



**Escola de Camins**  
Escola Tècnica Superior d'Enginyeria de Camins, Canals i Ports  
UPC BARCELONATECH

# Development of machine learning models for short-term water level forecasting

A case study on Storå River, Denmark

**Final Thesis developed by:**

Buse Onay

**Directed by:**

Prof. Allen Bateman Pinzon

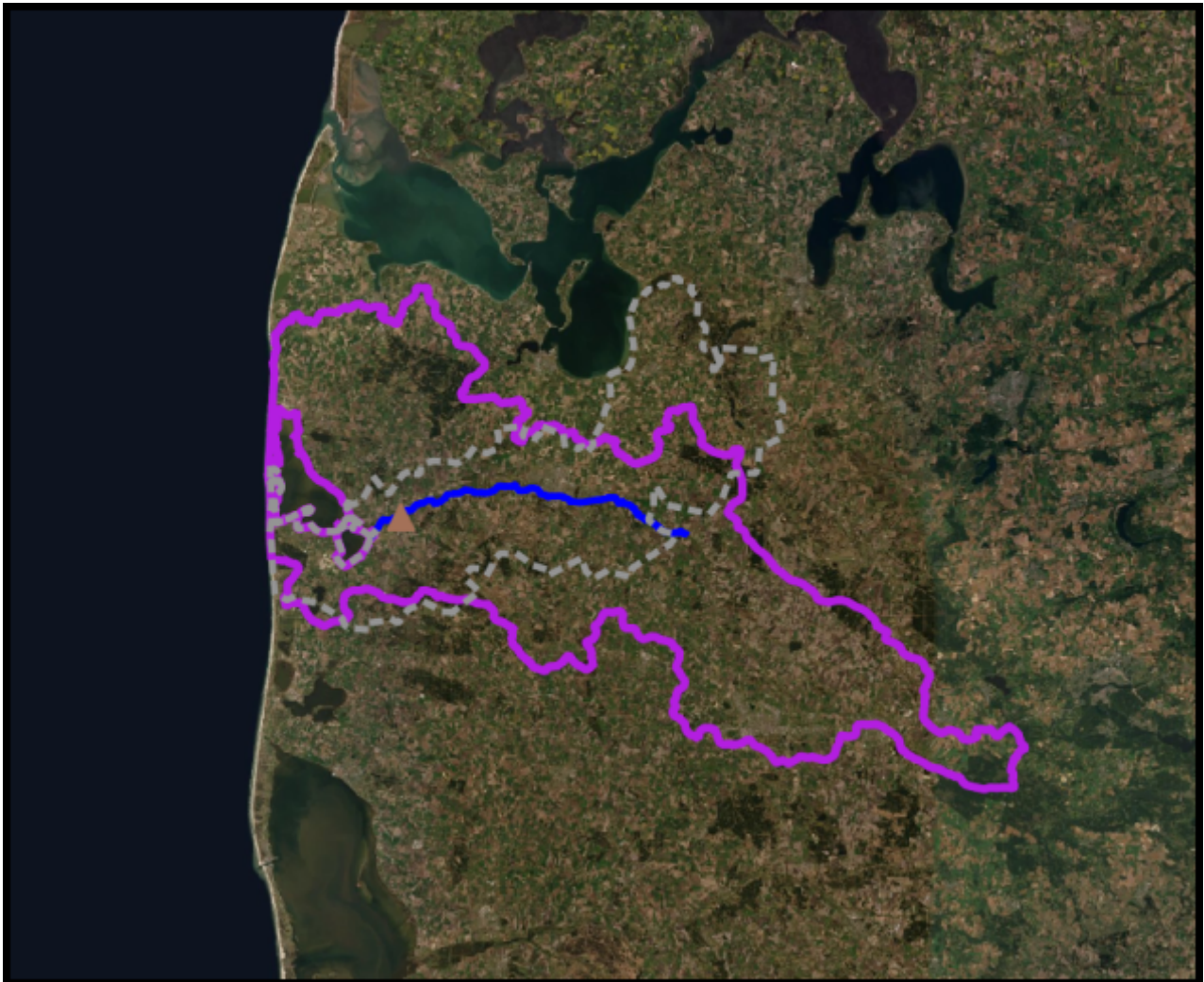
**Master in:**

Erasmus Mundus Joint Master Degree on  
Flood Risk Management

**Barcelona**

August, 2021

MASTER FINAL THESIS



## **Development of machine learning models for short-term water level forecasting**

A case study on Storå River, Denmark

Buse Onay

MSc Thesis  
August, 2021



# Development of machine learning models for short-term water level forecasting

A case study on Storå River, Denmark



Master of Science Thesis

by

**Buse Onay**

Supervisor

**Prof. Dr. Allen Bateman Pinzon (UPC Barcelona)**

Mentors

**Dr. Laura Frølich (DHI)**

**Dr. Nicola Balbarini (DHI)**

Examination Committee

**Prof. Dr. Allen Bateman Pinzon (UPC Barcelona)**

**Prof. Dr. Vicente Cesar De Medina Iglesias (UPC Barcelona)**

**Prof. Dr. Agustin Sanchez-Arcilla Conejo (UPC Barcelona)**

**Dr. Laura Frølich (DHI)**

**Dr. Nicola Balbarini (DHI)**

*This research is done for the partial fulfilment of requirements for the Master of Science degree in the  
Erasmus Mundus Flood Risk Management Programme*

*Barcelona*

*August, 2021*

© 2021 by Buse Onay. All rights reserved. No part of this publication or the information contained herein may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, by photocopying, recording or otherwise, without the prior permission of the author. Although the author and institutions involved have made every effort to ensure that the information in this thesis was correct at press time, the author and institutions involved do not assume and hereby disclaim any liability to any party for any loss, damage, or disruption caused by errors or omissions, whether such errors or omissions result from negligence, accident, or any other cause.

This work is licensed under the [Creative Commons Attribution-NonCommercial 4.0 International License](https://creativecommons.org/licenses/by-nc/4.0/)



## Abstract

*The impact of precise river flood forecasting and warnings in preventing potential victims along with promoting awareness and easing evacuation is realized in the reduction of flood damage and avoidance of loss of life. Machine learning models have been used widely in flood forecasting through discharge. However the usage of discharge can be inconvenient in terms of issuing a warning since discharge is not the direct measure for the early warning system. This paper focuses on water level prediction