Master Thesis
TVVR-25/5021

Case Study on the Application of Machine Learning in Hydrology

Leak Detection and Precipitation

Ola Davidsson
Axel Sweger

Division of Water Resources Engineering
Department of Building and Environmental Technology
Lund University

Case Study on the Application of Machine Learning in Hydrology

Leak Detection and Precipitation

By:
Ola Davidsson
Axel Sweger

Master Thesis
Division of Water Resources Engineering
Department of Building & Environmental Technology
Lund University
Box 118
221 00 Lund, Sweden

Water Resources Engineering
TVVR-25/5021
ISSN 1101-9824

Lund 2026
www.tvrl.lth.se

Master Thesis
Division of Water Resources Engineering
Department of Building & Environmental Technology
Lund University

Swedish title:	Fallstudie om Tillämpningen av Maskininlärning inom Hydrologi
English title:	Case Study on the Application of Machine Learning in Hydrology
Author(s):	Ola Davidsson Axel Sweger
Supervisors:	Amir Naghibi Alireza Taheri Dehkordi
Examiner:	Kenneth M Persson
Company Supervisors	David Karlsson Love Sjelvgren
Language	English
Year:	2026
Keywords:	Machine Learning; ANN; Leakage Detection; Precipitation;

Acknowledgements

We would like to express our sincere gratitude to our institutional supervisors, Assistant Professor Amir Naghibi and Alireza Taheri Dehkordi, for their expertise, continuous support and valuable feedback throughout our research.

A special thanks to our company supervisors, Love Sjelvgren and David Karlsson, for their support and constructive discussions during this project.

We are also grateful towards Kentima for providing computers as well as access to their office space during this project.

Finally, we extend our appreciation towards Johan Vikstrand at Kentima, who provided additional support, encouragement and insightful inputs

Abstract

This thesis presents a case study on the application of artificial intelligence (AI) in hydrology using two approaches. This topic was chosen since AI is an important tool, and water is a vital resource. The first case is on leakage detection with artificial data, while the second is on precipitation prediction with real data. For leakage detection, a random forest (RF) model and an artificial neural network (ANN) model were developed in series. The RF detects if there has been a leak or not; the detected leaks are forwarded to the ANN. The ANN determines where in the water distribution network the leak is located. This design achieved an accuracy of 88.6% and a precision of 94.1%. For precipitation, six models were developed; one spatial model and five area-specific prediction models. The models developed for precipitation outperformed the current satellite products with R^2 values above 0.5 for five out of six models. The results from both parts of the case study show promise, and hydrology is definitely a field into which AI should be further integrated.

Table of contents

List of Abbreviations

AI Artificial Intelligence

ANN Artificial Neural Network

API Application Programming Interface

CE Cross-Entropy

CFSR Climate Forecast System Reanalysis

IOA Index of Agreement

GEE Google Earth Engine

MAE Mean Absolute Error

MBE Mean Bias Error

ML Machine Learning

NOAA National Oceanic and Atmospheric Administration

SMHI Swedish Meteorological and Hydrological Institute

WDN Water Distribution Network

ResNet Residual Artificial Neural Network

RF Random Forest

RMSE Root Mean Square Error

1. Introduction

Water is essential for all life on Earth, supporting both human populations and ecosystems. For these reasons, it is vital that the resource is properly managed. However, this is not the case, and leakages in distribution networks are a pressing concern for all countries. The total loss of non-revenue water volume worldwide is estimated to be 126 billion cubic meters per year (Liemberger & Wyatt, 2018). This corresponds to 30% of the total input value of the world's piped water for municipal supply. In their paper, they presented this loss in terms of economics, which translates to a loss of 39 billion USD per year. The water leakage problem is prevalent in all countries, including Sweden. Ninety percent of the Swedish population is connected to the municipal drinking water network (Vatten, 2023), and large losses in the network will have an effect on most citizens. The losses in the municipal drinking water network in Sweden are estimated to be 19% (Vatten, 2023), an increase from previous years when the losses were estimated to be 16-17%. This increase may appear to be very small, but in terms of a whole country, this translates into large losses in terms of volume of water. The water input into the Swedish water network in 2023 was 844 million cubic meters (Vatten, 2023). This means that a 1% increase in loss corresponds to 8.44 million cubic meters of drinking water. The fact that the losses trended upward in 2023 is very alarming, especially when the need for water is often increasing in correlation with the increasing population.

Today, most leakages are actively fixed in Europe, meaning that no action is taken before the leak or the customer's complaint (Sörensen et al., 2024). This results in unnecessarily large amounts of wasted water, as this could have been prevented earlier. With a proactive approach, faults can be identified earlier and even be combated before they turn into leaks. This approach would lead to a decrease in water loss. However, it is unclear how this proactive approach should work, as it would require further investment in detecting faults and repairs. It is also unclear what the best solution is to detect faults in the network.

Precipitation is fundamental for municipal and industrial water supply, plays a vital role in agriculture in all areas of the world, and is essential for agricultural output. Furthermore, it is also a risk indication for areas that are prone to flooding, as precipitation often 1 is the cause of flash floods. Even with these important factors, precipitation is poorly monitored. This thesis is focused on the southern region of Sweden, namely Skåne. In Skåne there are currently 43 stations that record precipitation data. These stations are mainly located in larger cities, causing a skewed distribution of coverage. This means that areas dedicated to agricultural practices are neglected. There is, however, open-source satellite data available on Google Earth Engine (GEE), which provides precipitation data in a grid-like fashion. This data, however, tends to have significant error in comparison to local station data. It's also reasonable that this is the case since the satellites cover much larger areas and need to average out the values for these areas. With the current situation in mind, it explains the need for more accurate data for areas not covered by local stations.

The use of AI and machine learning (ML) has become more widely used in today's society, seeing usages in almost every field. However, artificial intelligence and machine learning are not new fields. The theoretical foundations of AI date back to the mid-20th century, when Alan Turing and John McCarthy presented the fundamental concepts underlying machine intelligence (Russell & Norvig, 2020). The reason why it has become more widely used today is due to the accessibility of computing power, which was not available back then, at least not to the extent that is required to functionally develop and use the AI models. With exponential growth and development of the processing power of computers, the use of AI and machine learning is no longer beyond reach. The possibilities of AI are almost endless, and its functionality can be applied in most fields. AI is short for artificial intelligence and covers a wide set of sub-theories; machine learning is one of them. A further presentation of what AI and ML are will be included in the theory section.

This thesis is a case study on the possibility of integrating AI within hydrology. The two cases included in the study are leak detection in a Water Distribution Network (WDN) and improving data quality from satellite precipitation data. Both cases will center around the usage of an Artificial Neural Network (ANN). For leak detection, the data is provided by leakDB, which is an open-source dataset containing artificially generated data. The precipitation data is provided from GEE (Gorelick et al., 2017) and the Swedish Meteorological and Hydrological Institute (SMHI) (SMHI, 2025), which are real measurement data. The goal of the leak detection model is to be able to detect leaks and pinpoint 2 where in the network the leak has appeared. For the precipitation, the goal is to improve and remove bias from satellite data, in combination with providing quality data in areas not covered by stations. This is to lay the foundations for a forecast model. Furthermore, it will also investigate the possibility of replacing active stations or providing an area-specific model, which can be a replacement for investing in a new weather station.

The thesis is written in collaboration with Kentima AB, an automation and security company interested in investigating whether it is possible to integrate machine learning into their existing software. This could also be used as a proof of concept to demonstrate that machine learning can be applied to human-machine interface (HMI) and supervisory control and data acquisition (SCADA) systems. The use of machine learning in this type of system has two advantages. Firstly, the HMI/SCADA application already stores a significant amount of data that can be used to train the model. Secondly, this is already the application used by operators, and it would be very easy to use if the model's predictions were accessible directly through it. The models developed in this case are not intended for Kentima to use but rather to be able to use to demonstrate the technology for their clients.

2. Theory

In the following section, the theories that are used in the thesis are presented. These theories are the foundation from which these models are developed. It is necessary to understand the basics of these theories in order to understand the development choices and the thought process behind them.

2.1 Artificial Intelligence, Machine Learning and Deep Learning

Artificial intelligence is a broad and evolving field integrated into many areas and applied in a variety of ways. This makes it very difficult to give an exact definition of AI (Hunt, 1975). The foundation of AI is pattern recognition, problem solving, or machine comprehension. AI is vast and therefore has been divided into subfields, one of which is machine learning, see Figure 1. ML means using a set of data points to be able to make predictions and find patterns.

In ML, there are different models that can be used; some of them utilise deep learning (DL) (Goodfellow et al., 2016). In DL, networks are structured to resemble aspects of the human brain with nodes and connections between them. How the connections are used is hidden in a "black box".

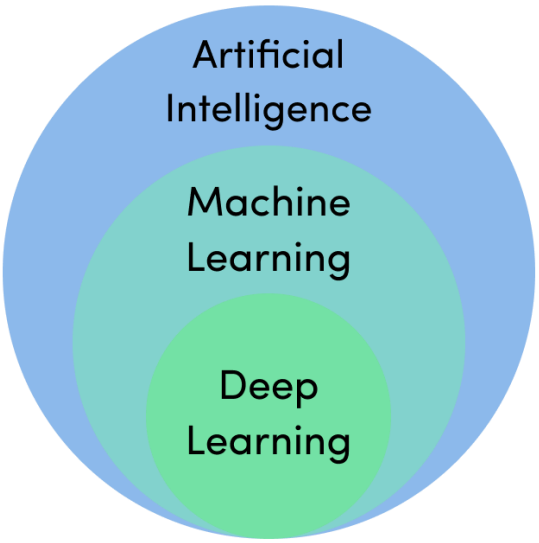

Figure 1: Artificial Intelligence (Goodfellow et al., 2016)

2.2 Machine Learning

An important part of ML is data exploration (Tukey, 1977). Before starting to train the model, it is crucial to understand, analyse, and prepare the data. Data exploration contains multiple steps to go from raw data to processed data that the model can use. The first and most fundamental step of data exploration is to identify what data is available, what it means, and the types of data. The next part is filling in missing data and removing outliers and anomalies. This is done to increase the accuracy of the model. Using visuals such as histograms, correlation matrices, and scatter plots can assist with locating some patterns early on. Once this is done, the data is divided into three parts; training, validation, and test data. Training data is usually around 70% of the data, and this is the data a model uses to train. Validation is also used during training, and this is what the model evaluates its progress against. When the

model is finalised, it is faced with test data that it has not seen at all during training. Training and validation data both usually represent 15% each, see Figure 2. Splitting the data into these three parts is standard practice and is done to prevent overfitting, which will be explained later in this section.

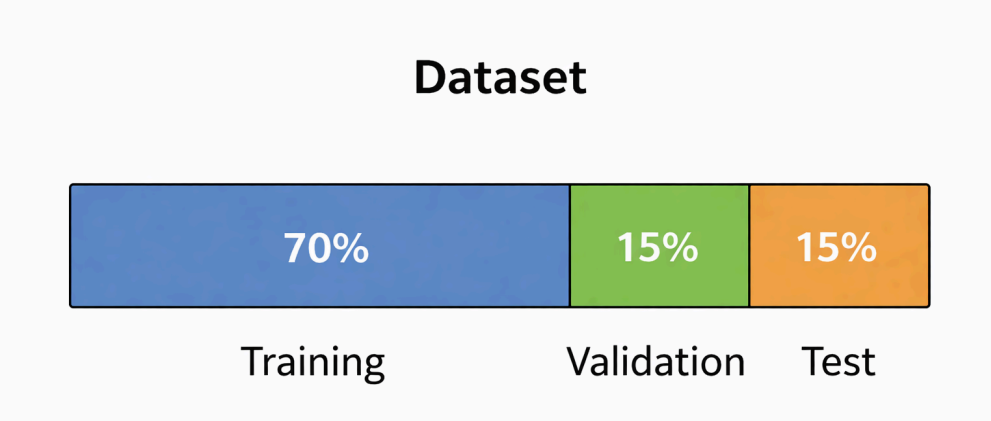

Figure 2: Splitting the dataset

Before feeding the data to the model, it is also advantageous to either standardise or normalise the data (Bishop, 2006). Standardisation rescales the data to have a mean of 0 and a standard deviation of 1. This is done using Equation (1) below, where $\hat{x}$ is the mean and σ is the standard deviation

$$x_{new} = \frac{x_i - \overline{x}}{\sigma} \qquad (1)$$

Normalisation rescales the data set so that each value is between 0 and 1. Equation (2) follows below, where x_{max} is the maximum value in the data set, and x_{min} is the minimum value in the data set.

$$x_{new} = \frac{x_i - x_{min}}{x_{max} - x_{min}} \quad (2)$$

Standardisation is commonly used when the data follow a Gaussian (normal) distribution and the model where the data are to be used assumes normally distributed features. Normalisation, on the other hand, is used when all features are in the same bounded range. This works well for ANN, as many of the activation functions work best in the range of [0,1] and [-1,1]. Activation functions will be discussed in the ANN section. Scalers can be used in combination with each other to further stabilise the data (Burbidge et al., 1988). The inverse hyperbolic sine (Arcsinh) is another scaler and is often used with data that is heavy-tailed. Besides this, arcsinh handles zeros cleanly and doesn't have to make exceptions or shift as a log scaler would. The formula for arcsinh is seen in Equation (3).

$$arcsinh(x) = ln(x + \sqrt{x^2 + 1}) \quad (3)$$

A more mathematical approach to deciding which to use is to apply a statistical test. The test applied is D'Agostino's K^2, which is appropriate for sample sizes greater than 300 (D'Agostino & Pearson, 1973). This test checks how the data follow a Gaussian (normal) distribution, which is the null hypothesis. The test is based on transformations of the data's kurtosis and skewness. Kurtosis measures how heavy or light the tails of a distribution are, which means that a high kurtosis represents a lot of extreme values, while a low kurtosis value means fewer extremes and a flatter distribution. Skewness measures whether a distribution is lopsided or asymmetric; a positive skewness means a tail to the right, while a negative skewness means a tail to the left. Kurtosis and skewness are calculated according to Equations (4) and (5), respectively. In some literature, kurtosis and skewness are denoted as $\beta 2$ and $\sqrt{\beta 1}$, respectively; however, in this thesis, they are denoted as g_1 and g_2.

$$g_1 = \frac{\frac{1}{n}\sum_{i=1}^{n}(x_i - \overline{x})^3}{(\frac{1}{n}\sum_{i=1}^{n}(x_i - \overline{x})^2)^{3/2}} \quad (4)$$

$$g_2 = \frac{\frac{1}{n}\sum_{i=1}^{n}(x_i - \overline{x})^4}{(\frac{1}{n}\sum_{i=1}^{n}(x_i - \overline{x})^2)^2} - 3 \quad (5)$$

With the calculated kurtosis and skewness, they can be transformed, which is denoted by *Z*. This thesis will not dive into the theory behind these transformations, as they fall outside the scope of this thesis. The final equation on which the hypothesis will be applied can be found in Equation (6). The null hypothesis holds if K^2 is approximately a X^2 distribution with 2 degrees of freedom. If the p-value is less than 0.05, then the data should be normalized, and otherwise, it should be standardised.

$$K^2 = Z_1(g_1)^2 + Z_2(g_2)^2 \quad (6)$$

An important thing to consider when applying either normalisation or standardisation is the handling of outliers or anomalies, since both of these methods can be heavily influenced by outliers/anomalies. For standardisation, it will shift the mean towards the outlier, which will increase σ. This can cause "normal" values to be closer to zero, which is not the case. For normalisation, this has an even greater effect due to it using x_{max} and x_{min}, which will typically be an outlier's value if it exists. If x_{max} is significantly larger than other values, it will cause the other values to collapse near zero. Reversely, for small values, it will cause it to collapse near 1. In order to combat these errors, an analysis of the data must be done to detect potential outliers/anomalies. This can be done through plots such as scatter plots, histograms, or time-continuous plots, depending on the data. When an anomaly is detected, it must be validated. Is it a true value or is it noise in the measurement? This applies mainly to real data, as artificially generated data generally does not have noise. If the values are confirmed to be anomalies

and are important for the model, a robust scaling is applied. Otherwise, they are to be scaled or potentially removed from the data.

2.2.2 Supervised Learning

Depending on the task, there are two ways to train a machine learning model. It has been divided into supervised and unsupervised learning (Mitchell, 1997). In supervised learning, the model has access to a training data set. The dataset contains both input data, also known as features, and a known target. Having this target means that the data is labelled. As an example, train a model to recognise whether a picture is of either flower type 0 or flower type 1. For each picture, the model knows if it was right or not. Once training is done, a new picture can be observed, and the model makes its prediction, type 0 or 1. Unsupervised learning instead, does not have access to a target. In this case, the model tries to cluster the pictures. The model does not tell type but creates groups on different features like colour, size, or shape of the leaves.

Supervised learning can then be split into two different categories, depending on the type of target used. The example used earlier in deciding which type of flower is a classification problem. The target is an option, and the model is either right or wrong, but supervised learning could also be a regression problem. In a regression problem, the target could be a value, and the model could be more or less right. As an example, inputs can be features of a throw and predict its length. In this example, the model is trained on how close the predictions are to the actual length of the throw.

2.2.3 Random Forest

Random forest (RF) is a machine learning model constructed from another model called a decision tree (Breiman, 2001). Instead of using just one decision tree, the RF model uses multiple for its prediction, see Figure 3.

Each of the decision trees gets a random subset of the full data to train on. For classification, the RF model answers what the majority of the decision trees' predictions are, and for regression, it is instead the average of their predictions. This makes the RF model better at dealing with larger datasets since it is less prone to overfitting.

