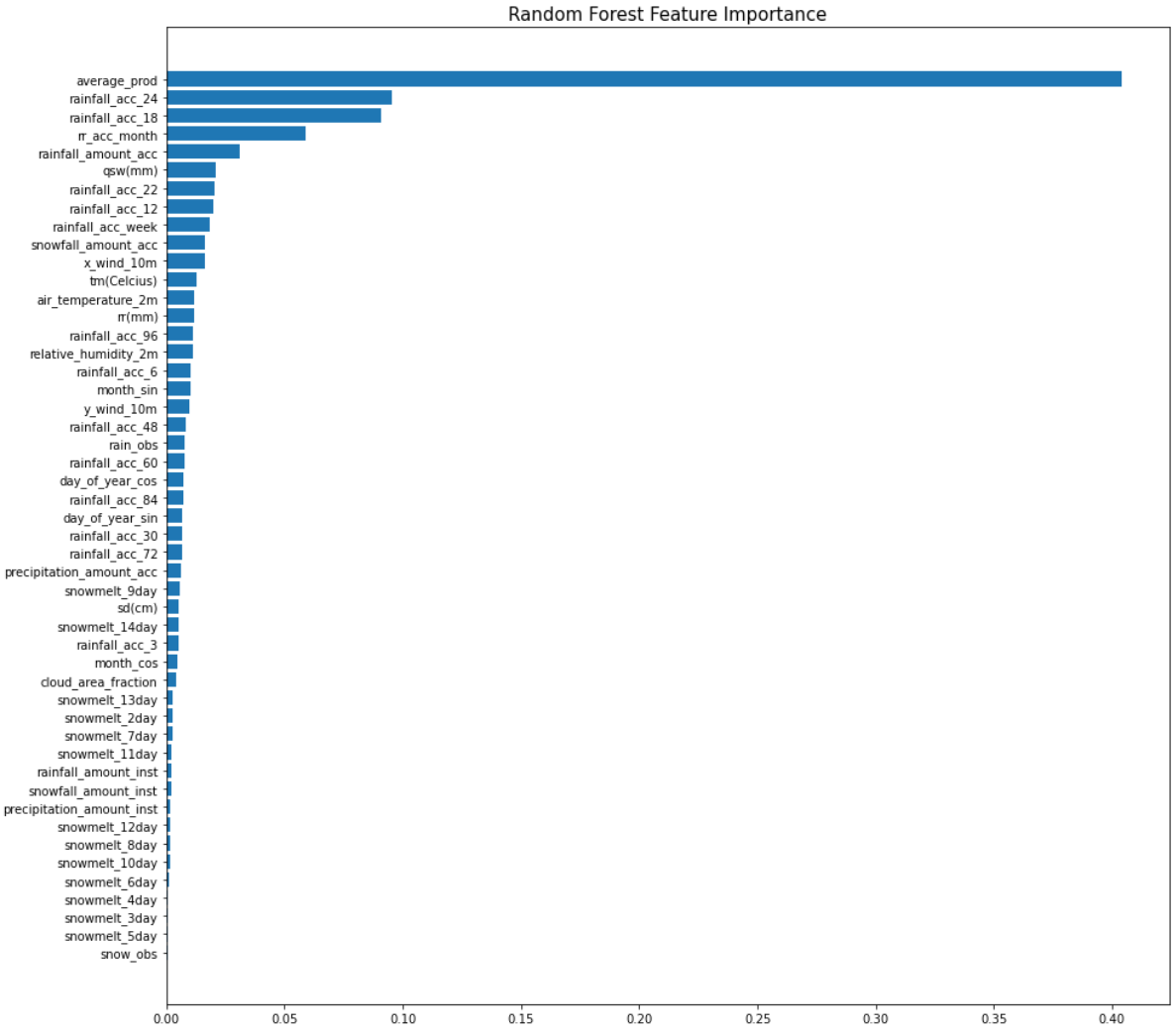


Figure 4.8: Feature importance using the random forest algorithm for Furegardane.

4.2.2 Feature Selection Results

Table 4.1 and Table 4.2 display the MAE values obtained for each model using the subsets derived from the feature selection process, as presented in Table 3.4. The MAE values for each model using the complete dataset have been calculated for comparison purposes. The subsets that yielded the lowest MAE for each model are denoted in bold font and were chosen for hyperparameter tuning.

Table 4.1: Mean absolute error [MWh] using different subsets with the Random forest regressor (RFR), Multi layered precepton neural network (MLP) and Long short term memory neural network (LSTM) algorithms for Bjørgum. The number of features in each subset is denoted in parenthesis.

	MAE			
Model	**Total dataset**	**Feature importance**	**Recursive feature elimination**	**Domain knowledge**
RFR	0.360 (49)	**0.359** (36)	0.361 (36)	0.383 (9)
MLP	0.521 (49)	0.337 (9)	0.345 (9)	**0.325** (9)
LSTM	0.369 (49)	**0.364** (16)	0.389 (16)	0.459 (9)

Table 4.2: Mean absolute error [MWh] using different subsets with the Random forest regressor (RFR), Multi layered precepton neural network (MLP) and Long short term memory neural network (LSTM) algorithms for Furegardane. The number of features in each subset is denoted in parenthesis.

	MAE			
Model	**Total dataset**	**Feature importance**	**Recursive feature elimination**	**Domain knowledge**
RFR	0.558 (49)	0.551 (18)	**0.550** (18)	0.551 (9)
MLP	0.742 (49)	0.601 (8)	0.642 (8)	**0.565** (9)
LSTM	0.593 (49)	**0.571** (18)	0.575 (18)	0.600 (9)

Table 4.1 shows that for Bjørgum the subset created using *Sklearn's feature importance* resulted in the lowest MAE for both the random forest regressor and the LSTM neural network, while the subset created based on domain knowledge gave the lowest MAE for the MLP neural network. Table 4.2 shows that for Furegardane using the subset created with *Sklearn's recursive feature elimination* resulted in the lowest MAE for the random forest regressor, while *Sklearn's feature importance* yielded the lowest MAE for the LSTM neural network. Finally, the MLP neural network had the lowest MAE using the subset created based on domain knowledge.

The results indicate that there is little difference in MAE values for the random forest algorithm when using different subsets. Although it was expected that the recursive feature elimination method would result in the lowest MAE, the difference between using this method and feature importance is minimal, and either subset is likely to yield similar model performance. In contrast, selecting the optimal subset has a clear advantage for the MLP neural network. The domain knowledge subset resulted in significantly lower MAE values for both locations. As the domain knowledge subset contains only the most basic features, it is apparent that the model is sensitive to noise. Lastly, while the LSTM neural network benefits from selecting the best subset, the difference in MAE is not as significant as for the MLP neural network. Lists of features included in each dataset are available in the appendix.

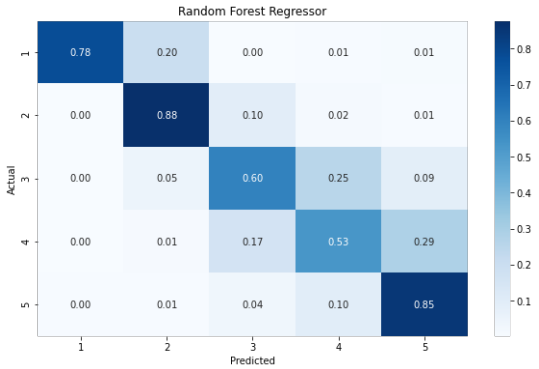
4.3 Model Selection

The MAE, MSE, and f1-score for each model were calculated using the test dataset after the feature selection process and hyperparameter tuning. Table 4.3 presents the results for Bjørgum and Furegardane, where the best score for each metric is highlighted in bold font. The table shows that, with the exception of the f1-score for Bjørgum, the LSTM neural network outperformed the other models at both locations. At Bjørgum, the random forest regressor had the worst scores for MAE and MSE but the best score for f1. This suggests that the LSTM neural network performs well on the most represented classes, whereas the random forest regressor performs better for the under-represented classes. To further investigate the models' performance, confusion matrices and plots of classification error are presented in Figure 4.9 and Figure 4.10.