on the Storå River, Denmark utilizing several machine learning models. Multiple Linear Regression, Random Forest Regression, Gradient Boosting Regression, and Feed-Forward Neural Network were selected as machine learning algorithms used in this study. While the first three models were utilized in the assessment of features, the neural network model was used to compare the prediction performance of the models at the end. The methodology was developed to understand the effect of different feature transformation and scaling techniques on the machine learning models' performance. Furthermore the effect of different feature sets on the machine learning models' performance was investigated. Moreover the importance of feature selection utilization through filter and hybrid methods which is a combination of filter and wrapper methods were analysed. The study revealed that the transformation of features to follow a Gaussian-like distribution did not improve the prediction accuracy further. Additional data through different feature sets resulted in increased prediction performance of the machine learning models. Using a hybrid method for the feature selection improved the prediction performance as well. The Feed-Forward Neural Network gave the lowest mean absolute error and highest coefficient of determination value. The results indicated the difference in prediction performance in terms of mean absolute error term between the Feed-Forward Neural Network and the Multiple Linear Regression model was 0.003 cm. It was concluded that the Multiple Linear Regression model would be a good alternative when time, resources, or expert knowledge is limited.*

Keywords: machine learning, water level forecasting, fluvial flood, multiple linear regression, random forest, gradient boosting, feed forward neural network, filter method, wrapper method, feature transformation

## Acknowledgements

*I would like to express my gratitude to the Danish Hydraulic Institute (DHI) for their collaboration in the Flood Risk Management Master Programme. I cannot begin to express my thanks to my mentors Laura Frølich and Nicola Balbarini from DHI for their invaluable guidance, encouragement, motivation, patience, and assistance throughout this research work. I would like to extend my sincere thanks to my supervisor Prof. Allen Bateman Pinzon for his on-point comments. Thanks to the European Commission Erasmus Mundus Joint Master Degree Programme Committee for awarding me with the Erasmus Mundus scholarship that allowed me to participate in this master programme. Special thanks to my family and friends for their encouragement and support while I was pursuing this master programme during the Covid-19 outbreak.*