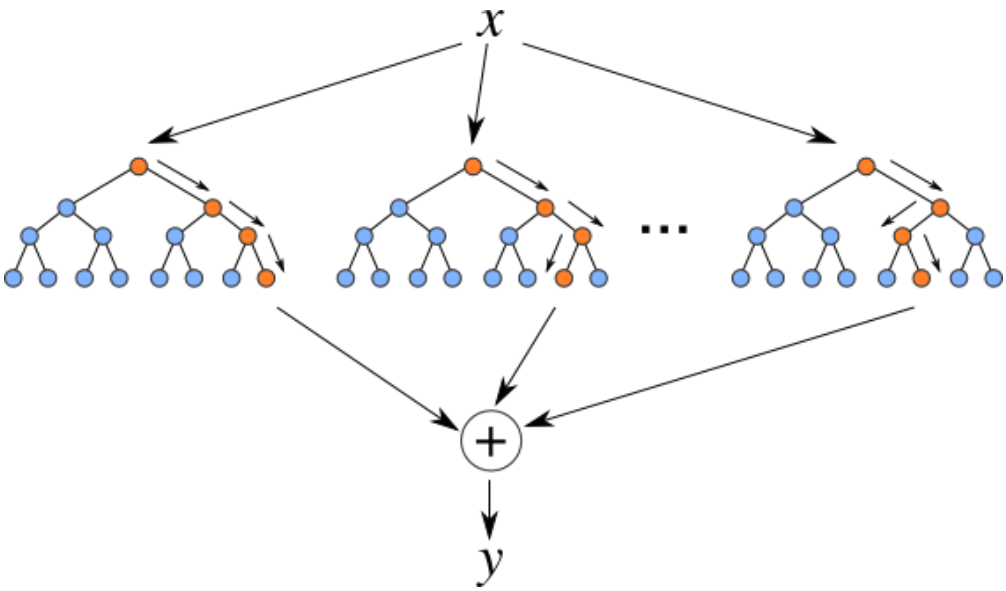

Figure 3: Random Forest (Sapakova, 2025)

A decision tree is constructed with a root node, decision nodes, and leaf nodes; see Figure 4. The root node contains the entire data set. Decision nodes see part of the dataset and split the data into new nodes. Leaf nodes are the furthest down in the decision tree, and they represent a prediction for the data that made its way there.

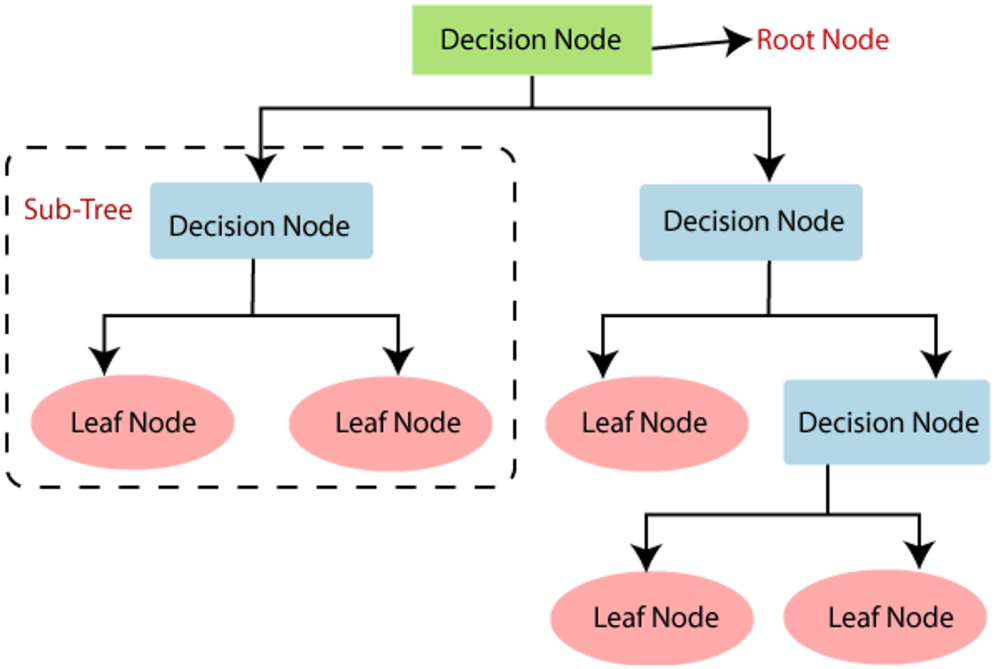

Figure 4: Decision Tree (Sapakova, 2025)

The way the decision tree is trained is node-by-node, calculating the Gini impurity. The Gini impurity for a decision node is a metric that determines how impure the nodes are after each decision. It is calculated as seen in Equation (7), and if a decision node scores 0, it has successfully divided all the inputs to the right prediction (Breiman et al., 1984). The decision tree tries different features and thresholds, and the combination that generates the highest purity is the one that the decision node will keep. In the equation, p_i is the proportion of data that belongs to class i. The n is the number of classes. Once the model is trained, new inputs follow the decision rules until it reaches a leaf node, where the decision tree outputs a prediction.

$$Gini = 1 - \sum_{i=1}^{n} p_i^2 \quad (7)$$

2.3 Artificial Neural Network

An ANN is a model inspired by the human brain (Goodfellow et al., 2016), where network nodes function similarly to biological neurons and are therefore often called neurons. This, in combination with layers, results in the ANN having the appearance of a web-like construction, as seen in Figure 5. An ANN consists of several layers, with a minimum of three. The first layer is called the input layer, which corresponds to the number of features that are available in the data set, plus one extra, which is the bias. The last layer is called the output layer. This layer can have several neurons as output, depending on the usage of the ANN. The layers in between are referred to as hidden layers. The number of hidden layers and sizes can be freely chosen by the designer.

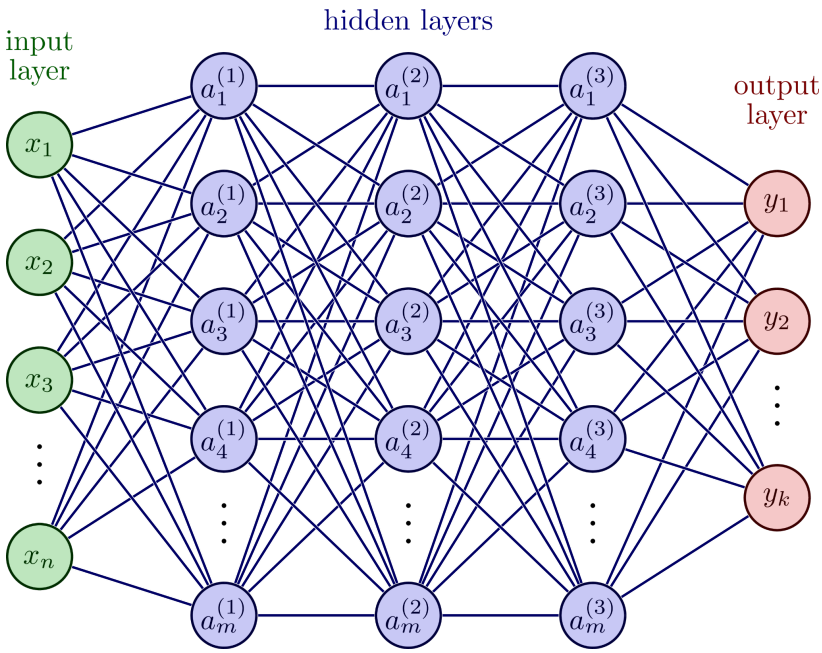

Figure 5: Artificial Neural Network (Shah et al., 2022)

In order to understand how an ANN functions, it is vital to have an understanding of its mathematical aspects. By examining only one of the neurons, the math becomes more manageable and easier to grasp. In order to make this even simpler, a forward propagation will be explored. Figure 6 below explains how the inputs interact with one neuron and how the weights are applied, as well as the bias.

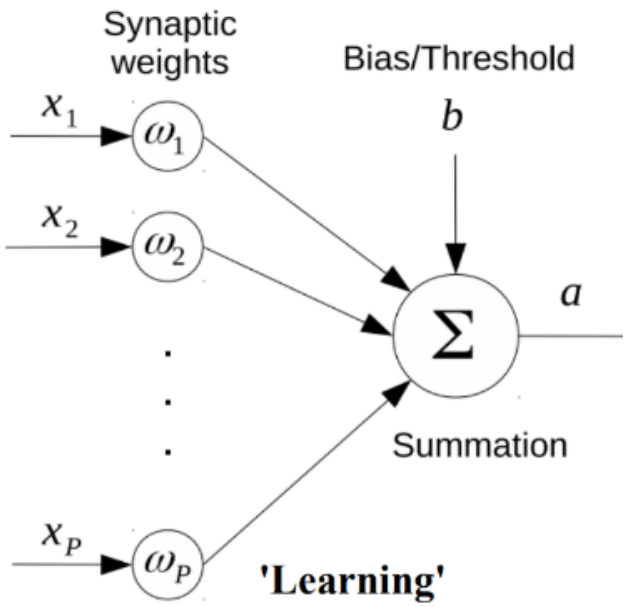

Figure 6: Single Neuron (Oshchepkov, 2024)

The input combined with the bias can be represented with Equation (8), where a is the function of the neuron prior to the activation function. The x's represent the input signals. The input signal can be the features if the neuron is in the first hidden layer, or it can be an output from a previous layer. The ω's are the synaptic weights; these are updated through each epoch in order to achieve a more accurate result. Finally, the b is referred to as the bias; this is in some literature referred to as ω_0. The bias is used with an input signal that has the value of 1 and only 1. This ensures that the bias is always determined by the value of ω_0. In this thesis, the bias will be presented by ω_0. The effect of this could be seen in Equation (9), which is the same as Equation (8).

$$a = \sum_{k=1}^{K} \omega_k x_k + b \tag{8}$$

$$a = \sum_{k=0}^{K} \omega_k x_k \tag{9}$$

Before a is sent as an output from the neuron, an activation function is applied. The activation function is set by the developer and must follow a few rules. A further presentation of what activation functions are and their effect was presented later in the thesis. For now, the activation function will be represented as γ. This gives the final output from a neuron, see Equation (10)

$$y = \gamma(\sum_{k=0}^{K} \omega_k x_k) = \gamma(\boldsymbol{\omega}^T \boldsymbol{x}) \tag{10}$$

2.3.1 Activation functions

Activation functions play a vital role in updates as well as in the outputs of an ANN(Goodfellow et al., 2016). As previously mentioned, activation functions are represented as γ functions and must abide by certain rules. Firstly, they need to be differentiable, which means they are continuous at every point in their domain. They also need to have an easily calculated derivative. This is not a strict rule in itself, but it greatly decreases the calculation time, since the activation function's derivative will be calculated multiple times throughout the training process. A few examples of activation functions can be found below, namely ReLU, tanh, sigmoid, and linear. Their equations and derivatives are presented in Table 1. One might notice that ReLU actually isn't a continuous function, but it still works as an activation

function. This is because the function only has an undefined derivative at exactly zero, but this is fixed by assigning a gradient when the neuron outputs exactly zero (which is very rare).

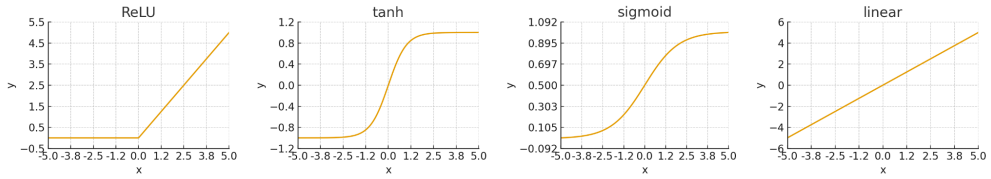

Figure 7: Activation functions

Name	$f(x)$	$f'(x)$
ReLU	$max(0, x)$	$(0, 1)$
tanh	$tanh(x)$	$1 - tanh^2(x)$
Sigmoid	$\frac{1}{1+e^{-x}}$	$f(x)(1 - f(x))$
Linear	x	1

Table 1: Activation functions and their derivatives

2.3.2 Loss computation

Once an output is calculated from the neuron, the "learning" part is applied. This is done through a loss computation using a cost function, which will backpropagate through the network and update the weights (Goodfellow et al., 2016). The cost function can be in various forms and be more useful for certain tasks. For example, mean square error (MSE) and mean absolute error (MAE) are commonly used for regression tasks, and cross-entropy (CE) and hinge are used for classification tasks. In the case of detecting leaks, it falls

under the category of classification, while precipitation falls under regression; cost functions should be chosen accordingly. The function for CE can be found below in Equation (11). In the function, *N* represents the number of samples. The summation starts from 1 rather than 0; this is due to the bias. Since the bias isn't a real sample, it does not have a comparable target value. Hence, it was removed from the cost function. The target value is represented by dn, so it is through the cost function that one compares the calculated value to the true value.

$$E(\boldsymbol{\omega}) = -\frac{1}{N}\sum_{n=1}^{N}[d_n log(y_n) + (1-d_n) log(1-y_n)] \tag{11}$$

In order to decrease the error through the learning process, the weights needed to be updated. This is done through backpropagation using gradient descent. The logic here is to move the error to its lowest value. With gradient descent, one calculates in which direction the error decreases the most in order to reach the minimum point. Using gradient descent, the weights are updated with Equation 12. In this equation, η is introduced, which represents the learning rate. This is an input set by the developer. The learning rate is used to restrict the step size that the model takes in order to minimise the error. This value needs to be fine-tuned through the development process, since a too large value can cause the model to miss the minimum point as the step size is too large. On the other hand, a too small value can have the same effect since it's not able to reach the minimum as the epochs run out. A visual representation of this can be found in Figure 8.

$$\omega_k^{(new)} = \omega_k^{(old)} - \eta \frac{\partial E}{\partial \omega_k} \tag{12}$$

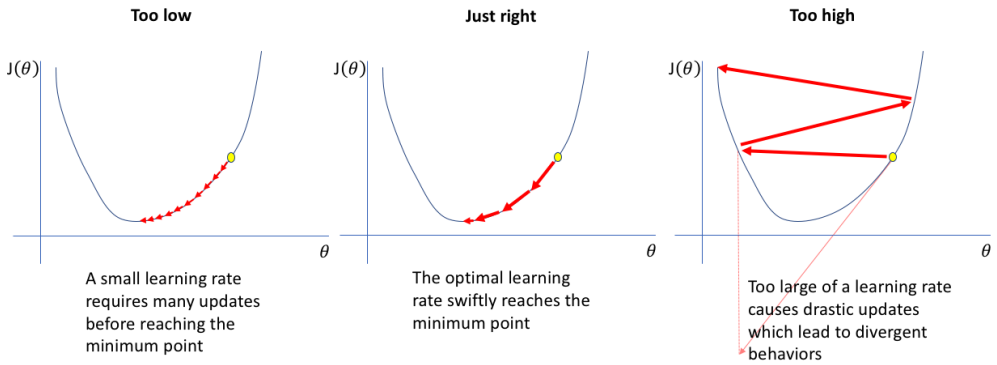

Figure 8: Learning rate (Jordan, 2018)

In this thesis, the primary cost functions used are MSE, weighted MSE, and CE. The MSE and weighted MSE are used for precipitation data, while the CE is used for leak detection. This fits since precipitation is a regression task, while leak is a classification task. The main difference between MSE and weighted MSE is the weights. In a regular MSE, the cost is computed as the average of the squared differences between prediction and true value, and for the weighted, this is scaled with a factor. The way this factor is scaled could be seen in Equation (14). The scaling factor aims to give higher weights to larger values or smaller ones, depending on the need.

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \tag{13}$$

$$WeightedMSE = \frac{1}{N}\sum_{i=1}^{N} w_i(y_i - \hat{y}_i)^2 \tag{14}$$

2.3.3 ANN summary

With the mathematical background explained, the training process of the ANN can be presented in three steps.

1. All ω_k are initialised with small random numbers, including the bias ω_0
2. The learning rate η is defined in the range of $0 < \eta < 1$
3. Repeat until convergence or the number of epochs
   a. Every node's output is calculated, leading to the output layer's output $y_n = y(x_n)$
   b. The derivative $\partial E / \partial \omega k$ is calculated.
   c. The weights ω_k are updated as seen in Equation (12)

This will result in a trained model. In order to evaluate the quality of the model, several aspects will be analysed. These aspects are presented in the optimisation section as well as improvements to be made to a model.

2.3.4 Residual Artificial Neural Network

For precipitation, a residual artificial neural network (ResNet) is used. The ResNet is an extension of the standard ANN by adding a residual component (He et al., 2016). For this model, a linear residual component is used and applied as a linear shortcut. This extends the model to having a linear component in addition to the deep nonlinear mapping. A residual component allows for more stable and faster training, as well as improving the generalisation of the model. The application of the ResNet is explained in the formula below, where ω_x is the linear component. The x denotes the input, while ω is a weight that is updated through the same process as ω^T .

$$y = \gamma(\boldsymbol{\omega}^T \boldsymbol{x}) + \omega x \qquad (15)$$

2.4 Optimization

2.4.1 Overfitting & Underfitting

It is important to make sure that the model is trained to make predictions on new data. This is why the data are divided into three parts. The goal of the model is not to be 100% accurate in the training. A good model is determined by how well it performs on the validation/test data.