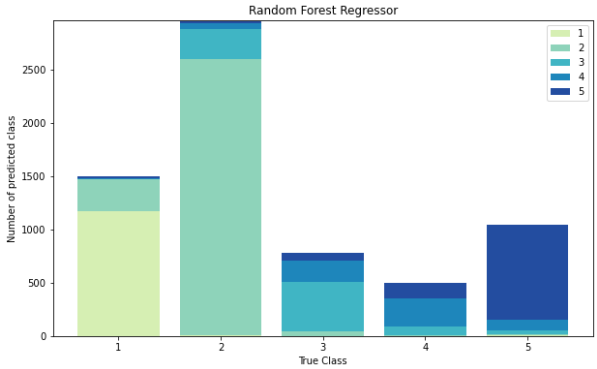
Table 4.3: Calculated mean absolute error [MWh], mean squared error [MWh^2] and f1-score [ ] for the random forest regressor, the MLP neural network and the LSTM neural network for Bjørgum and Furegardane.

	Bjørgum			Furegardane		
Model	**MAE**	**MSE**	**f1-score**	**MAE**	**MSE**	**f1-score**
RFR	0.395	0.573	**0.720**	0.713	1.198	0.435
MLP	0.372	0.483	0.653	0.689	1.097	0.354
LSTM	**0.316**	**0.453**	0.695	**0.573**	**0.942**	**0.458**

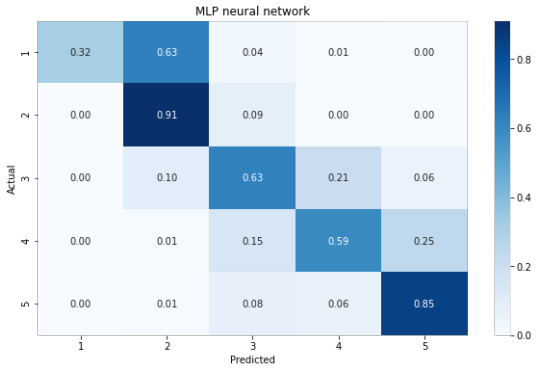
As explained in Section 3.4, the predictions were categorised into five intervals of power output. Figure 4.9 displays the confusion matrices and classification error plots for the random forest regressor, MLP neural network, and LSTM neural network for Bjørgum power plant based on these intervals. A description of the confusion matrix is presented in Section 2.4.3. The classification error plots reveal that class 2, representing values between 0MW and 25% of maximum power, is the most frequently occurring class. The confusion matrices demonstrate that all models accurately classify approximately 90% of data points in this class. Similarly, all three models perform well on the 5th class, representing values between 75% and 100% of maximum power, with accuracies between 0.85 and 0.88. The accuracy refers to the fraction of correct predictions as explained in Section 2.4.3. The models also exhibit similar performance for class 3 and 4, however, it is interesting to note that the LSTM neural network is outperformed by both models on class 4. Lastly, the models' performance on the first class (no production) differs significantly. The random forest regressor achieves the highest score for class 1 at 0.78, which is reflected in the f1-score presented in Table 4.3. The MLP neural network correctly classifies 32% of the data points, while 63% of the data points are predicted with values between 0MW and 25% of maximum power. The LSTM neural network predicts 59% of values in class 1 correctly, while 39% of the data points are predicted within class 2.



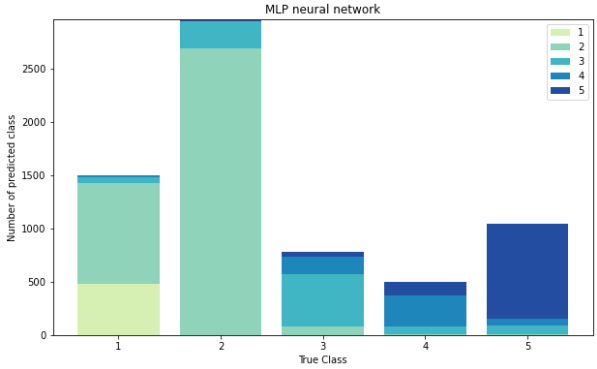
(a) Confusion matrix for Random forest



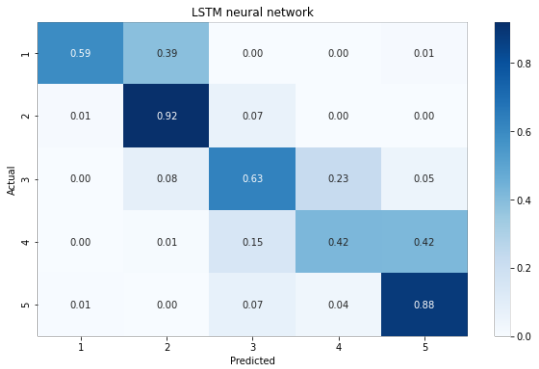
(b) Classification error for Random Forest



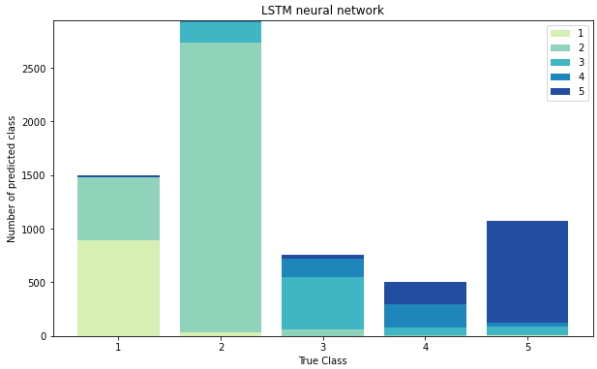
(c) Confusion matrix for MLP neural network



(d) Classification error for MLP neural network



(e) Confusion matrix LSTM



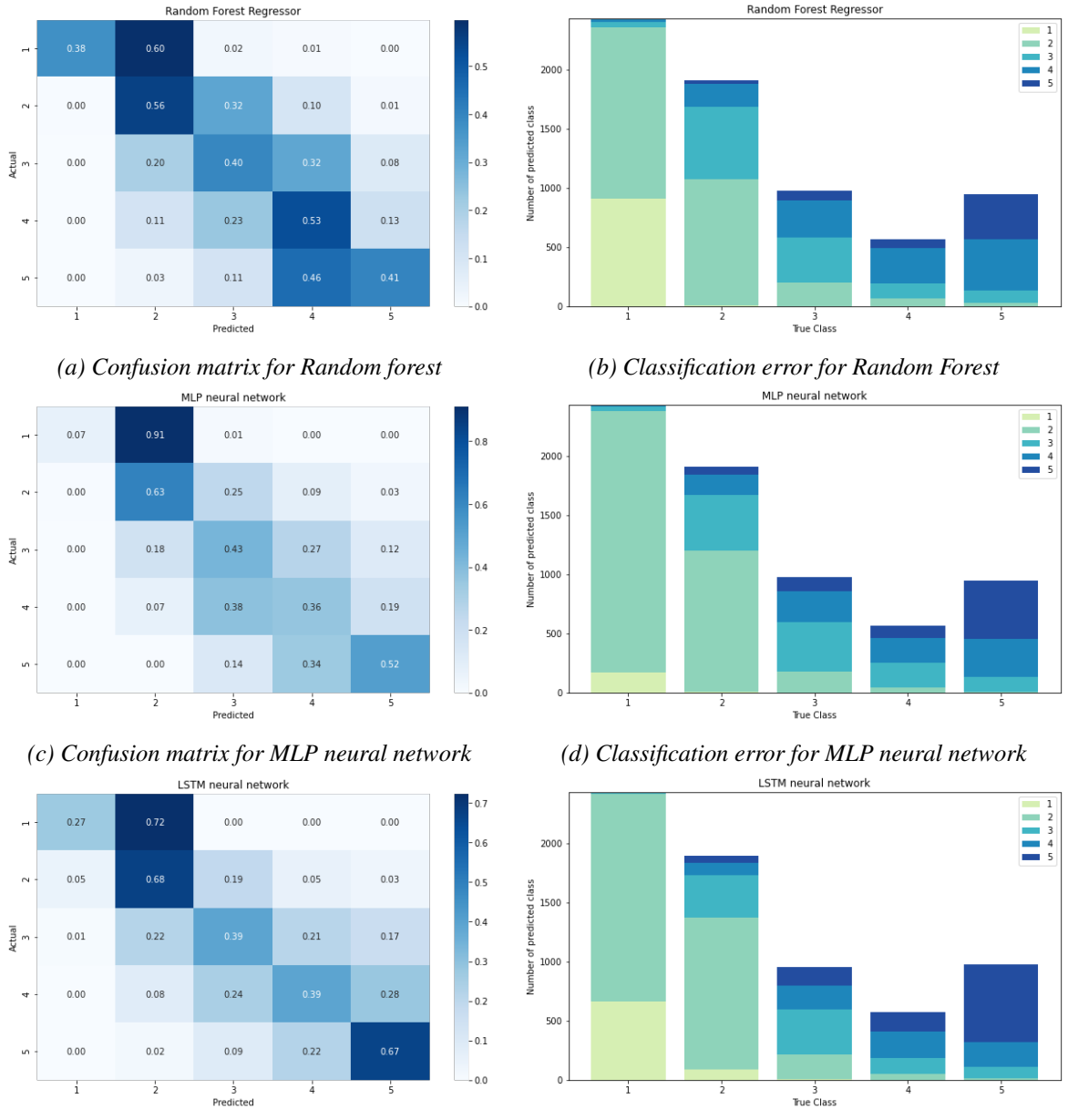
(f) Classification error for LSTM

Figure 4.9: Distribution of predicted values for Bjørgum power plant.