# Table of Contents

Abstract	i
Acknowledgements	ii
List of symbols and abbreviations	iii
<b>Introduction</b>	<b>1</b>
Background	1
Motivation	1
Objectives	4
Research Questions	4
Innovation and Practical Value	4
<b>Literature Review</b>	<b>5</b>
Background	5
Related Works	7
<b>Case Study</b>	<b>10</b>
Site Overview	10
Data Overview	12
<b>Research Methodology</b>	<b>16</b>
Methodology Schematic	16
Data Preparation	17
Data Collection and Visualization	18
Data Preprocessing	19
Feature Selection	20
Feature Transformation and Scaling	22
Data Split	23
Machine Learning	24
Linear Regression	25
Random Forest	26
Gradient Boosting	26
Artificial Neural Network	27
Evaluation Criteria	28
Improving Models	30
<b>Data Analysis</b>	<b>32</b>
Data Visualization and Preprocessing	32
Feature Selection	46
Correlation Analysis	46
Mutual Information	55

Persistence Model	56
Feature Sets	56
Data Split	57
Feature Transformation and Scaling	61
<b>Results and Discussion</b>	<b>67</b>
Part 1: Assessment of Feature Transformation and Scaling	67
Part 2: Assessment of Different Feature Sets	69
Part 3: Model Improvement	78
Effect of Hyperparameter Tuning	78
Effect of Recursive Feature Elimination	80
Part 4: Feed-Forward Neural Network	81
Part 5: Overall Assessment of Tested Methods	82
<b>Conclusion and Recommendations</b>	<b>96</b>
Main Conclusions	96
Limitations and Recommendations	99
Limitations	99
Recommendations	100
<b>References</b>	<b>102</b>

## List of symbols and abbreviations

ANN	Artificial Neural Network
ANN-GA	Genetic Algorithm-based ANN
APSFR	Areas of Potential Significant Flood Risk
ARIMA	Autoregressive Integrated Moving Average
ARMA	Autoregressive moving average
BPNN	Back Propagation Neural Network
CC	Correlation Coefficient
D-SVR	Distributed Support Vector Regression
ELM	Extreme Learning Machine
FRM	Flood Risk Management
KNN	K-Nearest Neighbors
LASSO	Least Absolute Shrinkage and Selection Operator
LR	Linear Regression
LSTM	Long Short-Term Memory
MAE	Mean Absolute Error
ML	Machine Learning
NNM	Neural Network Model
$R^2$	Coefficient of Determination
RBFNN	Radial Basis Function Neural Network
RF	Random Forest
RMSE	Root Mean Square Error
SARIMA	Seasonal Autoregressive Integrated Moving Average
SVM	Support Vector Machine
SVR	Support Vector Regression

# Chapter 1. Introduction

*This chapter presents the introduction of the research study. First floods will be discussed as a natural disaster in generic terms: How they occur, what are the consequences, how can it be avoided. The seriousness is highlighted with the increased precipitation in northern Europe in the context of this research work. It will continue with the motivation of this research study elaborating why it is needed to have water level forecasting in the Storå River. Afterwards, the objective of the study will be introduced and then research questions will be provided. Finally, innovation and practical value will be discussed.*