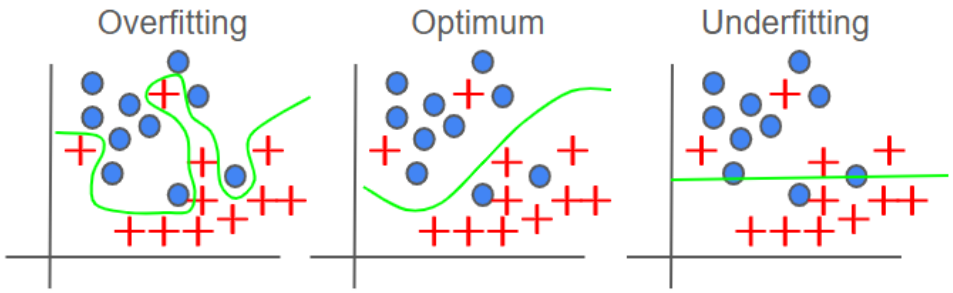

Figure 9: Underfitting, optimum and overfitting, generated for visualisation by authors

The graph in the middle of Figure 9 illustrates how a well-trained model makes its predictions. Not all predictions are correct, but the ones that are off are most likely noise or outliers and not representative. A more complex model, seen in the left graph, finds all the outliers in the training data. This has resulted in a model where the outliers have too much of an impact. Then, in the right graph, there is a model that is too simple. In this case, the model has not been able to find any of the patterns in the data. Overfitting and underfitting are important to avoid, and that is why a model is not evaluated on its training data (Goodfellow et al., 2016). It's important to note that overfitting tends not to be applicable when the data is artificially generated,

due to the data not containing any errors or noise, which means that the case of overfitting doesn't apply. The reasoning for underfitting still stands for artificial data.

2.4.2 Feature optimisation

To achieve a well-trained model, it is important to have good features. Not all data that can be collected is worth using, and sometimes the data at hand is not enough. If a lot of features are available, it is not necessary to use all of them. This is when feature elimination is applied (Guyon & Elisseeff, 2003). Feature elimination can be done in a few ways. A common one is feature importance. Feature importance looks at the mean decrease of impurity in the case of RF. This way, features with low impact can be removed to improve the functionality of the model.

If a model does not reach a satisfactory result, it is possible to create new features from the accessible data. This is called feature engineering and works in a couple of different ways. If the data is in a time series, then rolling features are common. It means looking at a group of previous data to make the prediction. Rolling features could be the mean, max, or min of the last n inputs. Feature engineering can also be giving the model the already existing data in new forms. It can be squaring, multiplying, or subtracting different features. There is not one right way to create and apply this, but rather a trial and error of combinations of the available features. A common starting point is to use logically combined features, meaning that the user already knows that the relations of these features are important, which helps the model to see that.

2.4.3 Regularisation

Regularisation can be applied to limit the model's performance on the training data in order to reduce overfitting. This is done through constraints on the weights, commonly called L1 (Tibshirani, 1996) and L2 (Hoerl & Kennard, 1970). They are applied in the cost function with a regularisation strength. Which determines the extent to which the regularisation is applied. L1 encourages the weights to be zero, resulting in a sparse network, while L2 penalises large weights more. The regularisation is applied through Equation (16), where α denotes the regularisation strength, and Ω is the regularisation function. The L1 and L2 are presented in Equations (17) and (18).

$$\tilde{E}(\omega) = E(\omega) + \alpha\Omega(\omega) \tag{16}$$

$$L1 = \Omega = \frac{1}{2}\sum_i \omega_i^2 \tag{17}$$

$$L2 = \Omega = \sum_i |\omega_i| \tag{18}$$

Another way to reduce overfitting is to apply dropout. The dropout function works by applying a probability of p to each neuron. In every step of the training, this probability is rolled to determine whether that neuron should be dormant in that step. This allows the ANN to have a balanced reliance on the neurons, which limits the network from being dominated by certain neurons.

2.5 Evaluation Metrics

In order to evaluate the models' performances, several metrics will be used. This is done to evaluate on a broader basis, which allows capturing more of its aspects. The metrics used for precipitation models are root mean square error (RMSE), mean absolute error (MAE), mean bias error (MBE), R^2 and index of agreement (IOA). The leakage problem is a classification problem. Therefore, it will be evaluated on a different set of metrics. When looking at

binary classification, there are four different outcomes. True positive (TP) predicted a 1 correctly, true negative (TN) predicted a 0 correctly, and false positive (FP) and false negative (FN) refer to predicting a 1 or 0 wrongly, respectively. The metrics looked at with classifications are built upon these outcomes. When the classification problem isn't binary, a common evaluation strategy is to look at the average by looking at the different classes as binary. The metrics commonly used to evaluate classification tasks are accuracy, recall, precision and F1 score. These metrics were presented further below in this section and are significant for the model development.

2.5.1 RMSE

The RMSE is a common metric used in regression tasks since it shows the average magnitude of prediction error. This error is presented in the same unit as the target value, which in this case is precipitation in mm. Since it has the same unit as the target, it will provide a clear numerical evaluation of the model. The formula for RMSE is presented in Equation (19); in this formula, n denotes the number of samples. The $\hat{y}$ is the predicted value, and y is the true value.

$$RMSE = \sqrt{\sum_{i=1}^{N} \frac{(y_i - \hat{y}_i)^2}{n}} \tag{19}$$

2.5.2 MAE

The MAE is a measurement of errors between the paired predicted value and true value, see Equation (20). Similar to RMSE, this also provides a numerical value to evaluate the model on. However, the MAE treats all errors the same and doesn't put extra weight on larger errors. This provides an error

that isn't skewed by large errors due to missed peaks or anomalies. The notations in MAE are the same as in RMSE.

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i| \tag{20}$$

2.5.3 MBE

The MBE measures the average bias in predictions. This provides indications if the model under- or overestimates the true value. MBE provides good insight into how the model is operating and how to evaluate the results, since it provides a direction of error. A negative MBE shows that the model is underestimating, while the opposite is true for a positive value. Optimally, this value is close to zero, but this metric should not be used alone since it doesn't provide a full picture of evaluation, since a model can have an MBE value of zero but still be completely off in its predictions. MBE is fairly simple to calculate using Equation (21).

$$MBE = \frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i) \tag{21}$$

2.5.4 R^2

R - squared or, in some literature, "coefficient of determination", is a metric that provides a numerical value of how well the variance in the target is visible in the model (Weisberg, 2005). This metric is also commonly used in regression models, since it provides how much better the model performs than just predicting the mean. The R^2 value is normally in the range of 0-1, where 1 means that it explains 100% of the variance, which is rare. A

negative value indicates that it is worse than just using the mean, see Equation (22).

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{N}(y_i - \overline{y})^2} \tag{22}$$

2.5.5 IOA

The Index of Agreement is a statistical metric used to measure how well the model prediction matches the observed data. The metric is a common metric used in environmental, hydrological, and climate modelling. It provides a numerical value normally between 0 and 1, normally. It's possible for it to be less than zero, but this only happens when the model prediction is worse than just using the mean. IOA's measurement is how well predictions match both the magnitude and pattern of the observed data, which is a more compact measurement than the other metrics. The IOA considers how close the predictions are to observation, how well the variability is captured, and how well the large errors are relative to the natural variability in the data. For reference, an IOA value of 0.5-0.7 is moderate agreement, while 0.70-0.90 is good agreement. The formula is presented in Equation (23), and the notations follow the same as earlier presented metrics.

$$IOA = 1 - \frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{N}(|\hat{y}_i - \overline{y}| + |y_i - \overline{y}|)^2} \tag{23}$$

2.5.6 Accuracy

Accuracy is the simplest one. It is just a percentage of how many predictions were correct. Accuracy is a good, reliable measurement for balanced data. For imbalanced datasets, accuracy could easily be unrepresentative. For a dataset with 95% negative and 5% positives, the accuracy could get very high by exclusively predicting negative for every input. The calculation for accuracy for binary classification could be seen in Equation (24). In a multiclass classification problem, it is calculated the same way, with the number of correct predictions divided by the total number of predictions.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{24}$$

2.5.7 Recall

Recall is a metric used to determine how well a model can predict true positives. Recall can also be called sensitivity and is used when it's very important not to miss any true positives. The recall does not consider the number of false positives the model has, see Equation (25). In multiclass classification, the equation is calculated for each class and then the average of that.

$$\text{Recall} = \frac{TP}{TP + FN} \tag{25}$$

2.5.8 Precision

Precision is the counterpart to recall. Instead of the sole focus on not missing any positives, precision is the metric of how many of the predicted positives are actually true positives. It's calculated as seen in Equation (26), and like recall, also class-wise when it is not binary classification. When training the

model, the balance between recall and precision is important and varies depending on the problem at hand. Focusing on recall can result in a lot of false positives, and focusing on precision could mean predicting a lot of false negatives. The precision metric is used when it is important not to predict false positives.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{26}$$

2.5.9 F1 Score

The way to look at the balance is by calculating the F1 score of the model, see Equation (27). Since the recall and precision both can be misleading, the F1 score is often a more dependable metric. The score will be between zero and one, and a well-balanced model will score closer to one. The F1 score should be used when the model mustn't over-predict in either direction.

$$\text{F1 Score} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{27}$$

2.6 Satellite products

The precipitation measurements were gathered through Google Earth Engine (GEE). This is an open-source platform that allows satellite products to publish their data. These satellites gather all types of data, but for this thesis, the precipitation will be the focus. The satellite products used are: CDR, CFSR, CPC, ERA5, and GPM. These products vary in provider, measurement frequencies, and pixel size. A rundown of each will be provided in this section. Important to note is the pixel size. Due to the satellites having an extremely large coverage range, they aren't able to gather data for every specific point. So they have been summarised to an average on a pixel. These

pixels are placed in a grid-like manner in order to cover the covering range. An example of this is provided below. So in order to retain the most appropriate pixel, a buffer range from the desired location is applied. The pixel that covers most of the buffer zone is picked. Another reason for this is that these pixels can overlap, so a certain location can exist in two pixels. The buffer zone ensures that the pixel that correlates best is picked.

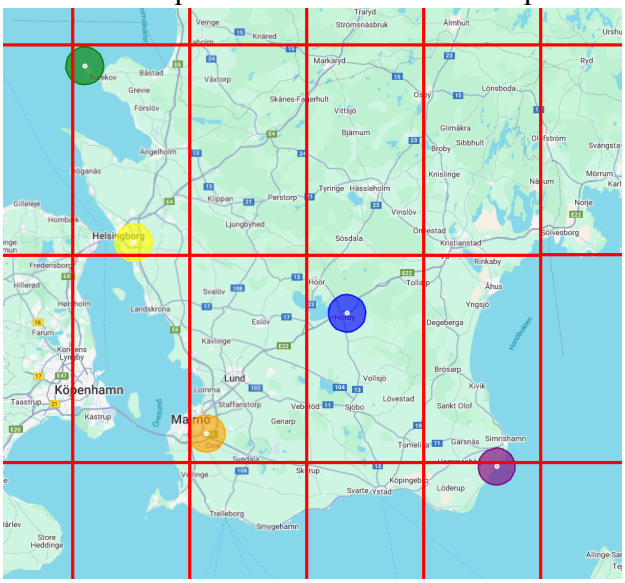

Figure 10: Pixels

2.6.1 CDR

The Persiann - CDR is provided by the National Oceanic and Atmospheric Administration (NOAA) and the National Climatic Data Center (NCDC). NOAA is an agency that is located in the USA and tasked with monitoring oceanic and atmospheric conditions as well as forecasting weather. NCDC has now merged with NOAA and is part of that agency. The full name of the

product is "PERSIANN-CDR: Precipitation Estimation From Remotely Sensed Information Using Artificial Neural Networks-Climate Data Record", which collects daily data. The product is based on the neural-network methodology described by Ashouri et al. (Ashouri et al., 2015). The CDR uses GridSat-B1 IR data that is derived from merging ISCCP 1B IR data along with GPCP v2.2. This product provides a pixel size of 27 830 meters.

2.6.2 CFSR

CFSR is provided by NOAA and the National Centers for Environmental Prediction (NCEP), which is a part of NOAA. The full product name is "CFSR: Climate Forecast System Reanalysis" and has a cadence of 6 hours (Saha et al., 2010). The CFSR was designed as a high-resolution, coupled atmosphere-ocean-land-surface-sea-ice system. It has been extended as an operational real-time product, which has data from 2018 to the present (2025). The CFSR offers a pixel size of 55 669 meters.

2.6.3 CPC

CPC is provided by NOAA. The full name of the product is "CPC Global Unified Gauge-Based Analysis of Daily Precipitation", and it has a pixel range of 55 500 meters. The CPC also uses gauges for its measurements. Gauges are true values provided by stations. This is used through statistical interpolation methods to produce a continuous grid. This has a fallback that local downpours are smoothed out over the region. The methodology of the gauge-based solution is explained in previous studies (Chen et al., 2008; Xie et al., 2007).

2.6.4 ERA5

The ERA5 is provided by the European Center for Medium-Range Weather Forecast (ECMWF) in collaboration with Copernicus Climate Change Service (C3S). The ECMWF is an independent intergovernmental organisation supported by the EU. Their product ERA5 is a reanalysis that combines model data with observations made globally into a combined data set. It has a cadence of 1 day and a pixel size of 27 830 meters (Hersbach et al., 2020). The precipitation is provided as a daily sum in meters.

2.6.5 GPM

"GPM: Global Precipitation Measurement (GPM) Release 07" or GPM for short, is provided by NASA. The GPM is an international satellite project that aims to provide next-generation observation of rain and snow. It works with a cadence of 3 hours and a pixel size of 11 132 meters. The GPM is described in an earlier thesis (Huffman & Tan, 2020)

2.7 LeakDB

The simulated data of the WDN is an open-source dataset known as LeakDB (Vrachimis et al., 2018). The dataset was constructed to further the research and development of algorithms that can detect and isolate leakages. With the dataset, there is also a scoring algorithm that can be used. The goal of this open dataset and the scoring algorithm is to create a benchmark for fairly evaluating all different types of leak detection.

The scenarios are created with the Water Network Tool for Resilience. This is a Python package created to assist in investigating the resilience of water distribution systems. The artificially created demands are based on real data and created with three different parts. The demand data has a weekly and a yearly periodic component combined with a random component to make the data more realistic. Flow and pressure are then generated to fit the demands.

At the point of this thesis, there were 1000 scenarios available. Each scenario is set up during a year with measurements every 30 minutes. The nodes are linked with each other in the same order in each scenario, but the distances between them vary. The dataset contains flow, demand, pressure, labels, leak information, coordinates, distances, and pipe information. How the nodes and links are connected can be found in an inp-file within the LeakDB data set. The graph plotted in Figure 11 was generated with EPANET. The numbers in blue and red represent the NodeID and LinkID, respectively.

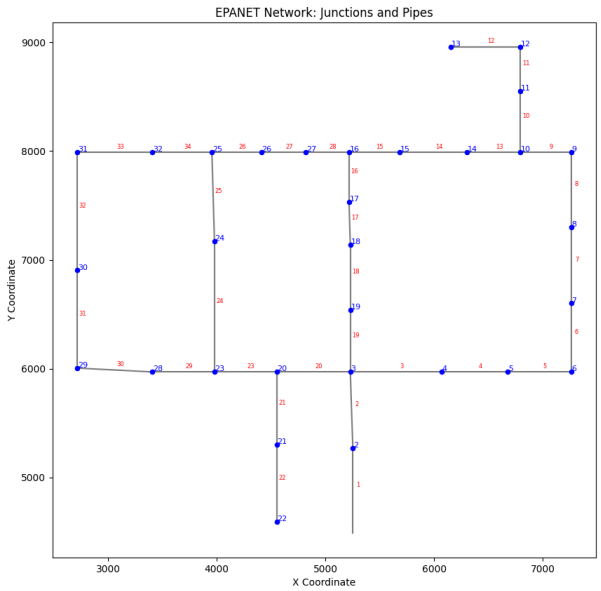

Figure 11: Hanoi network

Not all the scenarios actually have leaks, but for the ones that do, the leaks are either abrupt or incipient leaks. Incipient leaks are typically smaller

cracks that have the possibility of growing. This type of leak is often harder to find than an abrupt leak. An abrupt leak is often caused by a burst, and the leak will not necessarily grow as time goes on

3 Method

3.1 Precipitation

Here, the method for the real precipitation data will be covered. This section explains the stages: Data exploration, Model development and optimisation. The data exploration covers the acqusition of data and features that are used for the final model. The model development section explains the development processes and the evaluation metrics. Finally, the optimisation section covers the fine-tuning of the models.

3.1.1 Data Exploration

The initial approach for deciding which locations to use for the model was fully dependent on the available data from SMHI. SMHI is the Swedish Meteorological and Hydrological Institute and is a government agency responsible for gathering temporal data. These models' focus was on the region of Skåne, so an analysis of data availability of the region was conducted in order to decide on the final locations. For the region of Skåne, there were 27 available stations that gathered data on precipitation; however, of these 27, only a handful had data before 2023. This is important since only having data from 2023 can lead to insufficient data for model development. The chosen stations had data from 1995 to the present and a good geographical spread. The geographical locations can be found in Figure 10. The stations picked are: "Malmö", "Hörby", "Skillinge", "Hallands" and "Helsingborg". From these stations, the daily precipitation (mm) data were collected from 2019-05-10 to 2024-06-02. These dates were picked because earlier dates had large data leakage. While it was an option to artificially reconstruct the missing data, this was ultimately dropped, since the data was sufficient and reconstructed data could be hurtfulfor predicting the true values. In addition to precipitation, longitude, latitude and elevation were also

collected from SMHI in order to be used as features and for the GEE to extract data from the correct location. With the stations and locations picked, the extractions from GEE were performed. The GEE data were collected from the previously mentioned satellite products. For some, the precipitation needed to be transformed into daily. From the ERA5, additional data was collected, namely relative humidity, temperature, total cloud coverage and wind speed. This was done due to their central role in predicting precipitation (Doe & Smith, 2025). The data was then gathered in one CSV containing all measurements from all stations. This resulted in a total of 70 thousand data points for the model to be trained on, which means 14 thousand per station. More engineered features were created based on the data.

With the data properly stored and collected, the preprocessing began. In order to determine what type of scaler to apply, the data needs to be plotted to get a grasp of its distribution. From the plots, it's clear that the precipitation data was heavily stacked at zero with a heavy-tail containing large downpours. The K^2 test was also applied, which identified whether the data should be normalised or standardised. However, since some of the data is heavy-tailed, we want to apply a scaler to correct this as well. So the arcsinh scaler was applied to the data that fit the description. Following this, the standard scaler was applied to all the data.