To summarise, the LSTM neural network outperforms the other models on class 2,3 and 5. When examining the confusion matrix and classification error plot for the LSTM neural network, it becomes evident that the misclassified data points for class 1 are predicted with values within class 2, and the misclassified data points for class 4 are predicted with values within class 3 and 5. Therefore, the errors are not large, specifically they never exceed 25% of the maximum power. Based on this, together with the results for MAE and MSE, the LSTM neural network was selected as the final model for Bjørgum.

In Figure 4.10, the confusion matrices and classification error plots for the random forest regressor, MLP neural network, and LSTM neural network are displayed for Furegardane power plant. As for Bjørgum, the classification error plots reveal that the

dataset is imbalanced with most of the data points in class 1 and 2. Despite the over-representation of the first class, the classification error for the three models are high. All three models predict the majority of the data points in class 1 with values within class 2. An explanation to this could be undiscovered technical downtime, creating noise in the dataset. The second class is also the second most represented class, and the three models exhibit similar performance with accuracies between 0.56 and 0.68. Interestingly, the LSTM neural network is outperformed by the other models on class 3 and 4, though the difference is not large. Looking at the distribution of the predictions within class 3 and 4 for all three models, it is clear that these two classes are the most difficult to predict. Finally, for class 5, the LSTM neural network attains the highest accuracy of 0.67, while the random forest regressor and MLP neural network have accuracies of 0.41 and 0.52, respectively.



(a) Confusion matrix for Random forest

(b) Classification error for Random Forest

(c) Confusion matrix for MLP neural network

(d) Classification error for MLP neural network

(e) Confusion matrix LSTM

(f) Classification error for LSTM

Figure 4.10: Distribution of predicted values for Furegardane power plant

In summary, although the LSTM neural network had the best MAE, MSE, and f1-score, as shown in Table 4.3, a closer analysis of the distribution of the predictions reveal that it does not outperform the other models on all classes. Nevertheless, the LSTM still had notably better scores in the most represented classes and acceptable scores for classes 3 and 4. Consequently, the LSTM neural network was chosen as the final model for Furegardane as well.

4.4 Model Evaluation

The selected machine learning model was evaluated against both Småkraft's current method for forecasting hourly production the next day and a simple baseline model that predicts next day's production using only the previous day's production. In doing so, the MAE, MSE and f1-score were calculated for each method. The relative improvement for the existing method and the LSTM neural network compared to the baseline model is denoted in parentheses for each metric. The results are presented in Table 4.4 and Table 4.5 for Bjørgum and Furegardane, respectively.

Table 4.4: MAE [MWh], MSE [MWh^2] and f1-score [/] for the baseline model, Småkrafts current approach and the LSTM neural network for Bjørgum. The improvement of each metric compared to the baseline model in percentages is denoted in parenthesis.

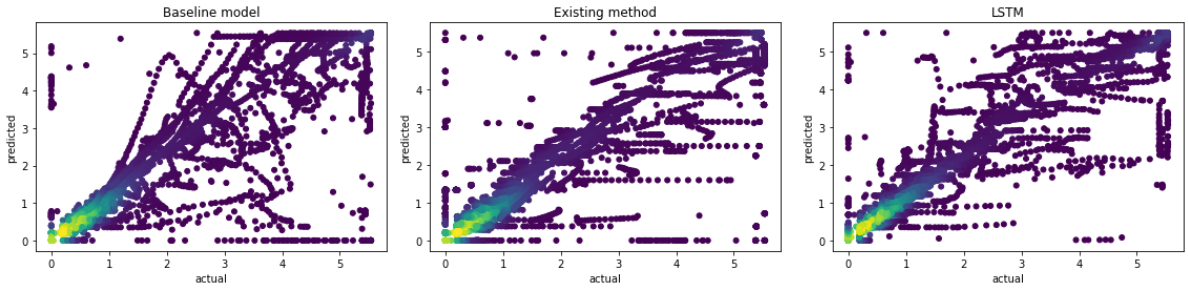
	Baseline model	Existing method	LSTM neural network
MAE	0.380	0.394 (-4%)	**0.316 (16%)**
MSE	0.771	0.741 (4%)	**0.453 (41%)**
f1-score	0.706	**0.723 (2%)**	0.695 (-2%)

Table 4.5: MAE [MWh], MSE [MWh^2] and f1-score [/] for the baseline model, Småkrafts current approach and the LSTM neural network for Furegardane. The improvement of each metric compared to the baseline model in percentages is denoted in parenthesis.

	Baseline model	Existing method	LSTM neural network
MAE	0.790	0.760 (4%)	**0.573 (27%)**
MSE	2.056	1.690 (18%)	**0.942 (54%)**
f1-score	**0.518**	0.501 (-3%)	0.458 (-12%)

The LSTM neural network yielded the best results in terms of both MAE and MSE for both locations. From an economic point of view, the MAE for the existing method at Bjørgum equals 39,968 NOK in imbalance fees per year (11.58 NOK/MWh), whereas the MAE for the LSTM neural network results in imbalance fees of 32,055 NOK per year. Similarly, at Furegardane, the MAE for the existing method equals 77,095 NOK in imbalance fees per year, while the MAE for the LSTM neural network results in imbalance fees of 58,126 NOK per year. Using these figures as a reference for the 176 power plants Småkraft owns in Norway, the LSTM neural network has a potential of reducing the imbalance fees by an amount between 1.4MNOK and 3.3MNOK per year.

Comparing the f1-scores across the three models, the LSTM neural network resulted in the poorest score at both locations. Småkraft's current method had the highest f1-score for Bjørgum power plant, while the baseline model yielded the best f1-score for Furegardane power plant. One possible explanation for this result is that the LSTM neural network accurately classifies the over-represented classes, but struggles with the under-represented classes. Another explanation is that the baseline model and the existing method have some large prediction errors, which is also indicated by the MSE. To investigate this further, scatter plots of predicted versus actual production are presented in Figure 4.11 and 4.12, and confusion matrices and classification error plots are presented in Figure 4.13 and Figure 4.14.

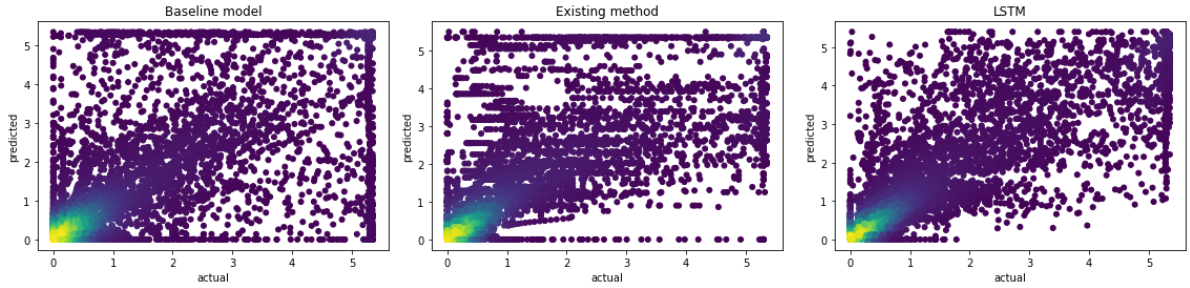


(a) Baseline model *(b) Existing method* *(c) LSTM neural network*

Figure 4.11: Scatterplot of predicted (y) versus actual (x) values of production.