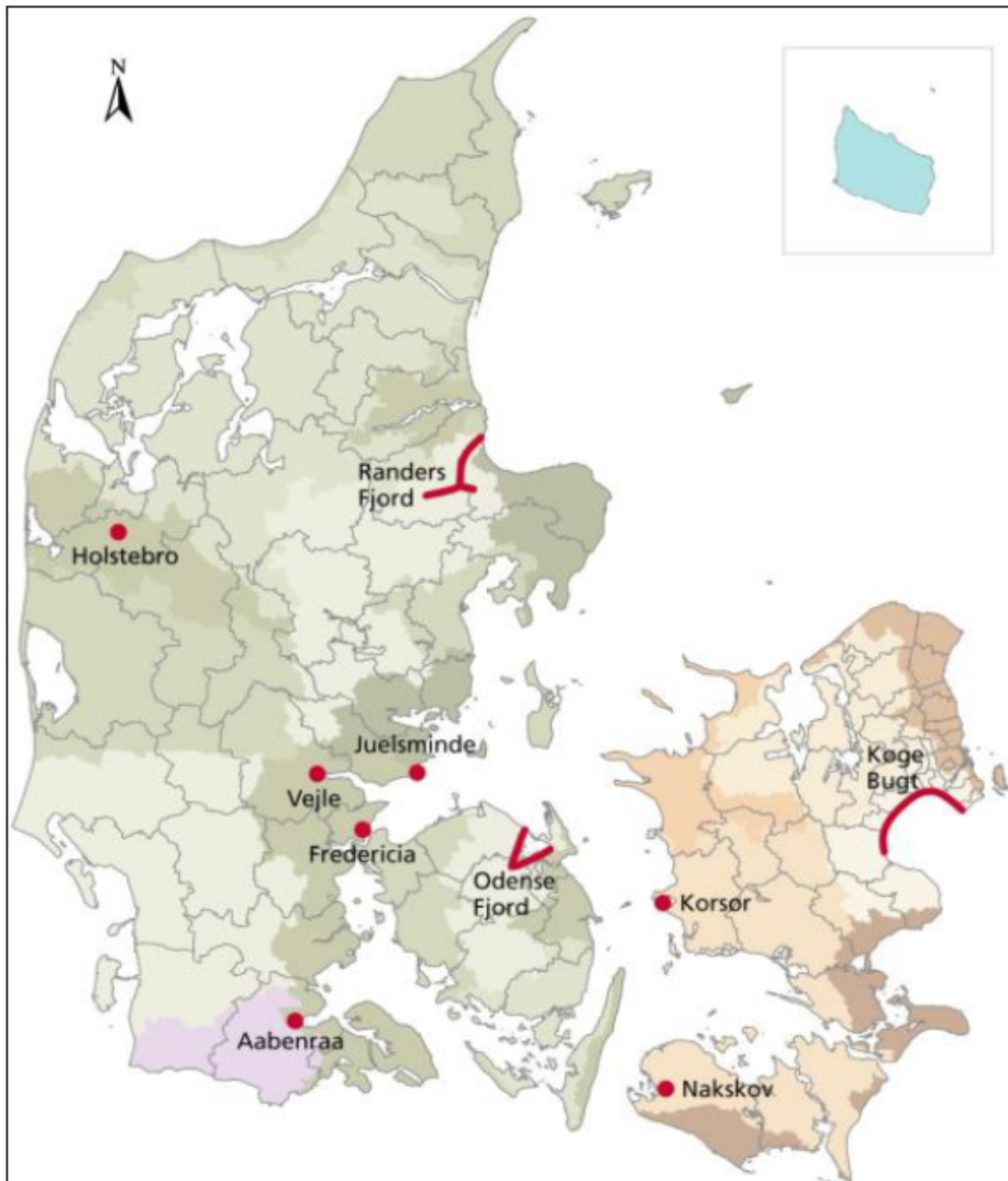
## 1.1 Background

Floods are among the most destructive natural disasters that have comprehensive consequences on human life and well-being along with social and economic losses at a community level. Among all weather-related disasters recorded between 1995-2015, flooding accounted for 47% affecting 2.3 billion people (UNISDR, 2015). Especially in the European Region, floods are the most common disasters, causing extensive damage and disruption (WHO, 2013). According to the European Environment Agency, 213 flood events were recorded between 1998-2003, affecting 3.145 million people. These floods brought more than EUR 52 billion overall losses (EEA, 2010). Unfortunately, the increasing population and urbanization rate in flood-prone areas increases exposure and thus, the damage potential of flooding. Furthermore, climate change, sea-level rise, and other anthropogenic factors exacerbate the current and predicted future flood risk. According to the World Health Organization, total precipitation during autumn and winter increased in northern Europe (WHO, 2013). Heavy and prolonged rainfall increases the stream's water level and poses a potential danger to riverine floods. Recently, several massive fluvial flooding events have highlighted the seriousness of the problem. It may not be possible to avoid flooding itself completely, but it is possible to reduce the negative impact and ease the aftermath with appropriate mitigation strategies, emergency preparedness, and recovery activities. Flood forecasting and early warning systems have been one of the most efficient and cost-effective measures for this purpose.