In order to expand the accessible data, engineered features were created. These engineered features are focused on creating a larger picture based on the available data. The ANN benefited from this as it got more features to train on, as well as finding easier correlation between these engineered features and the precipitation data. The engineered features consist of lag, wet/dry indicators, rolling features and seasonality features. It is important to note that for all the engineered features, the true values are kept out. I.e SMHI data. The reason for this is that the spatial model should work on areas without SMHI data.

The amount of lag for each model varied, depending on the evaluation of the result as well as the quality of available data. Furthermore, lag also consists of differences, which are calculated from each satellite's individual value from x-days ago. This allows the ANN to pick up comparable differences between the satellite products and take it into account in its prediction. The lags were also kept less than 14 days since further lag would just be interpreted as noise to the model and result in a less accurate model.

The wet/dry indicators were based on the mean of all the satellite products. This mean was used to indicate whether it had rained more than 0.1 mm over a changeable period. This period was held to less than a week for all models. This allows the model to further understand the seasonality of precipitation, as it has a direct value linked to dry periods.

Two rolling features were used, namely wet probability and intensity. The wet probability uses the same threshold as the wet/dry indicator, but instead it checks the satellite products data individually if it is over or under the threshold, which is returned as a factor that acts as a probability. The number of days included in each roll varied depending on the model, but was also limited to less than a week. The rolling intensity worked similarly to wet probability but instead returns the average rain intensity over the given time period. These rolling features give the model more memory, allowing it to understand rainfall history. Wet probability helps with wet periods, and intensity helps with heavy rain periods.

Finally, are the seasonality features. These features were created so the model understands the variation of the seasons. Since precipitation is heavily related to seasons, this is an important feature for the model to understand. The seasonal features were created with sin and cos transformations. This type of encoding creates a form of mapping connected to the edge of a circle. The number of points in the circle's edge is dependent on whether it's a year, month, week, or day.

3.1.2 Model development

Once the data exploration and preprocessing were done, the model development began. The initial approach was to deploy a grid search, which varied the models' depth and size. In addition, the grid search also varies the learning rate and batch size. Since this process is time-consuming, an early stopping was applied to hasten the process. The early stopping was monitoring the validation loss. If an improvement hadn't been found in 12 epochs, the training stopped, and the grid search continued with the next configuration. The grid search saved the best models based on metrics. These metrics were RMSE, MAE, MBE, R^2, and IOA, with a weight favouring a low RMSE as well as high R^2 and IOA. These metrics were given the largest priority due to their indication of model fitness and error evaluation. Once the grid search had been completed, it returned a list containing the best model architectures. This list held the top 10, which were moved on to model optimisation. The decision to hold the top 10 rather than just one was that in the optimisation stage, the loss function was slightly altered. This could have caused a swing in model performance. Furthermore, in the optimisation stage, the engineered features as well as other features were evaluated. This was done to see if the performance would increase if the features were dropped.

3.1.3 Optimisation

The first stage of optimisation was applied to the features to determine their importance for the model. Here, the features from the top 10 models were evaluated in order to find the optimal configuration for each of them. The importance of the satellite precipitation data was ignored in this optimisation. Since the data from each satellite product isn't that large, all the data is needed. This optimisation looked at the importance of each engineered feature and how it affected the result. In this scenario, the results were evaluated by the same metrics as the model development. With the optimal

combination of engineered features discovered for the top 10 list, the next grid search was applied.

The next grid search had a varying alpha in the weighted MSE loss function. Since the precipitation data is dominated by zeros, a larger weight of importance was placed on the values that aren't zero. A spanning dropout probability was placed between the hidden layers, which varied in the grid search. This is to reduce the overfitting and expand the search for the optimum configuration. An L2 regularisation was also applied. The reason for L2 was based on the feature evaluation, since most features and engineered features contributed a little to the result. This indicated that L2 should be used rather than L1. The L2 regularisation strength was also varied in the grid search. In order not to have an excessively large list to pick from in the final evaluation, all configurations that didn't improve from the best model architectures' performance were not saved. The grid-search was also limited to a top 5 per architecture if they met the prior requirement. This resulted in a max list length of 50. From this list, the metrics were reevaluated, and the best combination was picked for the final model.

For the spatial model, an additional final test was applied. The final test was on a new, unseen station for the model, Hästveda. This station is also monitored by SMHI, which is a requirement in order to evaluate the model. In order to test and ultimately apply the model. All the features had to be in the same order as the model was trained on. If this isn't met, the model will perform extremely poorly as it interprets the values incorrectly. The scaler must also be applied in the same way; the values could get too large or too small, causing more misinterpretation by the model.

For the temporal prediction models, the same rules as for the spatial model apply when testing. However, since the temporal prediction models are area-specific, they will be tested differently. Their time series was split into 80/20, resulting in the test data being the final 20% of dates in the time series. The option of shuffling the time series was available, since the lag and

previous days' features were precalculated for each day. However, this wasn't applied since a complete time series allowed for a better comparison of results as well as a more informative plot of its performance.

3.2 Water Leakages

This section explained what was done with the simulated leakage data. From data exploration to a developed and tested model.

3.2.1 Data Exploration

The data used for the leak detection came from the LeakDB dataset. As mentioned in 2.7, this is an extensive amount of data containing over 1 000 scenarios for the specific WDN. Each scenario contains between zero and two leaks that last for varying durations. As seen in Figure 11, there are 32 nodes connected with 34 pipes. The data for the nodes contains demand, pressure, and whether it is leaking, and for the pipes, the available data is flow. Each scenario is divided into 17 000 indices representing a year. Since the data is simulated, there are no other features outside the LeakDB dataset that can be collected.

The available data in the LeakDB dataset are demand, pressure, flow, metrics of the pipes and coordinates. In a real WDN, the distances between nodes would not change from scenario to scenario and have therefore been left out in the development of this model. Studies have shown that the pressure in a pipe drops if a leak occurs (Zhang et al., 2023). Logical reasoning also says that if there is a leak, then more water has to flow to be able to meet the demands in each node. Apart from being available in the dataset, this is why pressure, demand and flow were chosen as features.

The large amount of data available would be hard to work with, and therefore, only a selection of scenarios was used. The selection contained 60 scenarios with leaks, where half of them were abrupt, and the other half were incipient. Reducing the number of scenarios was not enough to ensure the best model possible, since the dataset was still heavily imbalanced. Most of the leaks did not last for longer than 1 000 of the 17 000 indices, meaning that the large majority of the data were on non-leaks. For each scenario, the starting index of the leak was set as a reference, then the 1 000 indices prior to and after were collected to further balance the data.

After balancing the data, histogram plots of the data were plotted to determine if and how to scale it. As seen in Figures 27 and 28 and also calculated with Equation (6), the best scaling was to standardise the data. The histogram plots were done without node 1 since node 1 has a very distinct set of values due to node 1 being a water source rather than a normal node. Since this node worked a little differently and never had any leaks in the scenarios, this was excluded from the data.

The features that were then used for the first training attempt were demand and pressure for each of the 31 nodes and flow for the 34 pipes. After a few attempts at improving the model, the direction of flow was included to expand the number of features. With the inp file, it was possible to see which pipe went between which nodes, and the flow could be both positive and negative. From this, "flow in" and "flow out" from each node were derived and used instead of just flow.

More engineered features were added over time, such as pressure times flow in, pressure times flow out and pressure squared. Since the data is a time series, it was also possible to add rolling features for each node. Rolling mean, standard deviation, max and min were added for all features, original and engineered. With around 1400 features to work with, a feature elimination was a necessity. The elimination was based on the training results of the RF model. The feature elimination worked by group, meaning demand

for all the nodes were summed up as a group of features, and during the elimination, a group was eliminated per run rather than a single feature.

Looking at the labels for the data, two different sets were created. The first one being binary, with 0/1 representing no leak and a leak in the system as a whole. The second was constructed as a multi-binary array of 0s and 1s representing a leak or no leak in each specific node in a time step.

3.2.2 Model development

A few attempts to train and develop both ANN and RF models were conducted before, during and after the data exploration. Important to note is that the development was tested in two separate ways. The first idea was simply to have an ANN model locate the leak. This would then be done with the multi-binary array as labels.

Due to the poor results with just a single ANN, a different approach was developed. A pipeline solution was implemented where an RF model would first predict a leak or no leak in the entire system. If the RF model found a leak, then an ANN model would locate it. In this pipeline, the two models were trained separately but on almost the same data. The RF model used all indices and the binary label for the entire WDN. The ANN model, instead, was only trained on the indices where there was an actual leak and the array as labels. This means that the ANN had never seen an index without leaks. In the pipeline, the RF first made its prediction, and every predicted leak was sent to the ANN that predicted the location of the leak.

Grid searches for both of the solutions were made for all incoming hyperparameters and architectures. Unfortunately, no changing of parameters could make the ANN solution comparable to the pipeline, and therefore it was dropped. For the RF 100, 200, 300 and 500 numbers of estimators were tested and thresholds between 0.50 and 0.95 with a step size of 0.05. For the

architecture of the ANN, one to five layers with sizes 256, 128, 64, 32 were tested with all possible combinations. The hyperparameters for the ANN were learning rate, batch size and epochs. The batch sizes used were 128, 64, 32 and 16. Learning rates were 0.001, 0.005, 0.01, and 0.05. The epochs were also grid searched, but eventually not important after the optimisation. All grid searches were evaluated on the accuracy of the models.

During the optimisation, early stopping was implemented. This is why the number of epochs was disregarded. The early stopping had a patience of 10, meaning that if the value loss of the model did not change over the last 10 epochs, the training was considered done. This is a common step to add to ANN models to decrease the time of training. Apart from early stopping, node dropouts were also added. A node dropout rate for each step was added to reduce the risk of overfitting. To further improve the result of the ANN model, weighted features were also tested, but without any noticeable change.

4 Result

4.1 Precipitation

In the following segment, the results from the precipitation models are presented. The preprocessing was the same for both the spatial and temporal prediction. In the model section, each variation's result was presented clearly. A further evaluation of the results was conducted in discussion, where the performance was compared to the satellite's individual performance.

4.1.1 Precipitation Preprocessing

The data was initially plotted as a histogram to give a general presentation of its distribution. From the graph below, it was clear that all the precipitation data needed scaling, while the relative humidity, temperature and wind speed already followed a typical bell curve form. This means that a standard scaler was sufficient for RH, T_C and wind_speed.

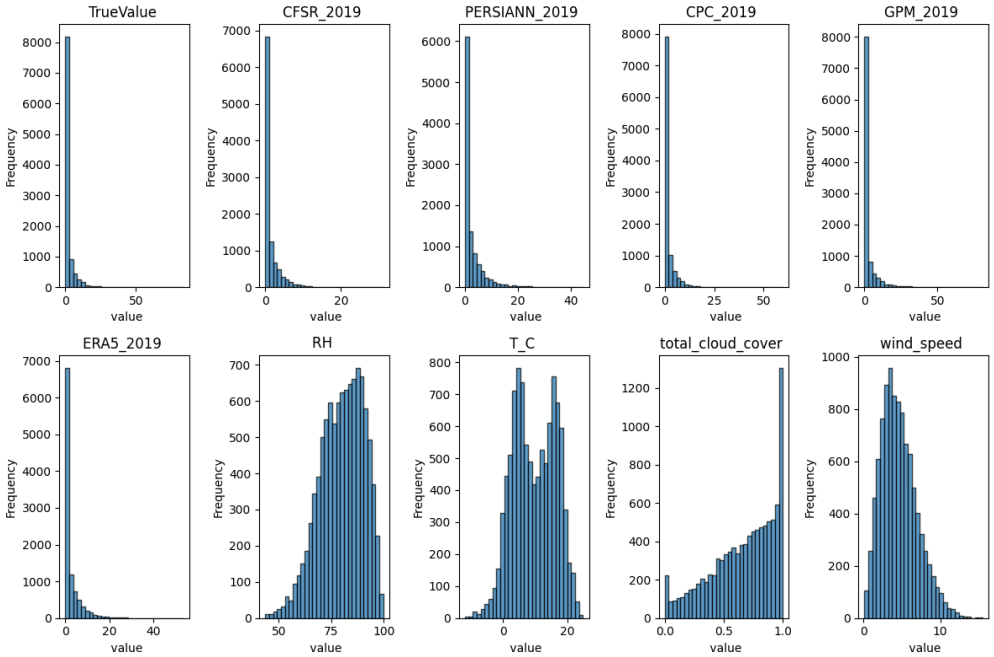

Figure 12: Histogram Unscaled

Now, a standard scaler was applied to all the graphs. This could be seen in Figure 13 below. Here, it was clear that the standard scaler was sufficient for the RH, T_C and wind _speed. The total cloud cover had also achieved a decent shape and did not need further scaling. However, for the precipitation data, further manipulation was required.

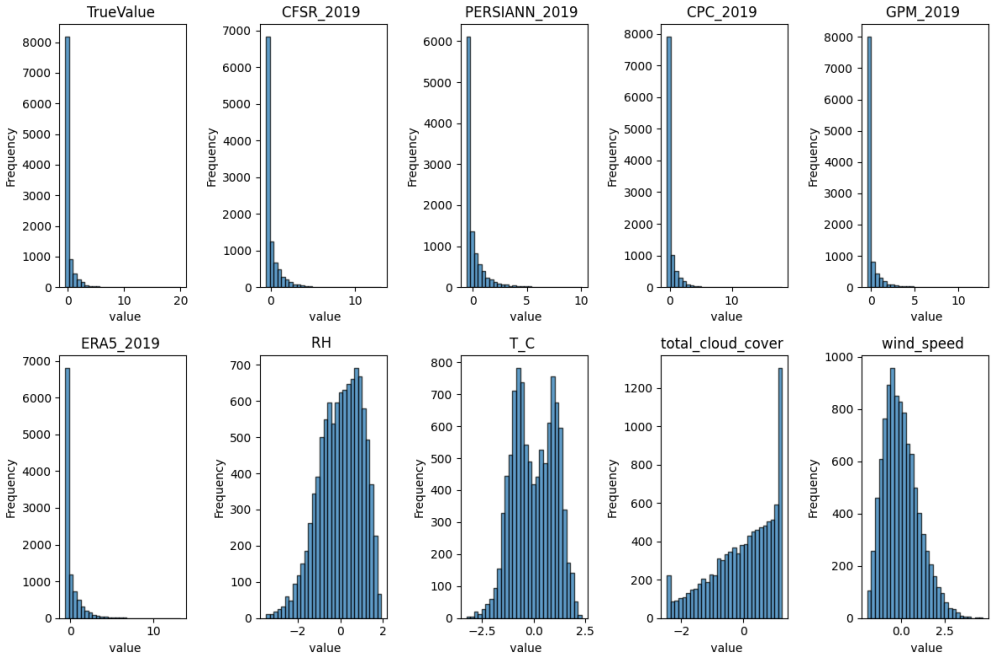

Figure 13: Histogram With Standard Scaler

For the final scaling, which was used for the model, an arcsinh scaler was applied before the standard scaler. Giving the result found below. The values were still skewed, but this is due to the large number of zeros. This was taken into account with the design of the engineered features.

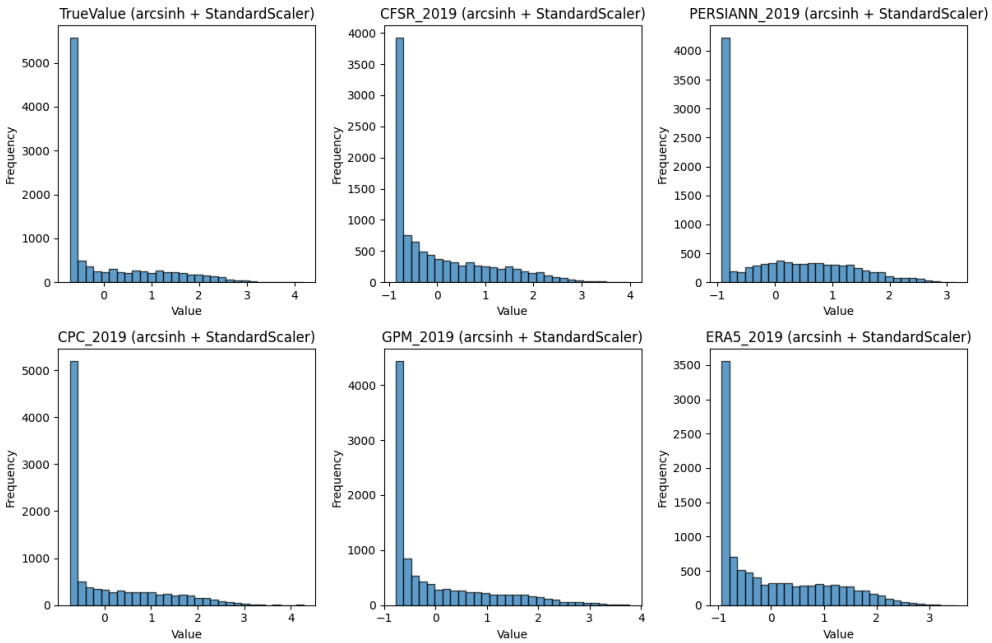

Figure 14: Histogram With Arcsinh And Standard Scaler

In order to remain consistent with scaling, all engineered features related to satellite precipitation data will be arcsinh and standard-scaled. The features related to humidity, temperature, cloud cover and wind speed will just be scaled with a standard scaler. This ensures that the engineered features don't get skewed due to scaling.

4.1.2 Spatial Model

The spatial model had the following architecture and made use of a "weighted MSE" as a loss function, see Table 2.