In Figure 4.11 the predicted (y) and actual (x) production have been plotted together for each hour using the test dataset at Bjørgum for the baseline model (a), the existing method (b) and the LSTM neural network (c). Data points are colored by density, with the most frequent values appearing yellow/green and the least frequent values purple. The figure indicates that the baseline and existing models more often predict no production when production does occur, compared to the LSTM neural network. Additionally, the plots demonstrates that the existing method often underestimates production, while the LSTM neural network exhibits a more evenly spread distribution of values. In Figure 4.12 the same plots have been created using the test dataset at Furegardane. Compared to the plots at Bjørgum, the data points in all three plots in Figure 4.12 are more spread out, demonstrating larger prediction errors. The density of data points at the edges of the plot for the baseline model (a) and the existing method (b) indicate that the models struggle to correctly predict full production and no production.

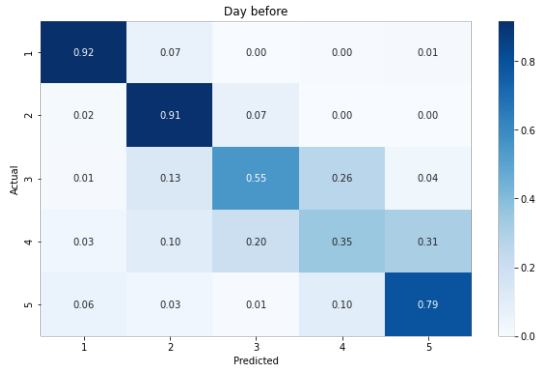
This problem seems to be improved by the LSTM neural network. The plot for the LSTM neural network (c) shows a more clear trend with fewer data points at the edges compared to (a) and (b).



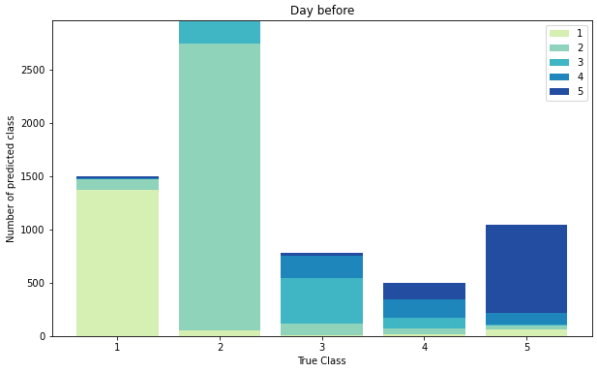
(a) Baseline model *(b) Existing method* *(c) LSTM neural network*

Figure 4.12: Scatterplot of predicted (y) versus actual (x) values of production.

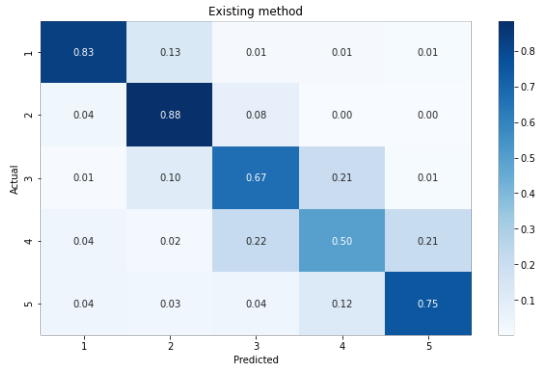
Figure 4.13 display the confusion matrices and classification error pots for the baseline model, existing method and LSTM neural network at Bjørgum. As seen in Figure 4.9 the LSTM neural network performs well on class 2 and 5, achieving accuracies of 0.92 and 0.88, respectively. The same values for the baseline model and the existing method reveal that the performance on class 2 is comparable with the LSTM model, while the performance on class 5 is weaker. When comparing class 3 and 4 across all methods, it can be seen that the performance of the methods is similar. Looking at the performances on class 1 on the other hand, the LSTM neural network exhibits poor performance compared to the other methods. While the existing method and the baseline model accurately predict data points within class 1 with accuracies of 0.83 and 0.92, respectively, the LSTM neural network only correctly predicts 59% of the data points with no production. Nonetheless, it is noteworthy that the confusion matrix values for the LSTM neural network are more concentrated around the diagonal compared to the other methods, which have values on the left side of the diagonal. This indicates that these methods, in some cases, predict data points with values that are too low.



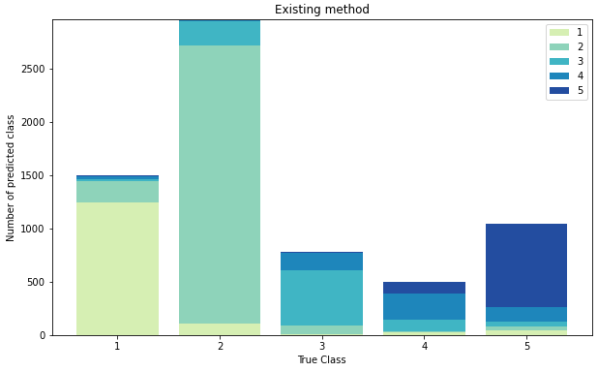
(a) Confusion matrix for the baseline model



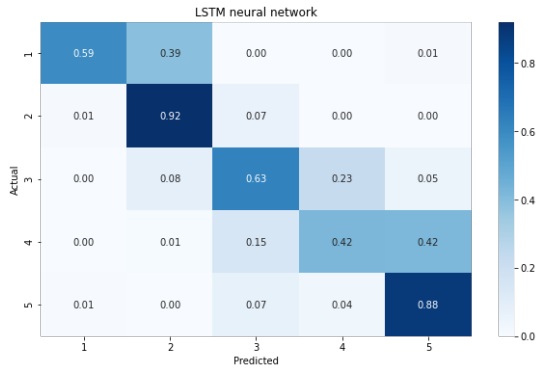
(b) Classification error for the baseline model



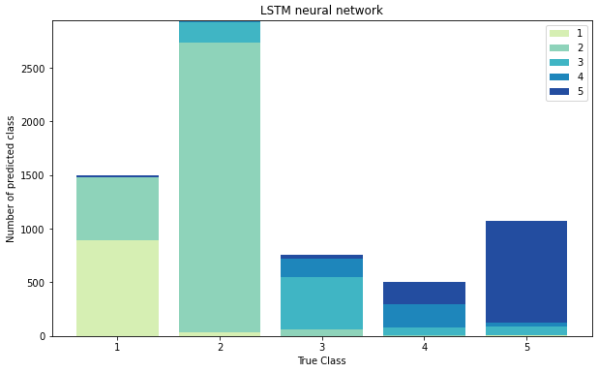
(c) Confusion matrix for the existing method