## 1.2 Motivation

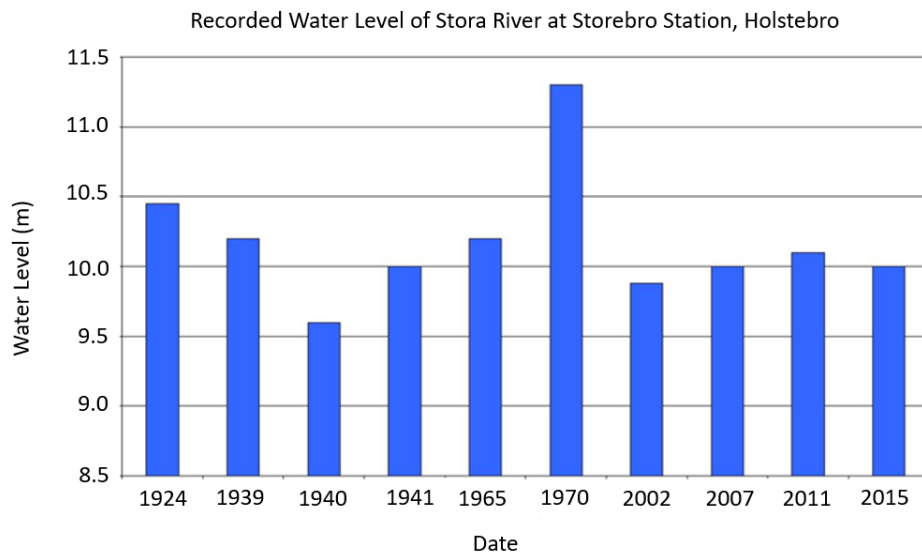
This study is motivated by the need for water level forecasting of Storå River, Denmark that is responsible for Holstebro municipality's flood risk. According to the EU Flood Risk Directive (2007/60/EC), member states are required to assess their territory for significant flood risk along with preparing flood maps, analyzing the damaging effect

of floods for human life and well-being, cultural and historical heritage, the economy and environment in these risky areas, and reducing the flood risk by adequate measures. As one of the member states, Denmark prepared Preliminary Flood Risk Assessments (PFRAs) to identify the Areas of Potential Significant Flood Risk (APSFRs) in the river basins and coastal areas at the end of 2011.



**Figure 1** | *Ten Risky Areas in Denmark appointed according to the EU Flood Directive.*

Ten risky areas were determined for flood potential due to storm surge, extreme river runoff, or both and presented in Figure 1. Jebens et al. (2016) explained the methodology of determining these areas. Among selected municipalities, Holstebro is the only one with an entirely fluvial risk source. The risk stems from the second largest river of Denmark, Storå River. The Holstebro city has been exposed to several floods throughout history due to the low-lying nature of the area and the capacity exceedance of the Storå River. Water levels regarding those events recorded in the Storebro Station which is located in the center of Holstebro, are presented in Figure 2. Among them, March 18, 1970 was by far the greatest one. The recorded water level at Storebro station in the city center was 11.3 m above sea level. More recent floods experienced by the Holstebro city are in 2007, 2011, and 2015 with the recorded water level of 10m, 10.1m, and 10m at the Storebro station. (Holstebro Kommune, 2015, n.d.)