Component	Description
Input layer	$R^{\text{input\_dim}}$
Dense (main path)	128 units, ReLU, L2(10^{-3})
BatchNorm	Main path
Dropout	0.2
Dense	128 units, ReLU, L2(10^{-3})
BatchNorm	Main path
Dropout	0.1
Dense	64 units, ReLU, L2(10^{-3})
Linear shortcut	Dense(1) applied directly to input
Output (main)	Dense(1)
Residual connection	Add(main output, shortcut)

Table 2: Architecture of the residual neural network model.

The engineered features used for the Spatial ANN model can be seen in Table 3.

Category	Features
Lag features	lag1, lag2, lag3
Diff features	diff1, diff2, diff3
Wet/dry	wet1, wet2, wet6
Rolling	wet probability7, intensity3
Seasonality	Month, Day

Table 3: Engineered features used in the model.

Results from the spatial model applied to Hästveda for 5 and a half years can be found in Table 4. The model retained a strong IOA value while having an R^2 larger than 0.5, which indicated a good fit. The RMSE, in combination with MAE, suggest the largest error was attained for days with heavy rain. This was expected due to the training data not having such frequent high rain days. The MBE showed that the model underestimated precipitation, which could be seen in the graph as well as in the scatter plot.

RMSE	MAE	MBE	R^2	IOA
3.5683	1.5800	-0.6581	0.5479	0.8291

Table 4: Spatial Metrics

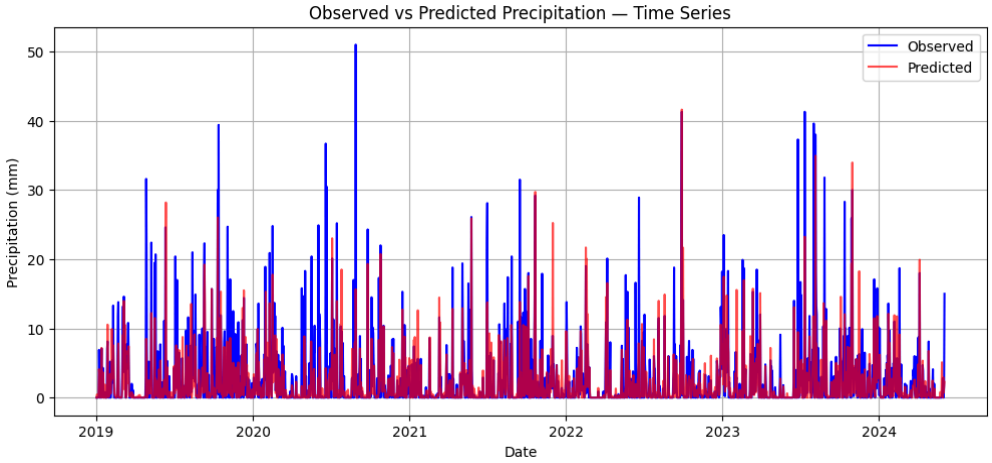

Figure 15: Spatial Forecast Hästveda.

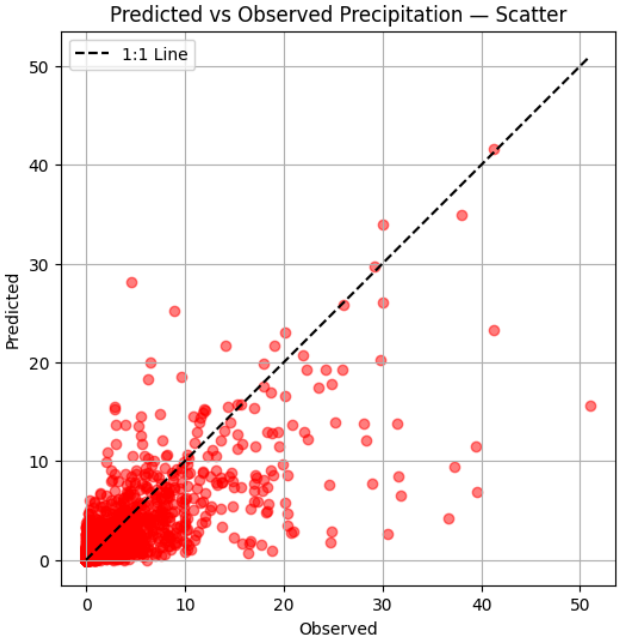

Figure 16: Scatter Hästveda

4.1.3 Prediction Models

The prediction model for Malmö retained similar values as the spatial model did for Hästveda, but with a higher R^2 value, see Table 5 and Figure 17. Notable here was the MBE value being -0.28, which showed that the model had a small negative bias. This could be seen in the scatter plot. From the prediction graph, it was quite clear that the error was dominated by the peaks, see Figure 18.

RMSE	MAE	MBE	R^2	IOA
3.5021	1.5567	-0.2212	0.5711	0.8584

Table 5: Malmö metrics

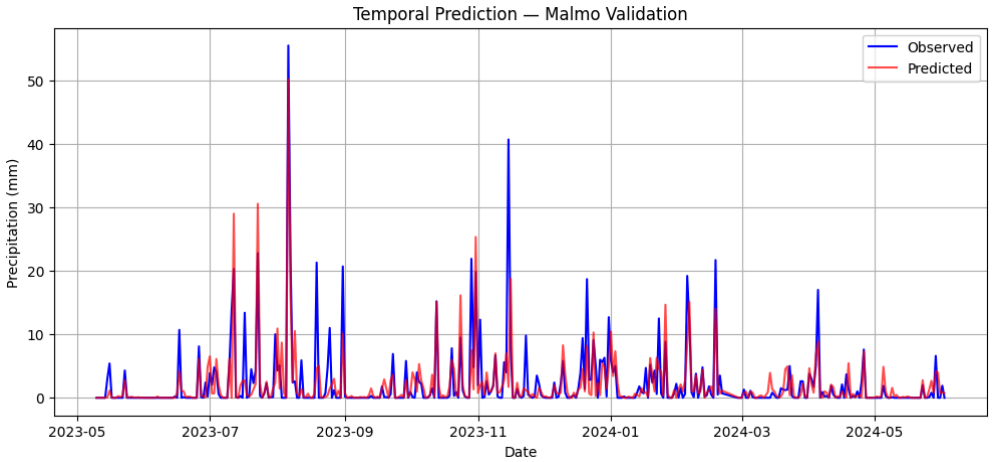

Figure 17: Temporal Prediction Malmö

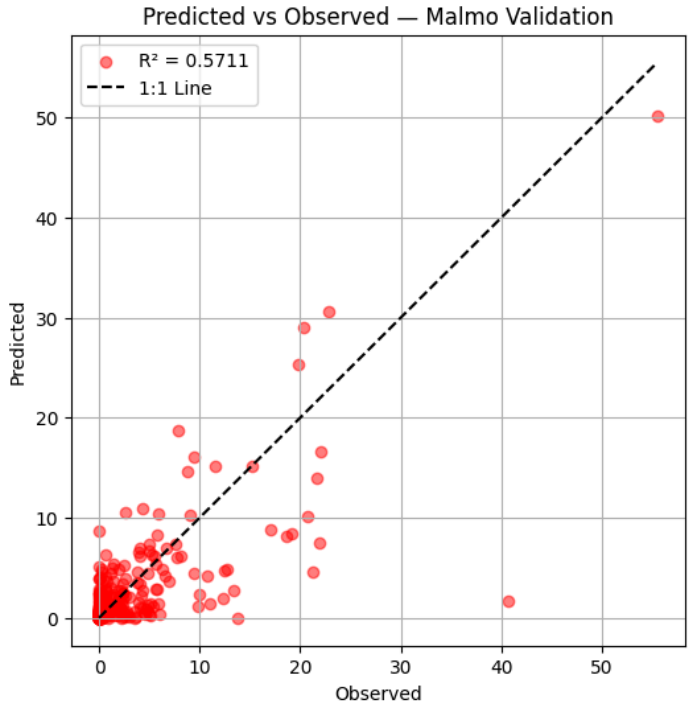

Figure 18: Scatter Malmö

The Hörby model performed relatively the same as the Malmö model, with a similar R^2 and IOA, see Table 6 and Figures 19 and 20. Its RMSE was lower than Malmö's, showing that the model had fewer large errors. The MBE was larger than Malmö, meaning that the Hörby model underestimated more. The Hörby model had a larger MAE than Malmö, which was interesting since the RMSE was lower. This pointed to the Hörby model being better at capturing peaks but falling short on normal days and less intense downfalls.

RMSE	MAE	MBE	R^2	IOA
3.1321	1.6527	-0.3665	0.5791	0.8488

Table 6: Hörby metrics

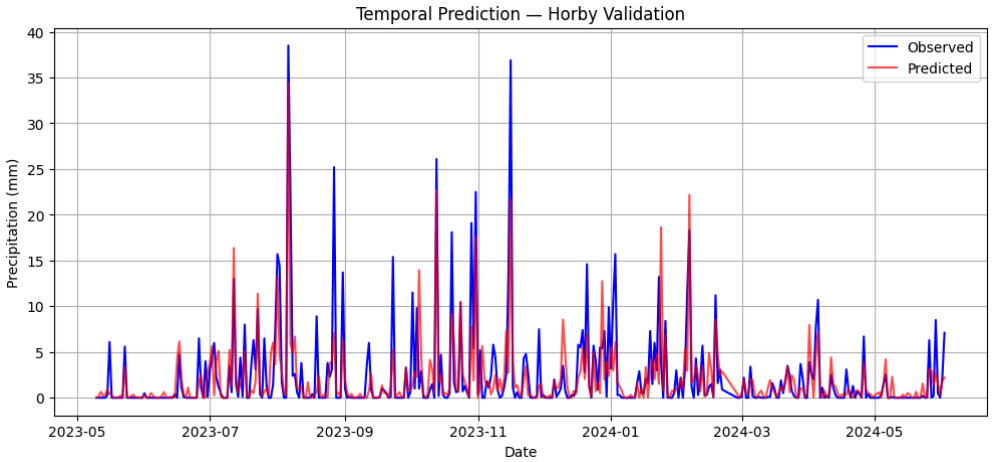

Figure 19: Temporal Prediction Hörby

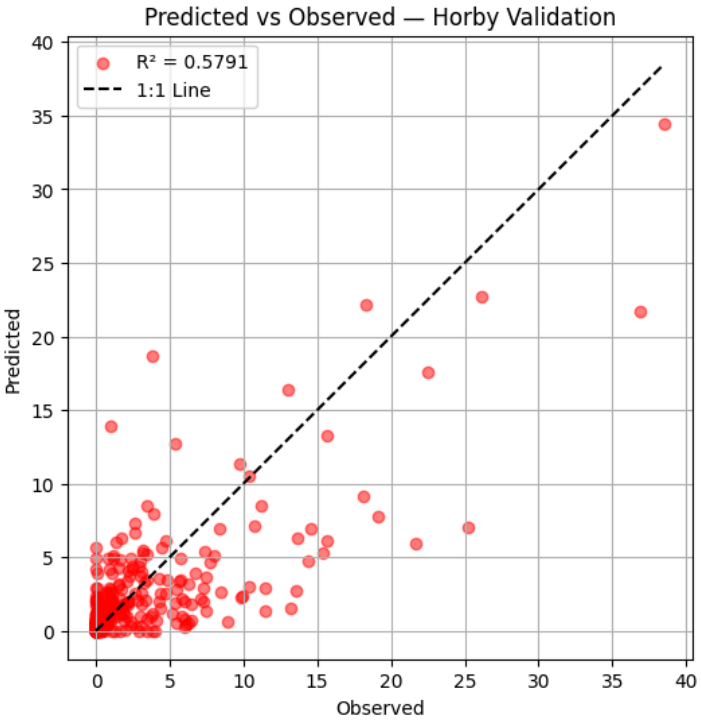

Figure 20: Scatter Hörby

Helsingborg achieved similar results as the previous models, but with a slightly lower R^2 and a higher MBE, see Table 7 and Figures 21 and 22. From the scatter plot, it was hard to see that the MBE would have increased in comparison to the other scatter plots. Numerous underestimations could be found in the region of prediction 0-10 mm, and observed 5-15 mm.

RMSE	MAE	MBE	R^2	IOA
3.2105	1.4868	-0.4594	0.5468	0.8424

Table 7: Helsingborg metrics

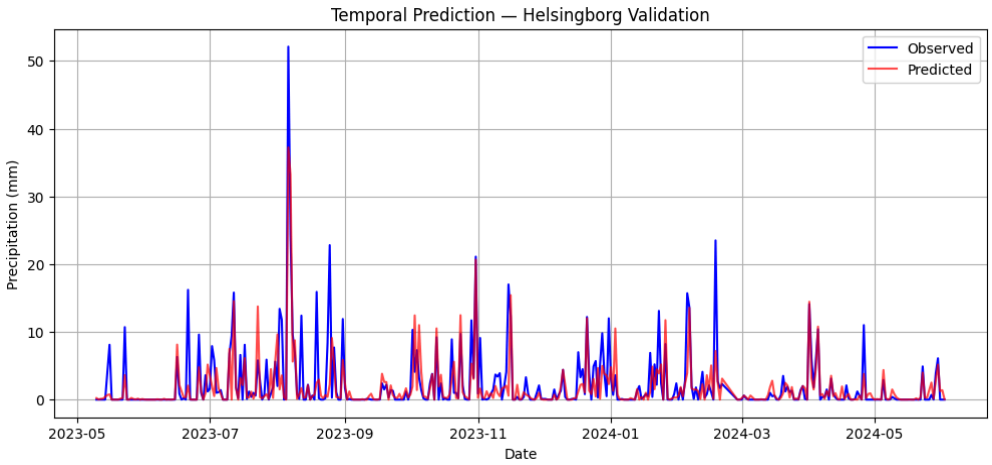

Figure 21: Temporal Prediction Helsingborg

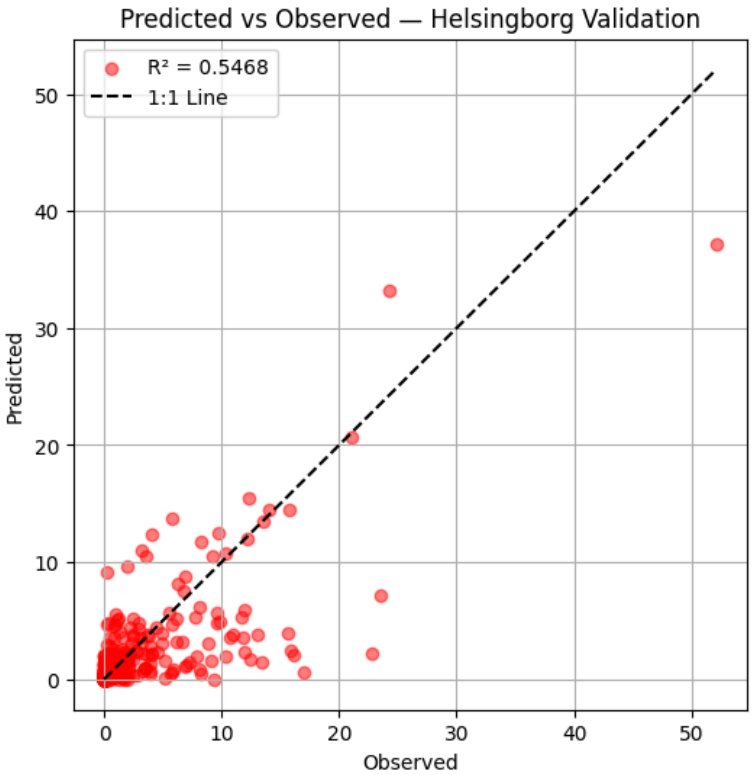

Figure 22: Scatter Helsingborg

The Skillinge model had a drop in IOA, resulting in it going below 0.8. The R^2 value dropped, but remained above 0.5. It also had a significant rise in MBE to roughly -0.6. This could be seen in the scatter plot. Notably, it achieved the lowest RMSE value of the models. This could be attributed to the validation data having peaks of lower magnitude than the other models, since the MAE value was similar to the other models, see Table 8 and Figures 23 and 24.