(d) Classification error for the existing method



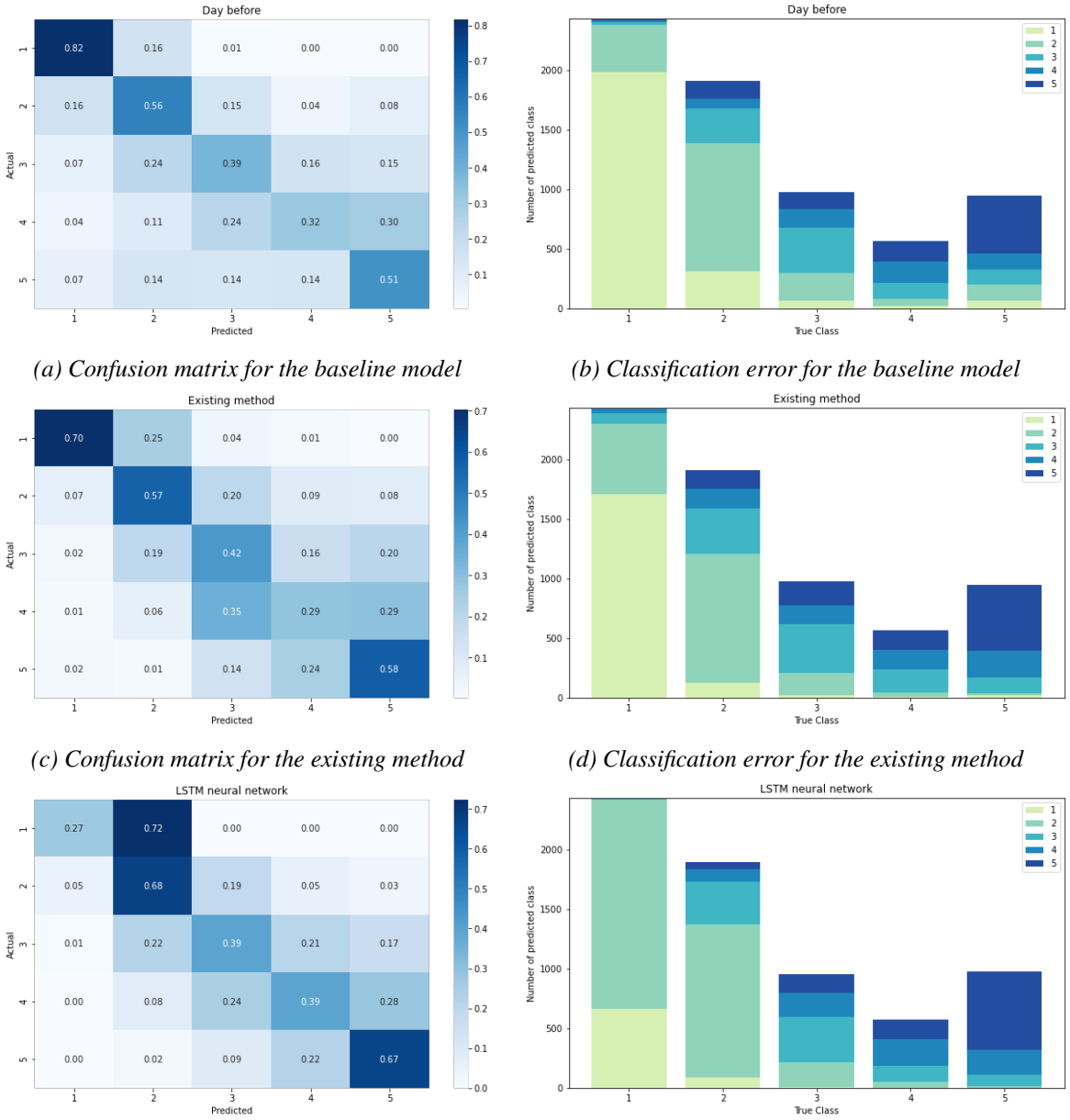
(e) Confusion matrix for the LSTM neural network



(f) Classification error for the LSTM neural network

Figure 4.13: Distribution of predicted values using the baseline model, Småkraft's existing method and the selected machine learning model (LSTM neural network) for Bjørgum power plant.

Figure 4.14 shows that the results for Furegardane are similar to that of Bjørgum's. The LSTM neural network outperforms the other methods on class 2, 4 and 5, and all three models perform similar on class 3. Same as for Bjørgum, the LSTM neural network performs poorly on class 1, where the model predicts 72% of the data points with values within class 2. The baseline model and the existing method on the other hand are able to predict data points within class 1 with accuracies of respectively 0.82 and 0.70. Lastly, it can be seen that the misclassified values are closer to the actual values for the LSTM neural network than for the two other methods, as indicated by the MAE and MSE. This is seen by the spread of values in the confusion matrix and the share of categories within the bars in the classification error plots.



(a) Confusion matrix for the baseline model

(b) Classification error for the baseline model

(c) Confusion matrix for the existing method

(d) Classification error for the existing method

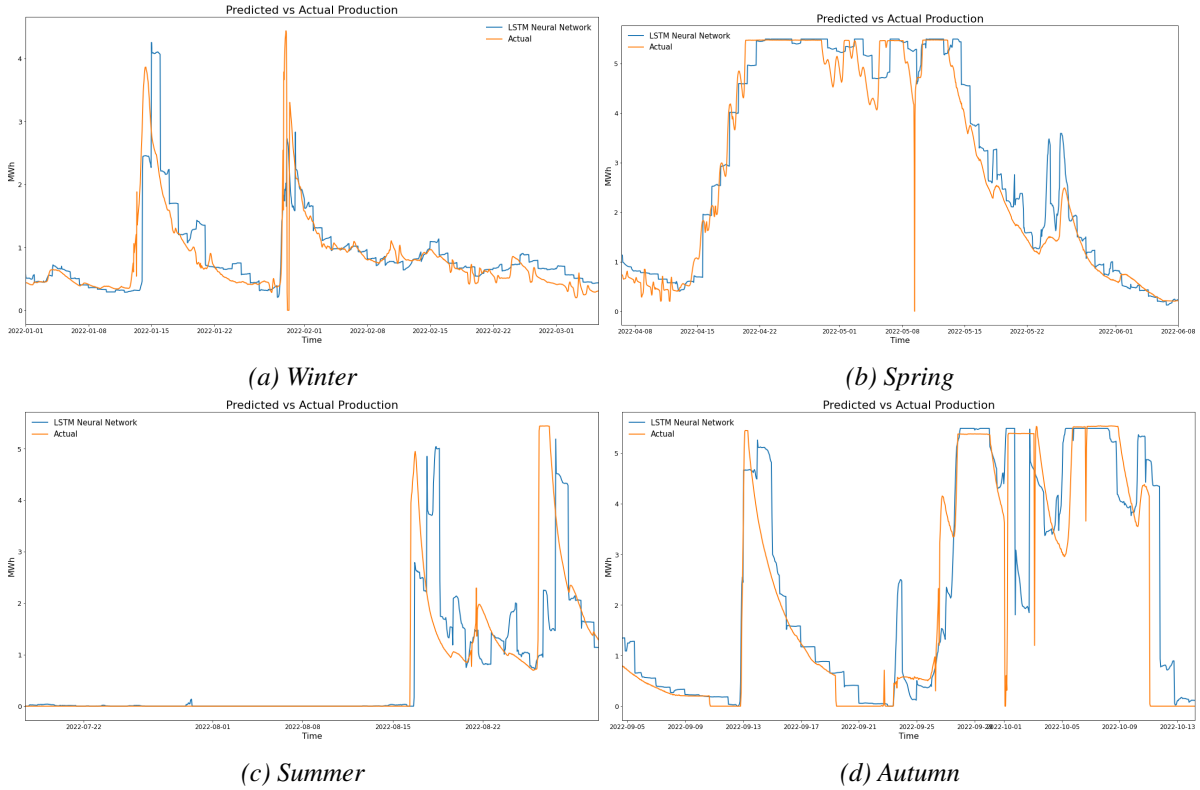
(e) Confusion matrix for the LSTM neural network

(f) Classification error for the LSTM neural network

Figure 4.14: Distribution of predicted values using the baseline model, Småkraft's existing method and the selected machine learning model (LSTM neural network) for Furegardane power plant.

4.5 Analysis of performance during various weather conditions

The third and final research question aimes to determine the weather conditions under which the model provides accurate predictions and those where it struggles. To investigate this, Figures 4.15 and 4.16 display plots of predicted production alongside actual production for periods during winter, spring, summer, and autumn at Bjørgum and Furegardane. These figures will be used to evaluate the performance of the machine learning model across various weather conditions.



(a) Winter *(b) Spring*

(c) Summer *(d) Autumn*

Figure 4.15: Predicted production plotted together with actual production for different periods during the year for Bjørgum.

Figure 4.15 (a) shows the predicted and actual production at Bjørgum during a period in winter 2022. The plot reveals that the model performs well for production values ranging between 0MW and 1.5MW, where production remains stable. This finding was also supported by the confusion matrix and classification error plot (class 2). The plot displays to peaks, one on the 14th of January and another on the 30th of January. The model seems to react at the right time during these two events, but underestimates the magnitude. For the first peak the model appears to correct itself once it receives new input on production levels through the *average production* feature. As a reminder, the *average production* feature represents the average production during the six hours leading up to the reporting time and is the dominant feature as seen in Figure 4.7. During the first peak, the observational based temperature increased from below 0°C to 4°C, and 6mm of precipitation was forecasted over a period of 8 hours which was supported by the observational based values. The forecasted temperature, however, appears to differ from the observational based values as they stay below 0°C. This could explain why the model underestimates the increase. For the second peak, the temperature during the hours leading up to the production peak was 2-3°C, but dropped below 0°C during the peak according to the forecast. At the same time, the forecast predicted 24mm of precipitation over a 24-hour period, which was consistent with the observational based values. The forecast indicated a combination of rain and snow. This could explain why the model struggles to correctly predict the magnitude of the power output.