**Figure 2** | *Water level through time at Storebro Station, Holstebro*  
Copyright 2021 Holstebro Kommune, Denmark.

For the city center, the current discharge model is utilized to issue a warning but this early warning system is not sufficient in covering all risky areas, especially the downstream of the city. Moreover, using discharge to issue a warning can be inconvenient, since discharge is not the direct measure for the early warning system. First, it needs to be converted into water level in order to be useful by the warning system. However, the rating curve of the area varies over a year due to altering the vegetation cover in the channel by cutting grass every spring and fall. Thus, the discharge to water level conversion creates some uncertainties in the area. Developing a water level forecasting model for the Storå River and utilizing this information in issuing warnings would be an appropriate solution to address fluvial risk in the area that can be applied quickly given the amount of information. Therefore, this study focuses on water level forecasting in the Storå River.

## **1.3 Objectives**

The objectives of this research study are river stage forecasting using different machine learning techniques, trying to achieve forecasting with 48-hours lead-time and investigating the effect of input selection on models' performance.

## **1.4 Research Questions**

Throughout the study, the questions listed below will be answered:

1. How does transforming features into normal distribution affect the machine learning algorithms' performance?
2. How do the different feature sets affect the model's performance?
3. How does using only the filter method and combination of filter and wrapper methods for feature selection affect the models' performance?
4. Which machine learning methods give better results based on the selected evaluation criteria?

## **1.5 Innovation and Practical Value**

This research aims to fill a research gap in fluvial flood forecasting for the selected site through predicting water level in the Storå River. It provides interesting insights to be used at other sites and that could strengthen fluvial flood forecasting for early warning, supporting prevention measures and remediation activities while a flood is happening. Moreover, the findings and the developed models may support and complement the existing system for water level forecasting in town by providing insights on areas nearby, especially downstream where most flooding is historically.

## Chapter 2. Literature Review

*This chapter focuses on a review of related literature in context with the study objective. This review of literature was conducted and presented in two sections. The purpose of this chapter is to provide an understanding on a background and the related works to this study. The background section was developed to give a reader insight into the current approaches in flood forecasting applications of machine learning models. The review was conducted for this topic with the purpose of obtaining a broad perspective. The related work section of this review is devoted to comparison of methods and insights of different machine learning models in water level forecasting in their applications found in case studies.*

### 2.1 Background

Over the last 20 years, machine learning methods have become a more popular tool to forecast various water source parameters. With the variety of machine learning algorithms, increasing computational power, and available data with the help of diverse and effective ways of measuring parameters, machine learning methods contribute more than ever to flood forecasting and detection. The amount of research in this area followed the same trend and a considerable amount of papers have been published. This progress led to some alterations in flood risk assessment in mitigation, response, and recovery phases as well (Wagenaar et al., 2020). Accurate flood forecasting enables authorities to develop effective flood risk management strategies by improving preparedness and increasing resilience before flooding, envisaging emergency response during flooding, and planning recovery after flooding. It has the utmost importance in flood hazard analysis and early warning systems to protect human life and their assets.

The current approaches for flood forecasting can be divided into two categories: the physics-based models and the data-driven models. Physics-based models use the known physical laws (Kisi and Ciziloglu, 2005), such as conservation of mass and momentum (Nguyen et al., 2013) in differential form through time and space (Hosseiny et al. 2020) which have been widely applied in the simulation of complex hydrological processes and flood dynamics (Kabir et al., 2020). Especially the late improvements in satellite remote sensing made it possible to reach fine-resolution data for terrain elevation and river morphology, enabling advanced physics-based modeling as seen in the Amazon River basin (De Paiva et al., 2013). While physics-based models show potential for flood prediction, they require extensive data to describe the site, and are computationally intensive, deeming their application to short-term forecasting as infeasible (Yang and Chang, 2020). Furthermore, most of these models are quite sophisticated, thus the development of these models requires in-depth knowledge and expertise about hydrological parameters (Hosseiny et al. 2020; Mosavi et al., 2018).

The data-driven models, e.g., machine learning (ML) models, on the other hand, focus on learning from historical data with developing relationships between features and target variables without having prior knowledge of the physical hydrological process (Mosavi et al., 2018). Although the data-driven methods also require large amounts of data, depending of course on the method, it becomes handy when the available physical models are not capable of capturing the physics in mathematical terms, the computational cost is impractical, or the available knowledge about the problems is limited. (Hosseiny et al. 2020).