RMSE	MAE	MBE	R^2	IOA
3.0231	1.4357	-0.5934	0.5071	0.7948

Table 8: Skillinge metrics

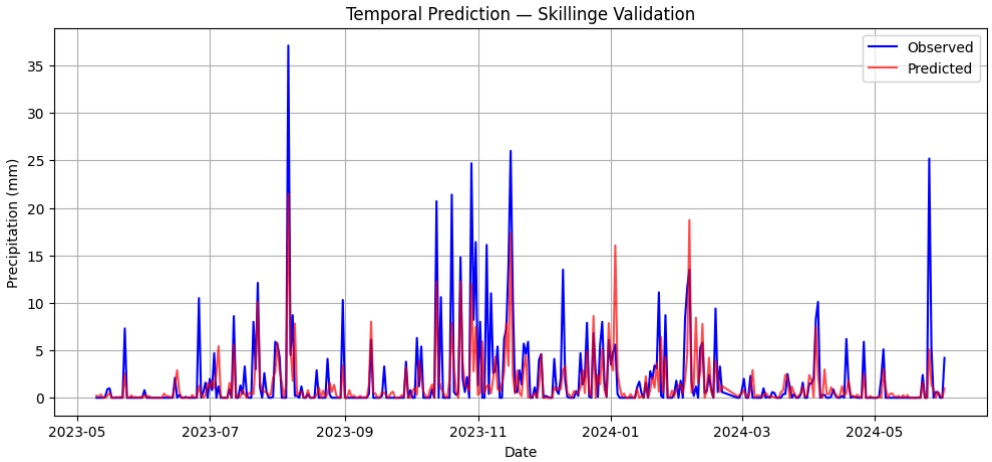

Figure 23: Temporal Prediction Skillinge

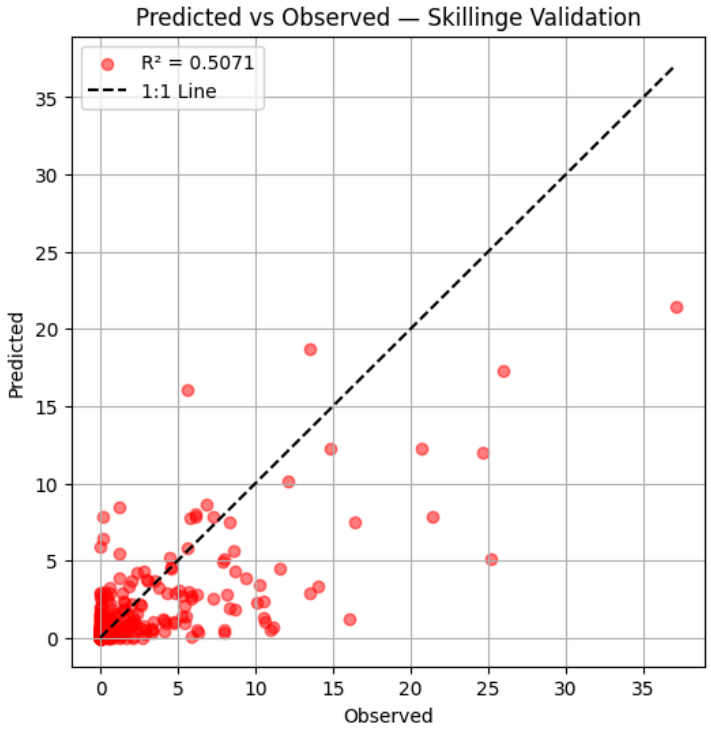

Figure 24: Scatter Skillinge

The Hallands model performed the weakest of them all, with an IOA < 0.75 and R^2 < 0.45. These poor results were not too surprising given the station's location in combination with the provided satellite data. Since the station's location was on an island, the precipitation tended to be more volatile. This, in combination with the satellite data having poor quality in areas close to the sea. In terms of RMSE and MAE, the models still performed relatively similarly to the other models. The scatter plot shows a clear trend of generally good estimations until the observed value reached 10 mm and

above. In this area, the model clearly under-predicts and wasn't able to capture these values, see Table 9 and Figures 25 and 26.

RMSE	MAE	MBE	R^2	IOA
3.7558	1.7946	-0.6435	0.4169	0.7229

Table 9: Hallands metrics

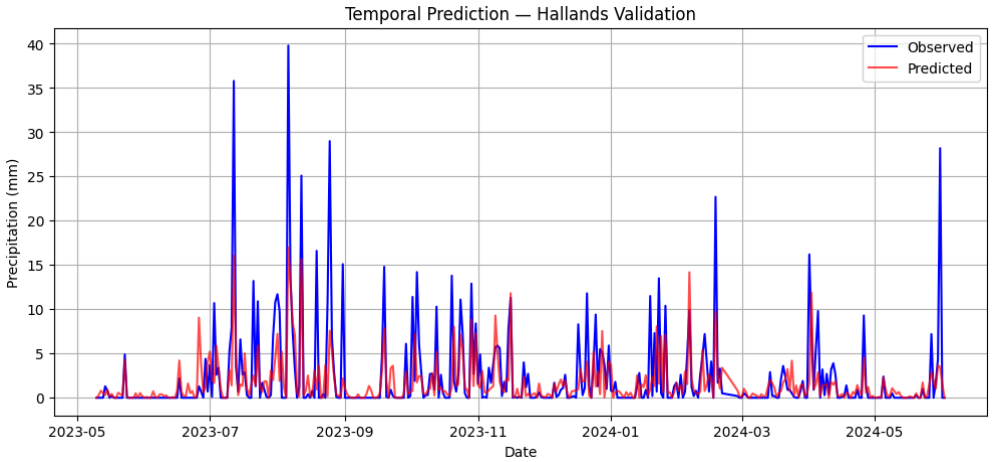

Figure 25: Temporal Prediction Hallands

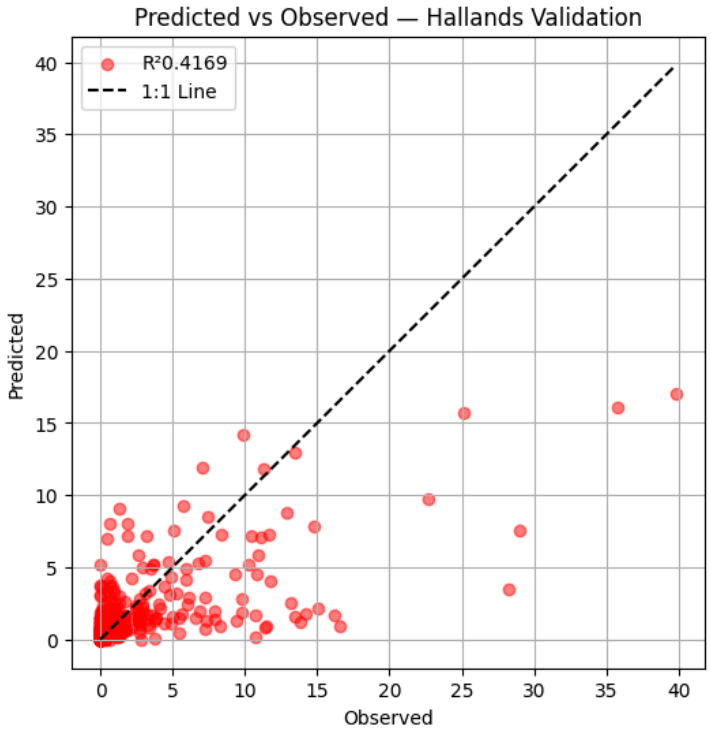

Figure 26: Scatter Hallands

4.2 Leakages

In this section, all results regarding the leakage detection are presented. Explanation and analysis of the results could be seen in the next part, the discussion. All results presented here were from the pipeline solution with RF followed by ANN. The single ANN solution was excluded.

4.2.1 Preprocessing

As illustrated in Figure 27, these were the original features before scaling it. The way that flow in and flow out are distributed made them hard to scale. Either way, all features 42 were standardized and this could be seen in Figure 28. The graphs had similar shapes, but the x-values had changed.

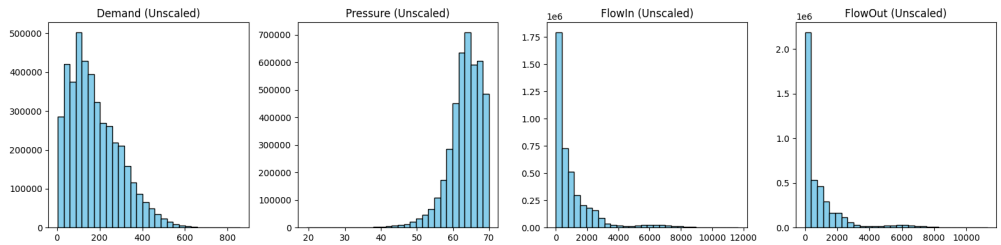

Figure 27: Histogram Unscaled

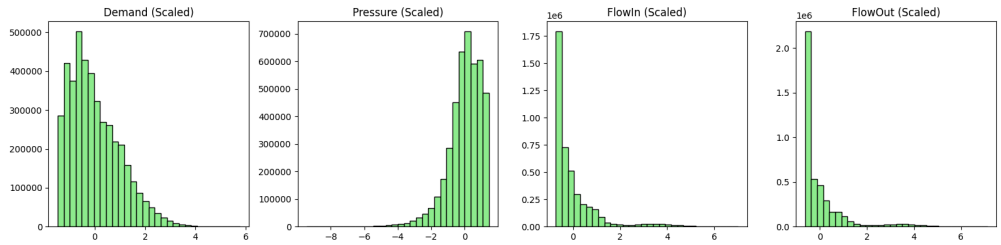

Figure 28: Histogram Scaled

As mentioned in the method section, feature elimination was also done. In Figure 29, accuracy was plotted against how many feature groups were included. Both two and seven feature scores are high, and to not risk sacrificing robustness, the one with seven feature groups was chosen. The selected features could be seen in Table 10.

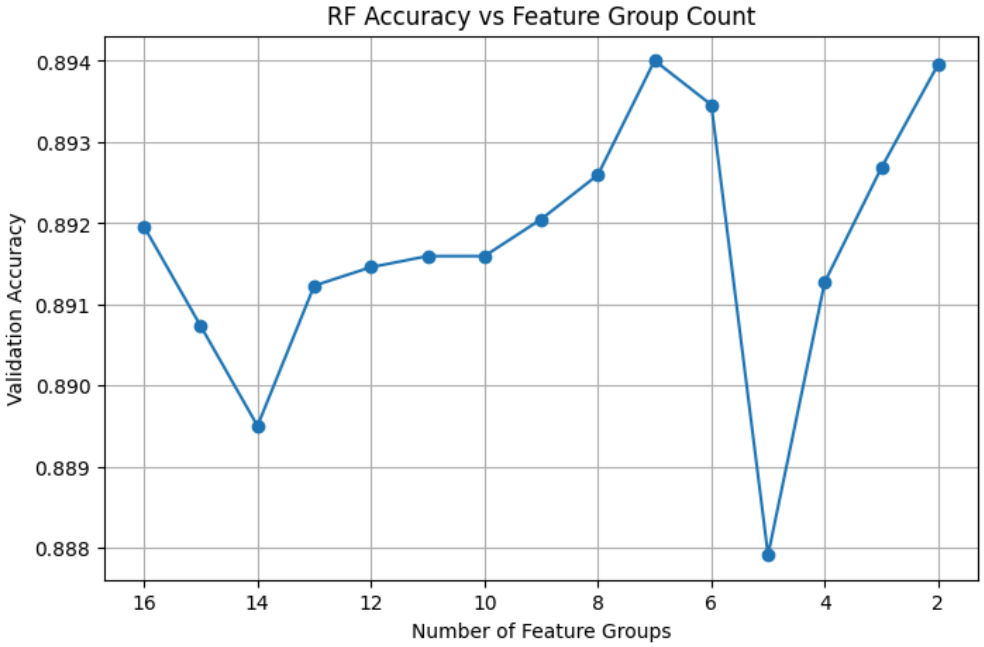

Figure 29: Accuracy depending on features

Feature Groups
Pressure × Flow In
Pressure × Flow Out
Rolling (Flow Out)
Rolling (Pressure × Demand)
Rolling (Pressure × Flow In)
Rolling (Pressure × Flow In × Flow Out)
Rolling (Pressure × Flow Out)

Table 10: Seven feature groups used.

4.2.2 Random Forest Model

After feature selection and grid search, the final result of the RF could be seen in Figure 30. The performance metrics for this model could be seen in Table 11. The model achieved an accuracy of 88% and an F1 score a bit lower. In this result, the threshold for the RF was set to 0.6. The increase in the threshold resulted in a decrease in FP. For the final RF model, 100 estimators were chosen

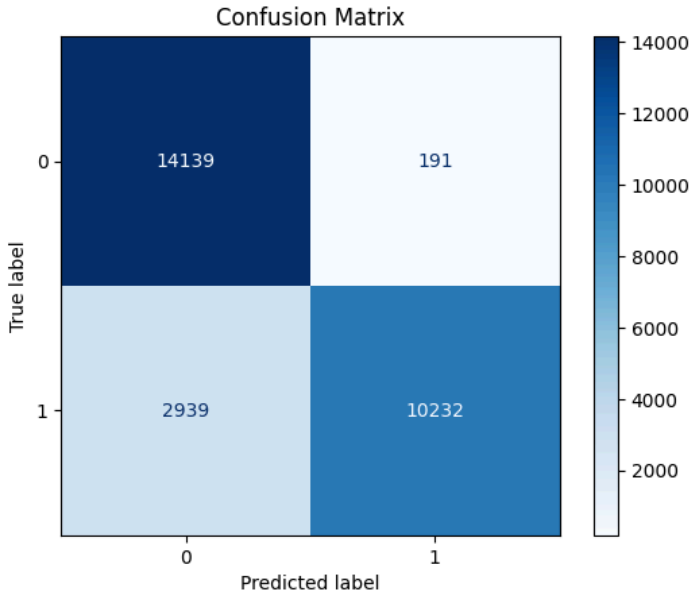

Figure 30: Confusion matrix for random forest

Accuracy	Precision	Recall	F1 score
0.8862	0.9817	0.7769	0.8673

Table 11: Performance metrics for the RF model

4.2.3 Artificial Neural Network Model

After the grid searches, the architecture of the ANN ended up as three hidden layers, 64-128-64. Since the ANN did not have a binary output, a normal confusion matrix could not be used. Instead, the result was displayed in the

heatmap that could be seen in Figure 31. The purple boxes in the diagonal were nodes that did not have any leaks in the used scenarios. The performance metrics for the ANN can be found in Table 12.

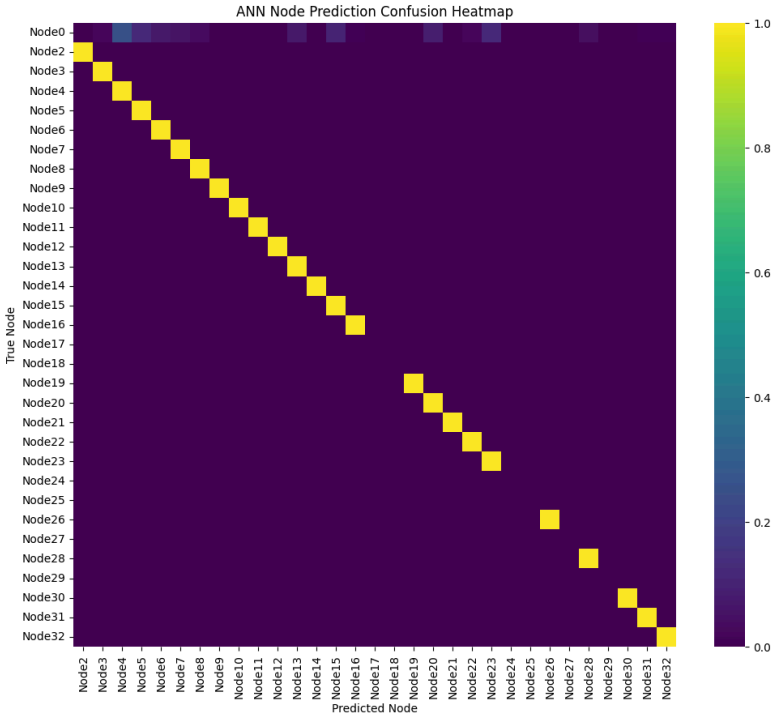

Figure 31: Heatmap for ANN model

Accuracy	Precision	Recall	F1 score
0.9815	0.9823	0.9815	0.9790

Table 12: Performance metrics for the ann model

4.2.4 Pipeline

The performance of each model individually did not show the full result, the metrics for the pipeline could be seen in Table 13. Same as in the case of the RF, these metrics can be somewhat rebalanced by adjusting thresholds. The threshold's implication on the result was evaluated in the discussion.

Accuracy	Precision	Recall	F1 score
0.8862	0.9409	0.7734	0.8410

Table 13: Metrics for the full pipeline

In addition to the full result of the pipeline, the best and worst recalls were shown in Tables 14 and 15. All the worst scenarios were incipient leaks, and all the best ones were abrupt leaks. Also worth noticing is that, except for scenario 51, every single leak had been partly discovered and located.

Scenario	Correct predicted	Number of leaks	Recall
51	0	5	0.00%
2	27	175	15.43%
66	53	184	28.80%
49	58	196	29.59%
78	113	374	30.21%

Table 14: Recall the worst 5 scenarios

Scenario	Correct predicted	Number of leaks	Recall
68	215	215	100.00%
76	218	218	100.00%
71	205	205	100.00%
84	36	36	100.00%
86	204	204	100.00%

Table 15: Recall the best 5 scenarios

5 Discussion

In the following section, a discussion regarding the results is conducted, and the performances of the models are evaluated. The section is divided into precipitation, leaks and development. In the development, a discussion of real and artificial data is conducted. Where advantages/disadvantages are analysed and their effect on the model result.

5.1 Precipitation

For precipitation, the discussion is split for spatial and temporal prediction due to the methodology and analysis of the result varying depending on the objective. Common to all is that a comparison against the satellite data is conducted since it is key to evaluating their respective performance.

5.1.1 Spatial

The Spatial model was trained on the following stations: Malmö, Skillinge, Hörby, Helsingborg and Hallands. These stations have a geographical spread covering most of Skåne, except for the northeast region, where the final test is conducted on the station Hästveda. This means that the quality of satellite data for each station plays a vital role in the test performance on Hästveda. In the graph below, the satellite data is plotted against the true value for Hästveda. In order to make the graph clearer, only the satellite data with the lowest RMSE is plotted. The other satellite products, RMSE and MAE, can be found in Table 16.