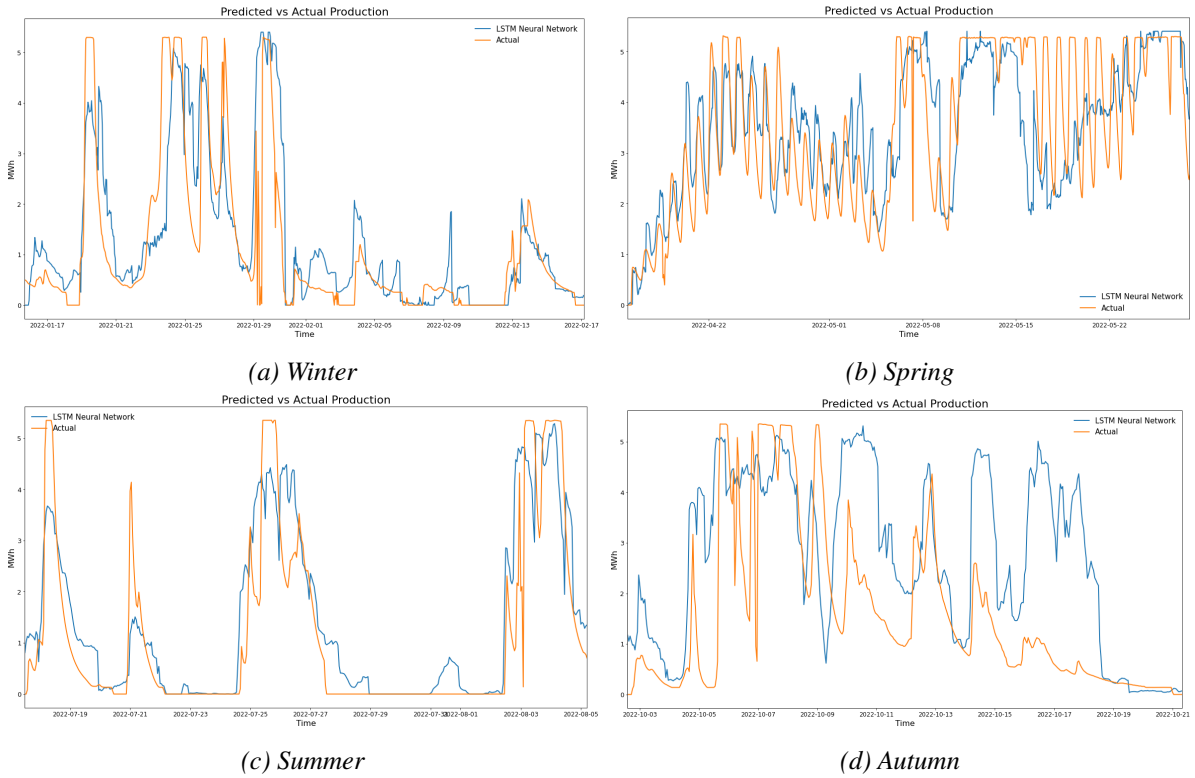
In Figure 4.15 (b), the predicted and actual production at Bjørgum during a period in spring 2022 are displayed. The plot reveals that production gradually increases from

minimal levels to maximum levels over the course of a week, from the 14th of April to the 20th of April, and remains constant for the following week. During this period, the temperature rises from below 0°C to a maximum of 6°C, while estimated snow melt ranges from 6mm to 20mm. In the following period, from the 28th of April to the 13th of May, the temperature appears to drop below 0°C at night and rise above 0°C during the day, which can explain the fluctuations in production. As production begins to decrease, on the 13th of May, the temperature remains above 0°C, but the estimated snowmelt is equal to zero indicating that the snow storage is empty. During this snow melting event, the machine learning model accurately predicts the production levels. However, after production dropped to around 1MW, the plot reveals two predicted peaks in production, around the 26th of May, that do not match the actual production levels. Examination of the input values reveals a discrepancy between observational-based and forecasted precipitation during this period, indicating that the forecasted rainfall events did not occur.

In Figure 4.15 (c), the predicted and actual production at Bjørgum are displayed for a period during summer 2022. The plot reveals that the power plant did not operate from the beginning of June to the middle of August, which is a recurring trend observed in the dataset dating back to 2017. This period is mostly characterised by drought, but there are some rainfall events during which the power plant does not start. For instance, on June 27th, almost 20mm of precipitation was forecasted during a 24-hour period, resulting in a slight increase in predicted production. It seems like the model recognises that the catchment area is not saturated as it does not predict a larger power output. When the power plant resumes operations in mid-August, the input values to the model indicate that around 25mm of precipitation was forecasted, which aligns with the observational based data. The model predicts the increase in production six hours later than the actual increase, and initially underestimates the increase before predicting a higher power output for the next day when the actual production starts to decline. In this situation the model may have corrected itself based on the *average production* feature, as was observed in Figure 4.15 (a) as well. The same situation appears at the end of August.

Figure 4.15 (d) displays the predicted and actual production at Bjørgum for a period during autumn 2022. On the 13th of September, the production increased from no production to full production over 12 hours. When comparing the predicted and actual production during this period, it can be seen that the predictions increased two hours earlier than the actual production, but then followed the actual production almost perfectly until the peak. As observed in the other plots, the model underestimates the magnitude of the peak, this time with 0.8MW, and then proceeds to increase the next day when the actual production decreases. Investigating the weather conditions during this period reveals a rainfall event of 40mm lasting around 30 hours. Following this peak, the production decreases to zero over the course of a week. When the power plant starts operating on the 23rd of September, the predictions exceed the actual power output. Investigating the input variables reveal that 20mm of precipitation was forecasted over a 24-hour period, while the observational based values indicate a smaller rainfall event of 10mm. At the same time, the rapid decrease in production before this event and the irregular fluctations in the following period indicate that there could have been

some technical issues during this time.



(a) Winter *(b) Spring*

(c) Summer *(d) Autumn*

Figure 4.16: Predicted production plotted together with actual production for different periods during the year for Furegardane.

Figure 4.16 (a) display the predicted and actual production at Furegardane for a period during winter 2022. Comparing this plot to the same one for Bjørgum indicate that there is much more variation in production for Furegardane. This applies to all four seasons shown in the figure. For the period during winter, the machine learning model seem to react mostly at the right time in terms of increase and decrease in production, however, the it both under- and overestimates the magnitude. Specifically, the model underestimates when the actual power output reaches maximum, whereas overestimation occurs when the power output falls below 2MW. The weather conditions during this period is characterised by multiple heavy rainfall events, and temperatures fluctuating around 0°C. In some instances, the temperature may explain why the model struggles to accurately predict the magnitude of the power output due to uncertainties related to precipitation type and freezing/melting.

Figure 4.16 (b) displays the predicted and actual production at Furegardane for a period during spring 2022. In the beginning of the plot there is an increase in production from minimal production to maximum production in the period from the 15th of April to the 24th of April, followed by a slight decrease until the 4th of May. Throughout this period there are daily fluctuations in production. Investigations of the weather situation at this time reveal that the temperature rise above 0°C during the day, and sink below 0°C during night time. At the same time the values for snow melting follow

the same overall trend in terms of increase and decrease in production. For the next part, there are some longer periods with full production, until the 15th of May where fluctuations can be observed again. The model predicts the period with full production well, but struggles to accurately predict the following fluctuations. During this period, specifically from the 15th of May until the 23th of May, the temperature seem to be above 0°C the whole time, but according to the values of estimated snow melting and snow depth the snow storage is empty. As these values are estimated through the winter without any observational corrections, any estimation error will be retained in the dataset. For this reason, it is not unlikely that the estimated snow storage is wrong at the end of the season. This would explain why the predicted production is restrictive during the last part of the plot.