Some data driven approaches combine different machine learning models. In the literature, there are numerous studies advocating a hybrid approach. Chen and Wang (2007) reported that a hybrid model of SARIMA (Seasonal Autoregressive Integrated Moving Average) and SVM (Support Vector Machine) performed better than SARIMA and SVM models alone in forecasting seasonal time series; Khashei and Bijari (2010) proposed a hybrid novel model of ANN (Artificial Neural Network) using ARIMA (Autoregressive Integrated Moving Average) models to improve predictive performance; Xie and Lou (2019) combined ARIMA and SVR (Support Vector Regression) to predict the water level more accurately; Phan and Nguyen (2020), proposed a hybrid approach by combining ARIMA with RF(Random Forest), SVM, KNN (K-Nearest Neighbors), and LSTM (Long Short-Term Memory). They revealed that the hybrid approach has advantages over individual base models.

Literature review revealed that there are various studies developed with different lead-times for flood predictions. They can be categorized under short-term and long-term predictions. Short-term flood predictions generally refer to hourly, daily, and sometimes weekly predictions, which also help issue warnings. Long-term predictions, contrarily, are often utilized by authorities, flood risk managers, or both in policy making regarding flood resilience. There is a diversity in timeline definition of long-term predictions. According to WHO, a forecasting period of more than ten days is defined as long-term; in another source, when prediction lead time to flood is three days longer than the confluence time, it is considered as long-term prediction (WMO, 2007; Mosavi et al., 2018). In this paper, a lead time of more than a week is considered as a long-term prediction.

Short-term and long-term predictions can ease the flood damage successfully. Various accomplished predictions for short-term lead-time have been encountered in the literature of machine learning methods. Toth et al. (2000) compared the performances of ANN and KNN in short-term rainfall predictions. According to results, ANN predictions were superior with lead-times from 1 to 6 hours. Leahy et al. (2008) demonstrated ANN usage in river level prediction with a 5-h lead time. Yu et al. (2017) compared RF and SVM performances using radar-derived rainfall data in real-time flood forecasting. Overall, SVM performed better, yet both models demonstrated satisfactory results for 1-h ahead forecasting.

For long-term predictions, Elsafi (2014) used ANN to forecast seasonal flooding which ended up providing reliable results in forecasting flood hazard in the Nile River. Singh and Borah (2013) utilized FFBPNN (Feed-Forward Back-Propagation Neural Network) to build several forecasting models for Indian summer monsoon rainfall which showed superiority over the existing models and predicted seasonal rainfall values for upcoming 5 years for India. Lin et al. (2006) compared the performance of SVM to benchmark ANN and ARMA (Autoregressive Moving Average) models for long-term discharge predictions and proved the SVM can be considered as a potential candidate in long-term discharge predictions.

## 2.2 Related Works

This research narrows down to water level forecasting in river systems taking into consideration the objectives in this study. Although river discharge is forecasted commonly in river systems, it is not favorable to issue warnings. First, it needs to be converted into water level using a rating curve. However, there are some uncertainties in this process due to the imperfect relationship between river discharge and water level. Water level forecasting, on the other hand, is more practical to issue warnings because the exceedance of a certain level is more actionable by authorities. Therefore, the water level forecasting model for river systems has attracted increasing attention due to its convenience in flood forecasting (Yu et al., 2006).

A fluvial flood or river flood occurs when the water level in its channel exceeds the capacity and eventually leads water to overtop its bank and inundate the surrounding areas. Heavy and prolonged rainfalls, torrential meteorological activities such as cyclones, typhoons, etc., and rapid snowmelt tend to cause water level rise in the channel and, thus, river flooding. The expected consequences can be counted as a loss of human life, property damage, deterioration of environmental conditions - particularly water quality, loss of crops and livestock. Other than these, the interruption of businesses which depend on the location, duration, vulnerability of the exposed community, and several other factors can be counted as the negative consequences as well. Accurate river flood forecasting is an essential non-structural measure that helps to minimize these potential damages and losses. Various machine learning methods (Chen et al., 2014; Kisi and Ciziloglu, 2005; Wu et al., 2009a; Wu et al., 2009) were successfully applied in river flood forecasting.