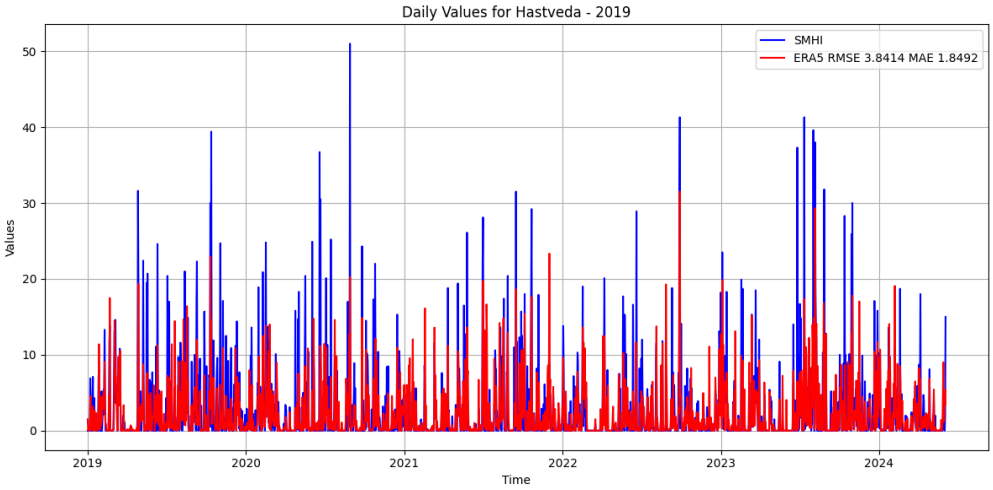

Figure 32: Hästveda satellite data

	CFSR	GPM	CDR	Era5	CPC
RMSE	4.4988	8.4890	5.4084	3.8414	4.5640
MAE	2.0053	3.4059	2.7633	1.8492	2.1050
R^2	0.2814	-1.5585	-0.0385	0.4761	0.2605

Table 16: Hästveda RMSE, MAE and R^2 value for Satellite Products

The model had an RMSE of 3.5955, which is lower than all the satellite products, which is an indication of good performance for the model. At this location, Era5 performed the best of the satellite products with an RMSE of 3.8414. Important to note is that the GPM had a very poor performance at this station while having a good performance at the training stations. This leads to the spatial model having worse performance due to its reliance on the quality of data from the satellite products. For our model, this has an even

larger effect since the GPM data is the most important feature for the model. A drop in performance from validation to test was expected, but this was magnified due to the GPM data. However, the model still managed to outperform the other products and retain a high IOA. This indicates the model is able to be used for areas without stations and provides a more accurate result than these satellite products. The model, however, is only trained and tested on a small region, and its performance on regions further away hasn't been tested or evaluated. It can be expected to have a worse performance in regions further away since it can have climate variations not seen by the model.

The model achieved an R^2 value of 0.5479, which, in statistical terms, is a decent or adequate result. However, precipitation is inherently noisy and known to be difficult to predict. This suggests that a result in the range of 0.5-0.6 is considered good within precipitation. A substantial drop in validation is expected due to the data's inherent noisiness, as shown in previous findings (Portuguez-Maurtua et al., 2022). Furthermore, guidance on interpreting the R^2 value has been provided in the literature (Kutner et al., 2005). Finding that the coefficient of determination must be interpreted in the context of the application domain, which, in this case, is inherently a noisy process. This means the model may yield a low R^2 value while still being an adequate model.

The spatial model showed improvements in all metrics in comparison to the satellite products. It outperformed most products greatly, except for the Era5, where it only marginally outperformed it. This proves that the model is able to increase the accuracy of the satellite data. However, further testing in the region is needed before applying it to a forecast model, but provides a good indication of the possibility of improving the precipitation data in unmonitored regions. Ultimately, the model achieved its goal in providing more accurate data and set the foundation to be used for the development of a forecast model.

5.1.2 Temporal prediction

For temporal prediction, each model was evaluated individually against the corresponding satellite data for that station. Starting off with the model for Malmö, the satellite data was plotted against the true station data. The same process as in spatial was applied here, where only the best-performing satellite product was plotted. In this case, it was GPM. The other satellite products RMSE, MAE and R^2 can be found in Table 17 after the graph.

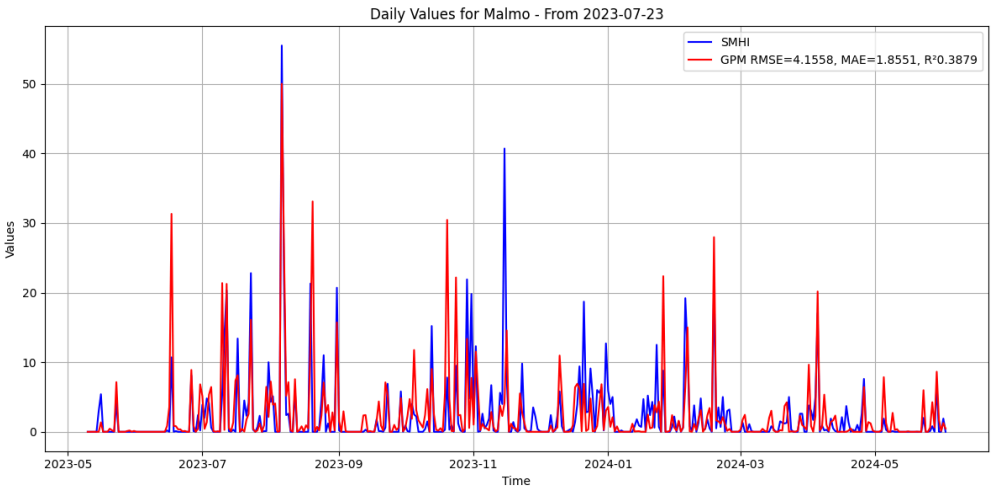

Figure 33: Malmö GPM satellite data

	CFSR	GPM	CDR	Era5	CPC
RMSE	4.3917	4.1558	6.5270	4.3955	4.6246
MAE	1.8713	1.8551	3.3888	1.9658	1.8415
R^2	0.3164	0.3879	-0.5099	0.3152	0.2420

Table 17: Malmö RMSE, MAE and R^2 value for Satellite Products

For this location, the satellite data was more stable than in Hästveda. Even if the CDR values are higher than the other, this is relatively smaller than in Hästveda. The model managed to outperform all the satellite products with a significantly lower RMSE value of 3.5021, while retaining an R^2 above 0.55 and an IOA of 0.85. This shows that the model captures roughly 50 percent of the variance of the true value and has a good agreement with the magnitude and pattern of the true value. In summary, the model increased the R^2 value by 47% while decreasing the RMSE and MAE to 0.6548 (mm) and 0.2984 (mm), respectively.

Following Malmö is Hörby, which is the station that is the furthest inland. In the graph below, the best-performing satellite product in R^2 is plotted against the true value. For this location, the best-performing satellite was ERA5 with an R^2 of 0.5250. A table containing all the satellites' RMSE, MAE and R^2 values is provided after the graph below.

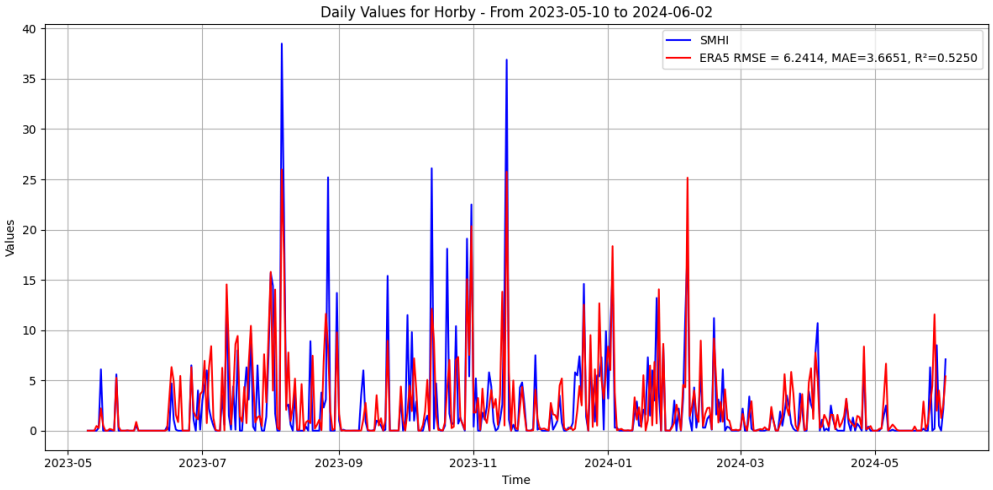

Figure 34: Hörby ERA5 satellite data

	CFSR	GPM	CDR	ERA5	CPC
RMSE	3.6971	6.4323	6.4957	6.2414	5.9150
MAE	1.8506	3.4494	3.8863	3.6651	3.2162
R^2	0.4087	0.3301	-0.6804	0.5250	0.3693

Table 18: Hörby RMSE, MAE and R^2 value for Satellite Products

The trained model showed a slight increase in R^2 of 10%, which is acceptable since the ERA5 captured the variance relatively well. Most notable here is the significant decrease in RMSE and MAE with 3.1093 (mm) and 2.0124 (mm), respectively. This is a significant improvement in accuracy. However, this improvement diminishes if it's compared to CFSR. The improvement is still present there, proving that the model is viable. It's interesting that the ERA5

is able to achieve a high R^2 value even with its large RMSE and MAE. This emphasises the importance of evaluating the model as well as data on several metrics in order to capture the whole picture.

The next station is Helsingborg. Here, the CFSR product performed the best in all categories. Its predictions are plotted below against the SMHI data. The other model's values can be found in Table 19 after the graph. Noticeable about the CFSR predominance is its inability to capture peaks, showing a relative negative bias.

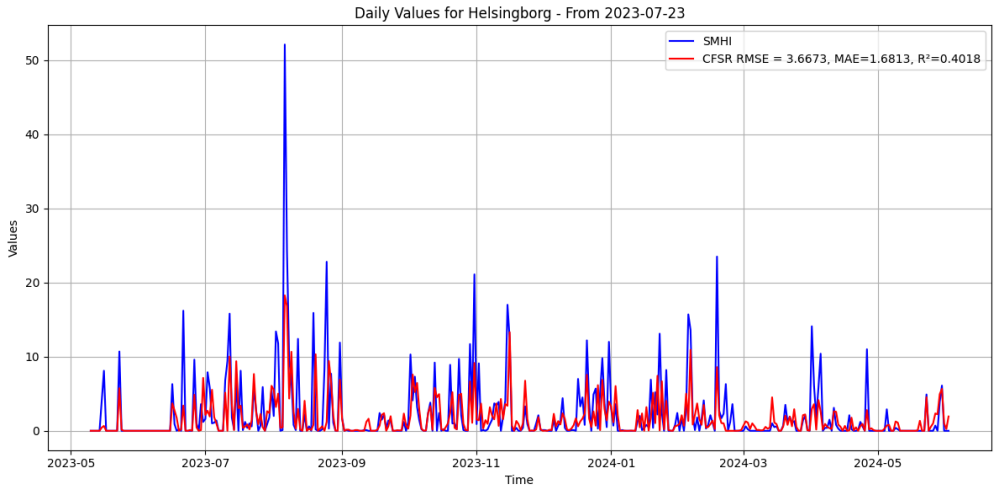

Figure 35: Helsingborg satellite data

	CFSR	GPM	CDR	Era5	CPC
RMSE	3.6673	3.9305	5.8044	3.7915	4.2287
MAE	1.6813	1.9556	3.2196	1.9801	1.7952
R^2	0.4018	0.3129	-0.4986	0.3606	0.2046

Table 19: Helsingborg RMSE, MAE and R^2 value for Satellite Products

For Helsingborg, the model showed the greatest improvements in R^2, with an increase of 36%, showing that the models capture the variance better than the satellites. The RMSE and MAE also decreased with the trained model, which is to be expected. The model retained a decrease of 0.4568 (mm) to RMSE and 0.1955 (mm) to MAE. From the satellite data, it's clear that the CDR had a poor performance at this location, which has an effect on the training of the model. For a potentially more accurate result, it would be worthwhile to remove the CDR data from the training and complement it with more data from the other satellites.

The Skillinge model improved in every metric. R^2 increases from 0.3889 to 0.5071, which is a 30% increase when compared to the ERA5, which performed the best of the satellite products. The RMSE saw a decrease of 0.3209 mm, which is decent when the satellite product already had a decent RMSE value. MAE saw a similar decrease of 0.2754 mm. The CDR product continues to give poor estimations with large RMSE and often a negative R^2 value.

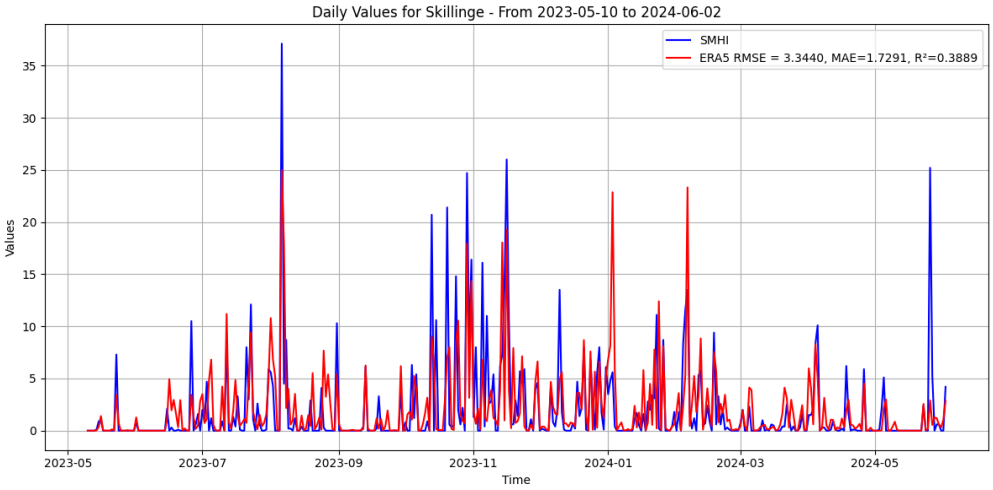

Figure 36: Skillinge satellite data.

	CFSR	GPM	CDR	Era5	CPC
RMSE	3.4436	3.9884	5.4871	3.3440	6.1117
MAE	1.6303	1.8806	2.9584	1.7291	2.0975
R^2	0.3519	0.1306	-0.6454	0.3889	-1.0413

Table 20: Skillinge RMSE, MAE and R^2 value for Satellite Products

For the Hallands model, we saw a generally worse performance in comparison to the other models. The results become even clearer when examining the training data. Firstly, the satellite product had an RMSE average of 6.86 mm, which is significantly larger when comparing the data from the other stations. Furthermore, the MAE also averages around 3.8 mm.

The R^2 values are also close to zero or negative. This means that a prediction of the average could outperform some of the models. When examining the full timeline of data for Hallands in Figure 38, it becomes even clearer why the model performed poorly. The period before the validation period is fairly stable, but following 2023-07 and forward, the precipitation becomes more volatile. For the model that has only been trained on prior data, this pattern is completely unseen and is a major challenge for the model to predict. In order to combat this, the validation period was moved. This didn't increase the results since the data wasn't sufficient. This means that the volatile period was too large and had a significant impact on the overall performance.

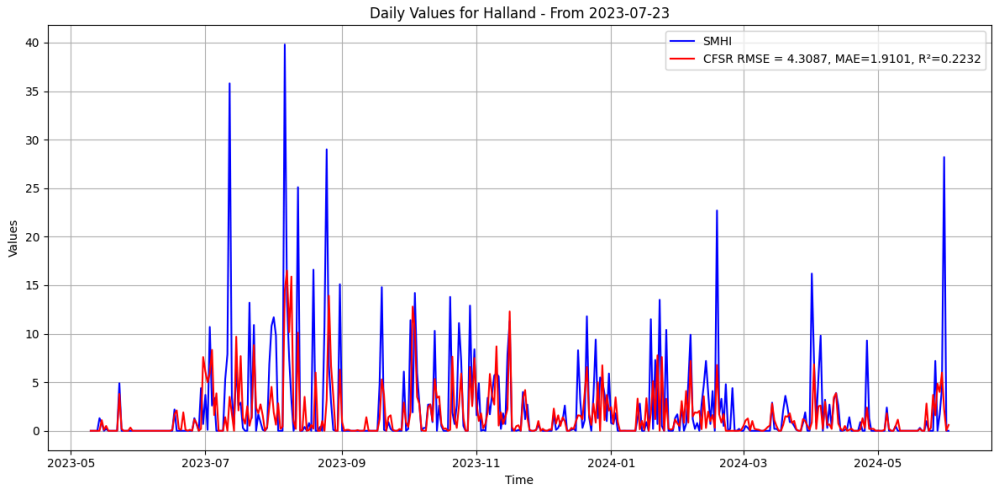

Figure 37: Hallands satellite data

	CFSR	GPM	CDR	Era5	CPC
RMSE	4.3087	8.8703	7.0971	7.2776	7.0830
MAE	1.9101	4.7569	4.2997	4.4447	3.8656
R^2	0.2232	-0.4489	-0.4384	0.1963	0.1182

Table 21: Hallands RMSE, MAE and R^2 value for Satellite Products

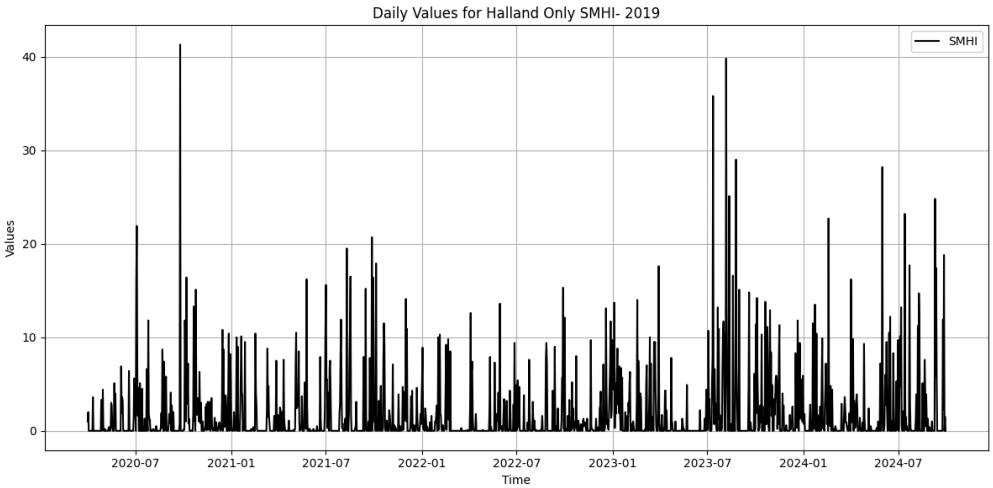

Figure 38: SMHI data for Hallands

In summary, the temporal prediction models were able to outperform the satellite products. However, they weren't accurate enough to replace active stations. In order for it to actively replace a manned station, it needs a higher accuracy. However, if the model can be properly trained on an area lacking a station, it would show potential to be used when manpower is scarce or a cheaper option is desirable. The same reasoning for the R^2 values applies here

as in the spatial model. It is also possible to further increase the accuracy of the model. This can be done through more accurate satellite products or other features that can help the model predict precipitation. Another option is to consider another AI model, for example, an LSTM, which makes use of an RNN. The LSTM can selectively store values as well as better understand context over long sequences. The model also proves it can be used for future development of a forecast model. In contrast to the spatial model, which is more general, the prediction models can be developed for area-specific use if desired. Area-specific models tend to outperform general models. However, a fair comparison of this can't be made from the results. If area-specific models are used to train a general forecast model, more computing power is needed since each area consists of a model rather than one general model.