In Figure 4.16 (c), the production at Furegardane during a period in summer 2022 is shown and compared to the predicted values. Looking at the first three peaks in the plot, occurring on the 18th, 20th and 25th of July, the timing of the peaks are accurately predicted, but the model fails to predict the magnitude of the peaks. The first peak was caused by a heavy rainfall event lasting around 10 hours, whereas the third peak was due to a three-day long rainfall event. For the second peak, there appears to be a discrepancy between the forecasted and observed precipitation. Similarly, just before the last peak, starting on the 2nd of July, there was a short increase in predicted production without any actual production, which can be attributed to another discrepancy between forecasted and observational based precipitation. On this day, the forecast predicted 6mm of precipitation, but only 0.6mm was observed. The last peak, lasting for five days, was caused by heavy rain over several days, and the model fits well when comparing predicted and actual production for this event.

Figure 4.16 (d) displays the predicted and actual production at Furegardane for a period during autumn 2022. Out of the four plots in Figure 4.16, this period seems to be the one with the highest prediction error. Although the predictions generally follow the same trend as the actual production, the model appears to overestimate production most of the time. On the 5th of October there is a significant deviation where the predicted power output is around 4MW, but the actual production is close to 0. This is the start of a five-day heavy rainfall event with observational-based values ranging between 23mm and 35mm each day. The sudden rise and fall in production during this time could indicate technical issues such as problems with the intake due to debris brought down by the river. During the period from the 9th of October to the 18th of Octber there are significant discrepancies between the predicted and actual production. Investigations of the weather data reveal large differences between forecasted and observational-based values for precipitation. This could explain why the model expects high power output, but the production remains low.

Chapter 5

Conclusions and Future Work

This thesis aimed to predict hourly production in run of river hydropower plants using publicly available weather data, and increase the accuracy compared to the solution Småkraft uses today. This was done by creating and optimising three different machine learning models, and selecting the best one for comparison. The three models, a random forest regressor, a multilayer precepton neural network and a long short term memory neural network were evaluated both on their overall performance and on 25%-intervals of maximum power output. The results revealed that the models performed differently on the intervals, but the LSTM neural network had the overall best performance and was selected for both locations.

The final model was then compared to the existing method, using a baseline model where the forecasted production was equal to the previous day's production, hour by hour. The results for Bjørgum showed that the LSTM had a 16% improvement in MAE and a 41% improvement in MSE compared to the baseline model, while the existing method had a 4% increase in MAE and a 4% decrease in MSE compared to baseline model. The existing method had a 2% improvement on the f1-score, while the LSTM has a 2% decrease in the f1-score. Similariy for Furegardane, the results showed that the LSTM had an improvement in MAE and MSE of 27% and 54%, respectively, compared to the baseline model. The existing method at Furegardane had an improvement in MAE of 4% and an 18% improvement in MSE compared to the basline model. At Furegardane the baseline model had the best f1-score, while the existing method had a 3% decrease and the LSTM neural network had a 12% decrease. This indicates that the LSTM neural network increases performance on the most represented classes, but not on all classes. However, the improvements in MAE and MSE indicate that even though the production is not always predicted within the correct class, the prediction errors are not large. If the improvement seen at the selected power plants are representative for all power plants, the findings suggest that utilising the LSTM neural network for the 176 power plants owned by Småkraft in Norway could potentially result in a decrease in imbalance fees by an amount between 1.4MNOK to 3.3MNOK.

In relation to predicting hydropower production, the thesis investigates three research questions. Firstly, the significance of different time intervals for weather variables related to production was examined. When comparing the feature selection results for Bjørgum and Furegardane, it became clear that the nature of the catchment area play an important role in the controlling the runoff. This was supported by the correlation analysis, even though the optimal values from the correlation plots did not match exactly with the results from the feature selection. For Bjørgum the feature selection process revealed that 24 hours and 72 hours of accumulated rainfall were the most important, while for Furegardane 24 hours and 18 hours were most significant.

Secondly, the most important features for predicting production were identified for

both locations. The analysis found that similar features were important at both location. The most important feature at both locations was the *average production* feature, representing the average production during the six hours leading up to the reporting time. This feature dominated both datasets, however, it was more dominant at Bjørgum than Furegardane. This can be explained by the production patterns for the power plants. As Bjørum has less variation in production from day to day, the average production from the day before is often representative for the next day. This is often not the case for Furegardane which has more volatile production patterns. Furthermore, various intervals of accumulated rainfall were important, together with the snowmelt-feature ($qsw(mm)$). At both locations the accumulated rainfall during one month was selected as one of the most important features, which provides the models with information on the degree of saturation for the ground.

Finally, the performance of the final model was evaluated across different seasons and weather situations, demonstrating its accuracy in predicting production under various conditions. The plots for Bjørgum and Furegardane was quite different due to the different production patterns, however, similar problems arose at both locations. Firstly, instances with discrepancies between forecasted precipitation and observational based precipitation were found at both locations. In these cases, the input to the models are wrong and consequently the predictions will be inaccurate. Secondly, both models tend to underestimate the peaks. Specifically for Bjørgum, the model tends to underestimate the peaks and then, due to the dominant *average production* feature, correct itself with the new input. Generally, the analysis showed that both models have good timing. Bjørgum performed especially well on the lower values where the production was stable. Furegardane performed good during the snow melting event, however, the accuracy decreased at the end of the event when the estimated snow storage was empty.

In conclusion, the use of machine learning to predict hourly production in a run of river hydropower plant has proven to be successful. However, the accuracy of the model is highly dependent on the quality of the input data. Furthermore, the research questions provided valuable insights into the relationship between weather variables and hydropower production, with certain time intervals and specific features being identified as most significant for accurate predictions.

One major constraint of the model is its dependence on a sufficient amount of high-quality historical data, rendering it impossible to use on newly built power plants. Future research could explore alternative approaches for applying the model to power plants with limited historical data. One potential solution could involve identifying the k nearest neighbors (power plants) in terms of location, as well as comparable catchment area characteristics and climate, and then creating an ensemble model based on the models of each identified neighbor. Another limitation is the model's inability to accurately predict every interval of power output. Future research could aim to enhance the accuracy of the under-represented intervals while maintaining the model's performance on the over-represented intervals.

Chapter 6

Appendix

6.1 Subsets used in each model

Table 6.1: Subsets used in the random forest regressor, created using recursive feature elimination, feature importance and domain knowledge at Bjørgum (in random order).

Feature importance	Recursive feature elimination	Domain knowledge
average_prod	average_prod	average_prod
snowmelt_11day	snowmelt_13day	month_sin
snowmelt_7day	snowmelt_11day	rainfall_amount_acc
snowmelt_5day	snowmelt_7day	qsw(mm)
snowmelt_3day	snowmelt_5day	rr_acc_month
snowmelt_2day	snowmelt_3day	air_temperature_2m
snow_obs	snowmelt_2day	rainfall_acc_24
rr_acc_month	rr_acc_month	rainfall_acc_48
rain_obs	rain_obs	rainfall_acc_72
qsw(mm)	qsw(mm)	
tm(Celcius)	tm(Celcius)	
sd(cm)	sd(cm)	
rr(mm)	rr(mm)	
rainfall_acc_120	rainfall_acc_120	
rainfall_acc_108	rainfall_acc_108	
rainfall_acc_96	rainfall_acc_96	
rainfall_acc_84	rainfall_acc_84	
rainfall_acc_72	rainfall_acc_72	
rainfall_acc_60	rainfall_acc_60	
rainfall_acc_48	rainfall_acc_48	
rainfall_acc_24	rainfall_acc_24	
rainfall_acc_18	rainfall_acc_18	
rainfall_acc_12	rainfall_acc_12	
rainfall_acc_6	rainfall_acc_6	
rainfall_acc_week	rainfall_acc_week	
rainfall_amount_acc	rainfall_amount_acc	
day_of_year_cos	day_of_year_cos	
day_of_year_sin	day_of_year_sin	
month_cos	month_cos	
month_sin	month_sin	
y_wind_10m	y_wind_10m	
x_wind_10m	x_wind_10m	
relative_humidity_2m	relative_humidity_2m	
air_temperature_2m	air_temperature_2m	
snowfall_amount_acc	snowfall_amount_acc	
precipitation_amount_acc	precipitation_amount_acc	

Table 6.2: Subsets used in the MLP neural network, created using recursive feature elimination, feature importance and domain knowledge at Bjørgum (in random order).