5.2 Leaks

5.2.1 Single ANN Solution

The ANN-only solution was excluded from the result part due to the low scores that were achieved. After training and grid search, the highest accuracy achieved was around 55%. This was achieved by the model only predicting no leak for every instance. The reason this worked so much worse than the pipeline is probably the way the labels are constructed. For the pipeline, the RF has almost 50/50 in zeroes and ones, and the ANN has the probability somewhat distributed over the 31 nodes. For the ANN, the model instead receives just above 50% zeroes and then the rest of the probability is distributed over the 31 nodes.

5.2.2 Feature Elimination

As mentioned in 3.2.1, a lot of features were involved in the development of the pipeline. The groups that were finally used in the models could be seen in Table 10. As seen in Figure 20, using either two or seven feature groups scored the highest accuracy for the RF. However, using only two feature groups could reduce the robustness of the model. The ANN used the same features as the RF did, but the metrics for the ANN were not as affected by the feature selection.

It seems that rolling, for the most part, had a higher importance than the non-rolling features. Since a large majority of leaks last for some period of time, looking at a leak often means the node leaked before that. None of the original features made it to the final model, apart from demand. All of them are present without rolling in at least one feature group. Notably, four of the feature groups used are rolling with pressure multiplication with some other features. Pressure is most likely the feature most linked to detecting the leaks in this WDN.

5.2.3 Random Forest

Firstly, the result for the RF model, the final metrics, is shown in Table 11. The accuracy almost hit 90%, which is good but not great. Since the data in the leakage detection is simulated, it is noise-free and has no other measurement errors. It should therefore theoretically be possible to reach 100%. Why it does not reach that high could be coupled with a few different things. Not all nodes had leaks in the scenarios used, but node 25 did, and in Figure 31, none of those has reached the ANN. The reason node 25 has not made it through to the ANN is most likely due to the smaller sample sizes, where node 25 has a leak. In the method, it is mentioned that each leak should have 1000 indices prior to and succeeding the start of the leak. If the leak happens to only last a few indices or if the leak appears just before the end of the scenario, there will not be as many indices for this node. With this

in mind, maybe it would have been better to take more scenarios and make sure that there are just as many leaks in every node.

The F1-score of the RF reached 86.7% and could be increased slightly by altering the threshold. In this case, the focus was more shifted to increasing the precision. This is, as mentioned, up to the user and dependent on the task.

5.2.4 ANN

The first thing to look at when it comes to the ANN is the heatmap in Figure 31. In a perfect model, the entire diagonal should be yellow and all other boxes purple. The purple boxes in the diagonal mean, in this case, that the data that reached the ANN did not have any leaks in these nodes. If there were leaks but the ANN failed to place them, it should look more like it does for row 1, with lighter-colored boxes spread out. The reason that row 1 looks worse than the rest is that the false positives are summed up in this row. The ANN has not seen any inputs with no leak during training and has no correct way to handle the false positives.

The results for the ANN are not 100% representative for two different reasons. The first thing is something that potentially could decrease all the metrics in Table 12. As seen in Figure 30, the RF misses some of the positives. It could be that these positives are a lot harder to predict, and since the ANN never sees these data points, they only negatively impact the RF. The second thing that would result in a better ANN score is the false positives. The ANN is exclusively trained on indices where the leak is present. This means that all false positives from the RF will affect the score of the ANN.

So the metrics in Table 12 are not wrong, just not showing exactly how well the ANN can actually locate the leaks. Still, the metrics that the ANN shows

are somewhat reasonable. As mentioned in the RF part, it should be possible to score 100% on simulated data. The ANN is not really all the way there, but close enough considering the false positives from the RF. The ANN is trained to always point out a leak, and since it does not have the possibility to correct the RF, the false positives will affect all four metrics.

5.2.5 Pipeline Solution

For the final product, the pipeline. The metrics of the pipeline could be seen in Table 13. The first thing that can be seen in the table is that the recall score is very low. Recall being at 77.3% means that almost 23% of the indices with leaks are being missed, some by the ANN, but this is mostly the result of the RF. This might seem like an underdeveloped model. What is important to note are Tables 14 and 15, and also the 94% precision of the pipeline. Now, what does this mean for the result? The worst recall scenario 51 is at 0%. This is because scenario 51 is the scenario with node 25 leaking. There is not enough data for node 25 to be a valid prediction for the model. Without scenario 51, the leaks in all scenarios are found, even if not every index is predicted. This, in combination with the high precision, suggests that if a leak is predicted, it is very likely to be a true leak.

There is one more thing that is interesting about the worst compared to the best recalls. All five of the worst recalls are scenarios with incipient leaks, and all five of the best recalls were abrupt leaks. Incipient leaks start off small, and it is possible that the pipeline has a hard time locating the leaks until they reach a certain size, and that could be the reason for the lower scores. This might also be why scenario 51 is not found, not only because of such few samples, but because the leak never reached its peak.

So how useful can a model that misses 23% of the leaks be? Well, considering the heatmap in Figure 31, looking at the row for node 0, meaning no leak, the ANN model does not predict the leak in the same node every

instance. This means that the model in combination with some counter could probably sort out all false positive leaks. If the counter looks at the last 100 data points and, hypothetically, there have been 20 predicted leaks in different nodes, they are not considered real leaks. If the counter for the last 100 data points reaches more than 20 for a specific node, then it is considered a real leak. Using the model in this way could increase its usefulness and also means that the issue with false positives is decreased.

Focusing on reducing false positives could be crucial for some tasks. Consider a company that has to send a maintenance crew to every predicted leak and go through the WDN node, which costs both time and money. Not finding a leak can also be costly, and therefore, it is an important result that almost all the leaks were at least partly found. Finding leaks sooner will reduce water waste, which is very important due to the scarcity and cost of water around the globe.

5.3 General

In this thesis, two case studies were conducted within hydrology. Both use ANN as their solution. However, the two cases differ in their respective tasks as well as data. For the precipitation, it was a regression task with real data, while leak detection was a categorization task with artificial data. The type of data directly affected how the data exploration was conducted. Firstly, anomalies aren't present within artificial data, which reduces the amount of data alteration. Furthermore, noise isn't normally present in artificial data. Sometimes it is, but often it is Gaussian-generated noise. This results in the data being almost completely accurate, which isn't the case with real data. For the precipitation data, there were plenty of missing data points for specific days. That needed to be altered. From this, it is clear that artificial data is easier to work with and produces better results. However, since the

data is artificially generated, this limits its usage in the real world, and the real data achieves the opposite.

As mentioned earlier, this thesis mainly focused on the use of ANN. This doesn't mean that an ANN model is the optimal solution for these tasks. Potential other models could achieve the same or better results. The goal was rather to highlight areas where AI and machine learning had the potential to be implemented, as well as the benefits of implementing it.

6 Conclusion

In the following section, a short conclusion of the case studies was presented. The results will be reflected on, whether they were satisfactory or not.

6.1 Precipitation

In this case study, two types of AI models for precipitation have been investigated. One spatial model, trained to predict the precipitation at a given location in Skåne. The other models were prediction models, which were area-specific. These models had the goal to be able to predict similar values as the local station.

In conclusion, the precipitation models have varying results. The spatial model achieved good results and outperformed the satellites in its task. This proves that a spatial model can be used to achieve higher-quality data in unobserved areas. This lays a foundation for the implementation of a forecast model that uses a spatial model's output as a feature.

For the temporal prediction, the results were not fully satisfactory. In the case of replacing active stations, the model fell short in all locations. However, the prediction models were able to increase the accuracy of the data. This shows that the temporal prediction model is an option in areas lacking coverage from stations. This is a cheaper alternative to a local station and diminishes the manpower needed.

6.2 Leaks

In this work, it has been investigated if machine learning can be used to detect and locate leaks in WDN. The results from the simulated data are promising, with almost every leak detected and localised. This indicates that

machine learning-based solutions have the potential to improve the work that is done today and reduce the times customer complaints to detect leaks.

Water is one of the world's most vital resources. Improving the leak detection could greatly increase the efficiency of the water distribution and also decrease the water waste. This is an important work, and machine learning could be an important part of this.

6.3 General

This thesis has investigated two different approaches to integrating AI into hydrology. The results gathered demonstrate once again how AI can be applied across a wide range of fields. Considering how vital water is and the waste and scarcity that can be seen globally, further involving AI in hydrology is an important area of research. The thesis also highlights key differences in designing a model based on data type. Furthermore, how it's applicable to rather simple models.

7 Future Work

For the spatial model, there are improvements to be made before it can be used for its end goal. Firstly, more accurate satellite models should be looked into. These would enable the model for more precise results. Secondly, the model needs to be further tested on other areas within the region of Skåne. This is needed for further proof of concept. With these two steps improved, the model will be ready to use a data source for a forecast model.

In regard to the temporal prediction models, they clearly need further development if they are to replace real stations. The improvements needed follow the same plan as for the spatial model. Another option is to change the goal of the model. Rather than replacing stations, it could be developed to support the stations. In this version, the model will focus on estimating the values of missing dates. This allows the model to achieve higher accuracy through the usage of real values in the training process. The need for this was discovered through the data collection. The stations had a lot of periods with missing dates, with varying periods.

The leakage detection part of the thesis is only tested on simulated data. Future work should focus on real data and, hopefully, also real-time detection to determine how soon a leak can be detected. There is a challenge with labels on real data, due to the fact that not every node has leaked. A potential solution to this could be the use of spatial data. In the leakDB dataset, the lengths of the pipes vary from scenario to scenario. This is not the case in a real WDN, as the pipe length and location would be fixed. This also allows the model to be expanded to give more accurate leak location instead of being constrained to nodes. With this in mind, the result of this thesis still shows potential as a viable solution and can be used as a foundation for future work.

The models have been developed as planned, but due to some issues with the Application Programming Interface (API) provided by Kentima, they are not integrated into the software. This was one of the targets of the thesis, and this

will be addressed in a future step. However, simulating values in the software was successfully tested, so integrating the models is definitely a possibility. Further implementations beyond leakage detection are possible as well, such as predicting maintenance, water quality monitoring, and energy optimisation. Expanding the system to incorporate such functionalities could significantly enhance operational efficiency, reduce costs, and improve system reliability. Consequently, these future implementations would provide substantial benefits not only to Kentima and its clients but also to society at large by contributing to more sustainable and resilient water management systems.

References

Ashouri, H., Hsu, K., & Sorooshian, S. (2015). Persiann-cdr: Daily precipitation climate data record from multisatellite observations for hydrological and climate studies. Bulletin of the American Meteorological Society, 96 (1), 69–83. https://doi.org/10.1175/BAMS-D-13-00068.1

Bishop, C. M. (2006). Pattern recognition and machine learning. Springer.

Breiman, L. (2001). Random forests. Machine Learning, 45 (1), 5–32.

Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (1984). Classification and regression trees. Wadsworth.

Burbidge, J. B., Magee, L., & Robb, A. L. (1988). Alternative transformations to handle extreme values of the dependent variable. Journal of the American Statistical Association, 83 (401), 123–127.

Chen, M., Shi, W., Xie, P., Silva, V. B. S., Kousky, R. W. H., & Janowiak, J. E. (2008). Assessing objective techniques for gauge-based analyses of global daily precipitation. Journal of Geophysical Research, 113 (D04110). https://doi.org/10.1029/2007JD009132

D'Agostino, R. B., & Pearson, E. S. (1973). Tests for departure from normality. Biometrika, 60 (3), 613–622.

Doe, J., & Smith, J. (2025). Rainfall prediction features significance analysis [Accessed: 2026-01-05]. Journal of Informatics and Web Engineering. https://mmupress.com/index.php/jiwe/article/download/1473/840/15197

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

Gorelick, N., Hancher, M., Dixon, M., Ilyushchenko, S., Thau, R., & Moore,

D. (2017). Google earth engine: Planetary-scale geospatial analysis for everyone. https://doi.org/10.1016/j.rse.2017.06.031
Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection(Vol.3). http://www.jmlr.org/papers/volume3/guyon03a/guyon03a.pdf
He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 770–778. https://doi.org/10.1109/CVPR.2016.90 64
Hersbach, H., Bell, B., Berrisford, P., Hirahara, S., Horányi, A., Muñoz-Sabater, J., Nicolas, J., Peubey, C., Radu, R., Schepers, D., Simmons, A., Soci, C., Abdalla, S., Abellan, X., Balsamo, G., Bechtold, P., Biavati, G., Bidlot, J. R., Bonavita, M., . . .Thépaut, J.-N. (2020). The era5 global reanalysis. Quarterly Journal of the Royal Meteorological Society, 146 (730), 1999–2049. https://doi.org/10.1002/qj.3803
Hoerl, A. E., & Kennard, R. W. (1970). Ridge regression: Biased estimation for nonorthogonal problems. Technometrics, 12 (1), 55–67.
Huffman, G. J., & Tan, J. (2020). Integrated multi-satellite retrievals for the global precipitation measurement (gpm) mission (imerg). In V. Levizzani, C. Kidd, D. Kirschbaum, C. Kummerow, K. Nakamura, & F. J. Turk (Eds.), Satellite precipitation measurement (pp. 343–353, Vol. 67). Springer. https://doi.org/10.1007/978-3-030-24568-9_19
Hunt, E. B. (1975). Artificial intelligence. Academic Pr.
Jordan, J. (2018). Setting the learning rate of your neural network [Accessed: 2025-12-18]. https://www.jeremyjordan.me/nn-learning-rate/
Kutner, M. H., Nachtsheim, C. J., Neter, J., & Li, W. (2005). Applied linear statistical models (5th). McGraw-Hill/Irwin.
Liemberger, R., & Wyatt, A. (2018). Quantifying the global non-revenue water problem (tech. rep.). https://doi.org/10.2166/ws.2018.129
Mitchell, T. M. (1997). Machine learning. McGraw-Hill.
Oshchepkov, P. (2024). Trigger-level electron counting in the light dark

matter experiment using artificial neural networks [Bachelor thesis, Lund University].
Portuguez-Maurtua, M., Arumi, J., Lagos, O., et al. (2022). Filling gaps in daily precipitation series using regression and machine learning in inter-andean watersheds. Water, 14 (11), 1799. https://doi.org/10.3390/w14111799
Russell, S. J., & Norvig, P. (2020). Artificial intelligence: A modern approach (4th). Pearson.
Saha, S., Moorthi, S., Pan, H.-L., Wu, X., Wang, J., Nadiga, S., Tripp, P., Kistler, R., Behringer, D., & Coauthors. (2010). The ncep climate forecast system reanalysis. Bulletin of the American Meteorological Society, 91 (8), 1015–1057. https://doi.org/10.1175/2010BAMS3001.1
Sapakova, S. (2025). Development of machine learning methods for market trends. 65
Shah, S. Y., Larijani, H., Gibson, R. M., & Liarokapis, D. (2022). Random neural network based epileptic seizure episode detection exploiting electroencephalogram signals. Sensors, 22 (7), 2466. https://doi.org/10.3390/s22072466
SMHI. (2025). Smhi open data [Accessed: 2025-10-25].
Sörensen, J., Nilsson, E., Nilsson, D., Gröndahl, E., Rehn, D., & Giertz, T. (2024). Evaluation of ann model for pipe status assessment in drinking water management (tech. rep.). https://doi.org/10.2166/ws.2024.104
Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society: Series B (Methodological), 58 (1), 267–288.
Tukey, J. W. (1977). Exploratory data analysis. Addison-Wesley.
Vatten, S. (2023). Resultatrapport för vass drift 2023 (tech. rep.) (Accessed: 2025-09-13).https://www.svensktvatten.se/globalassets/dokument/organisation--styrning/va-statistik/r2024-06-resultatrapport-for-vass-drift-2023-3.pdf
Vrachimis, S. G., Kyriakou, M. S., Eliades, D. G., & Polycarpou, M. M.

(2018). Leakdb: A benchmark dataset for leakage diagnosis in water distribution networks. Proceedings of the 1st International WDSA / CCWI Joint Conference. https://doi.org/10.5281/zenodo.1313116

Weisberg, S. (2005). Applied linear regression (3rd). John Wiley & Sons.

Xie, P., Yatagai, A., Chen, M., Hayasaka, T., Fukushima, Y., Liu, C., & Yang, S. (2007). A gauge-based analysis of daily precipitation over east asia. Journal of Hydrometeorology, 8, 607–626. https://doi.org/10.1175/JHM586.1

Zhang, Y., Jiang, Z., & Lu, J. (2023). Research on leakage location of pipeline based on module maximum denoising [Describes how pipeline leak events create pressure drops (negative pressure waves) used for detection]. Applied Sciences, 13 (1), 340. https://doi.org/10.3390/app13010340

Note:
New chapters always start on odd page numbers

Pages with colour are expensive to print, please only use colour where absolutely necessary. Do not use coloured headings, tables and web links!

You will get 2 copies/author printed at the V-house janitor's office on TVRL's expense, unless a special agreement is made with your supervisor/sponsor.