Feature importance	Recursive feature elimination	Domain knowledge
average_prod	average_prod	average_prod
rainfall_acc_week	rainfall_acc_week	month_sin
rainfall_amount_acc	rainfall_amount_acc	rainfall_amount_acc
qsw(mm)	qsw(mm)	qsw(mm)
rr_acc_month	rr_acc_month	rr_acc_month
rainfall_acc_72	rainfall_acc_72	air_temperature_2m
rainfall_acc_60	rainfall_acc_48	rainfall_acc_24
rainfall_acc_48	rainfall_acc_24	rainfall_acc_48
rainfall_acc_24	tm(Celcius)	rainfall_acc_72

Table 6.3: Subsets used in the LSTM neural network, created using recursive feature elimination, feature importance and domain knowledge at Bjørgum (in random order).

Feature importance	Recursive feature elimination	Domain knowledge
average_prod	average_prod	average_prod
month_cos	snowmelt_3day	month_sin
rainfall_amount_acc	rainfall_amount_acc	rainfall_amount_acc
qsw(mm)	qsw(mm)	qsw(mm)
rr_acc_month	rr_acc_month	rr_acc_month
air_temperature_2m	air_temperature_2m	air_temperature_2m
rainfall_acc_72	rainfall_acc_72	rainfall_acc_24
rainfall_acc_60	rainfall_acc_60	rainfall_acc_48
rainfall_acc_48	rainfall_acc_48	rainfall_acc_72
rainfall_acc_24	rainfall_acc_24	
rainfall_acc_week	rainfall_acc_week	
sd(cm)	sd(cm)	
y_wind_10m	y_wind_10m	
precipitation_amount_acc	precipitation_amount_acc	
day_of_year_cos	day_of_year_cos	
day_of_year_sin	tm(Celcius)	

Table 6.4: Subsets used in the random forest regressor and LSTM neural network, created using recursive feature elimination, feature importance and domain knowledge at Furegardane (in random order).

Feature importance	Recursive feature elimination	Domain knowledge
average_prod	average_prod	average_prod
rr_acc_month	rr_acc_month	month_sin
rr(mm)	rr(mm)	rainfall_amount_acc
rainfall_amount_acc	rainfall_amount_acc	qsw(mm)
rainfall_acc_96	rainfall_acc_96	rr_acc_month
rainfall_acc_24	rainfall_acc_24	air_temperature_2m
rainfall_acc_22	rainfall_acc_22	rainfall_acc_3
rainfall_acc_18	rainfall_acc_18	rainfall_acc_6
rainfall_acc_12	rainfall_acc_6	rainfall_acc_18
rainfall_acc_week	rainfall_acc_week	
rainfall_amount_acc	rainfall_amount_acc	
month_sin	month_sin	
x_wind_10m	x_wind_10m	
relative_humidity	relative_humidity_2m	
air_temperature_2m	air_temperature_2m	
snowfall_amount_acc	snowfall_amount_acc	
qsw(mm)	qsw(mm)	
tm(Celcius)	tm(Celcius)	

Table 6.5: Subsets used in the random forest regressor, created using recursive feature elimination, feature importance and domain knowledge for MLP neural network at Furegardane (in random order).

Feature importance	Recursive feature elimination	Domain knowledge
average_prod	average_prod	average_prod
rr_acc_month	rr_acc_month	month_sin
rainfall_amount_acc	rainfall_amount_acc	rainfall_amount_acc
rainfall_acc_24	rainfall_acc_96	qsw(mm)
rainfall_acc_22	rainfall_acc_24	rr_acc_month
rainfall_acc_18	rainfall_acc_18	air_temperature_2m
rainfall_acc_12	rainfall_acc_week	rainfall_acc_3
qsw(mm)	snowmelt_14day	rainfall_acc_6
		rainfall_acc_18

Bibliography

Anderson, D., H. Moggridge, P. Warren, and J. Shucksmith (2015), The impacts of run-of-riverhydropower on the physical and ecological condition of rivers, *Water and Environment Journal*, *29*(2), 268–276. 1.1.1

Bøckman, T., S.-E. Fleten, E. Juliussen, H. J. Langhammer, and I. Revdal (2008), Investment timing and optimal capacity choice for small hydropower projects, *European Journal of Operational Research*, *190*(1), 255–267. (document), 2.1

Bojek, P. (2022), Hydroelectricity, technology deep dive: More efforts needed, accessed on 29/11/22. 1.1.1

Breeze, P. (2018), *Hydropower*, Academic Press. 1.1.1

Breeze, P. (2019), *Power generation technologies*, Newnes. 1.1.1

Davie, T. (2019), *Fundamentals of hydrology*, Routledge. 2.1

Deng, X., Q. Liu, Y. Deng, and S. Mahadevan (2016), An improved method to construct basic probability assignment based on the confusion matrix for classification problem, *Information Sciences*, *340*, 250–261. (document), 2.4.3, 2.6

eSett (2023), Nordic imbalance settlement handbook: Instructions and rules for market participants, accessed on 18/05/23. 1.1.3

Frogner, I.-L., A. T. Singleton, M. Ø. Køltzow, and U. Andrae (2019), Convection-permitting ensembles: Challenges related to their design and use, *Quarterly Journal of the Royal Meteorological Society*, *145*, 90–106. 3.1.3

Gers, F. A., J. Schmidhuber, and F. Cummins (2000), Learning to forget: Continual prediction with lstm, *Neural computation*, *12*(10), 2451–2471. 2.4.2

Hartmann, D. L. (2015), *Global physical climatology*, vol. 103, Newnes. 3.1.3

Hastie, T., R. Tibshirani, J. H. Friedman, and J. H. Friedman (2009), *The elements of statistical learning: data mining, inference, and prediction*, vol. 2, Springer. (document), 2.3.2, 2.4.3, 2.5

Hochreiter, S., and J. Schmidhuber (1997), Long short-term memory, *Neural computation*, *9*(8), 1735–1780. 2.4.2

Hyndman, R. J. (2011), Moving averages. 2.3.1

iha, I. H. A. (2022), A brief history of hydropower, accessed on 29/11/22. 1.1.1

Impram, S., S. V. Nese, and B. Oral (2020), Challenges of renewable energy penetration on power system flexibility: A survey, *Energy Strategy Reviews*, *31*, 100,539. 1.1

James, G., D. Witten, T. Hastie, and R. Tibshirani (2013), *An introduction to statistical learning*, vol. 112, Springer. (document), 2.3.1, 2.4.1, 2.4.2, 2.2, 2.4.2, 2.4.2, 2.3, 2.4.2, 2.4, 2.4.2

Keras (), Model training apis, accessed on 5/05/23. 2.4.2

Klimaservicesenter, N. (2023), Observasjoner og værstatistikk, accessed on 05/04/23. 3.1.1

Lussana, C., O. E. Tveito, A. Dobler, and K. Tunheim (2019), senorge_2018, daily precipitation, and temperature datasets over norway, *Earth System Science Data*, *11*(4), 1531–1551. 3.1.4

Mitchell, T. M., et al. (2007), *Machine learning*, vol. 1, McGraw-hill New York. 2.4.1

Norges Bank (2023), Valutakurser, accessed on 16/05/23. 1.1.3

Norwegian Ministry of Petroleum and Energy (2022), The power market, accessed on 18/05/23. 1.1.3

NVE (2015), Mini-, mikro og småkraftverk, accessed on 18/05/23. 1.1.2

NVE (2016), About nve, accessed on 18/05/23. 1.1.2

NVE (2022), Brukerveiledning nevina, accessed on 05/04/23. 3.1.1

Polák, M. (2021), A brief history of the kaplan turbine invention, *Energies*, *14*(19), 6211. 1.1.1

Rumelhart, D. E., G. E. Hinton, and R. J. Williams (1986), Learning representations by back-propagating errors, *nature*, *323*(6088), 533–536. 2.4.2

Saloranta, T. (2012), Simulating snow maps for norway: description and statistical evaluation of the senorge snow model, *The Cryosphere*, *6*(6), 1323–1337. 3.1.4

scikit learn (a), sklearn.ensemble.randomforestregressor, accessed on 15/04/23. 3.3.1, 3.3.3

scikit learn (b), sklearn.neural$_n$*etwork.mlpregressor*,*accessedon*15/04/23. 3.3.1

Småkraft (2023), Om smÅkraft as, accessed on 11/05/23. 1.1

United Nations, F. C. o. C. C. (2022), Un climate change conference (cop26) at the sec glasgow 2021, accessed on 11/05/23. 1.1

Varsom SeNorge (), Om varsom senorge, accessed on 11/04/23. 3.1.4