Different water quantity variables (water level, discharge, runoff depth, precipitation, peak flow) have been used in river systems for flood forecasting depending on the available data, selected site, and the objectives of the study. A vast amount of papers utilized river flow as an input parameter to forecast river flooding (Lima et al. 2016; Atiquzzaman and Kandasamy, 2015; Li and Cheng, 2014; Abrahart et al., 2007; Aqil et al., 2007; Bae et al., 2007; Chang et al., 2007; Corzo and Solomatine, 2007; Jia and Culver, 2006; Wang et al. 2006). Some papers used water level as an input parameter (Yu et al., 2006; Chau, 2007; Zehra, 2020). There are also other papers using runoff depth

(Wang et al., 2015); peak flow (Sun and Trevor, 2017); precipitation (Wang et al., 2015) as an input parameter.

Thirumalaiah and Deo (1998) used an ANN model in river stage forecasting for River Godavari, India. Daily continuous water level data from 1988 to 1991 for the observed station, upstream stations, or both are utilized in the paper. The trial-and-error method is used to determine the number of input stations for ANN. The network is trained by using three different algorithms: error back propagation, conjugate gradient, and cascade correlation. According to the results, remarkably high iterations and training time are required for the back propagation algorithm. Training time for the cascade correlation algorithm, contrarily, takes a small fraction of time. Additionally, the paper argues the usefulness of using both the given and upstream station in forecasting water level because using the data from only the given station would be enough if the training algorithm is chosen wisely.

Liong et al. (2000) successfully implemented an ANN model into river stage forecasting in Dhaka, Bangladesh. Daily river stage data collected from eight gauge stations between 1991-1996 is used to build the ANN model. Data recorded for three years with great variety in river stage is selected for model training, and remaining data reserved for verification. As a result, highly accurate results are obtained even for a 7-lead-day model. Sensitivity analysis was also performed and 3 out of 8 input neurons were eliminated without significantly affecting the accuracy of water prediction, which enables policymakers to reduce unnecessary data collection and operational cost.

Chang and Chen (2003) used RBFNN (Radial Basis Function Neural Network) in water stage forecasting for an estuary influenced by high flood and tidal effects. The RBFNN consists of an unsupervised and supervised learning scheme. For the first stage, fuzzy min-max clustering and for the second stage, multivariate linear regression has been used. Hourly water-stage data from 6 stations in the Tanshui River under tidal effects are utilized to construct the model. The result showed that the RBFNN gives accurate results with a 1-h lead-time for water-stage forecasting.

Sung et al. (2017) addressed water level forecasting for a tributary affected by the main river condition using machine learning models in South Korea. The existing ANN model, which uses rainfall and upstream water level as input, is improved by adding multiple water level data on the main river to include the backwater effect from the main river. Hourly rainfall and water level data only during the monsoon period from 21 June to 20 September between 2007-2016 is used to construct the model. Several ANN models with different lead times and complexity are built, and their performances are compared. Based on results, the best ANN water level forecast model performed a small error with lead-times of 1-2 hours, and ANN models are able to include backwater effects in water level forecasting without the use of complex physical models.

Yu et al. (2006) performed real-time stage forecasting using an SVR model in Lan-Yang River, Taiwan. Hourly water level data from two stations on the river and hourly precipitation data from 10 rainfall stations in the basin between 1990-2004 have been used to build the SVR model. The SVR model can predict the flood stage with lead-times of 1-6 hours based on results.

Multiple studies compared the performance of machine learning models in water level forecasting for river systems. Wu et al. (2008) compared several machine learning models in river stage forecasting of Yangtze River, China. Those models were LR (Linear Regression), NNM (Neural Network Model), ANN-GA (Genetic algorithm-based ANN), Conventional SVR, D-SVR (Distributed Support Vector Regression), namely. According to validation results, the D-SVR model is better at predicting water level than other proposed models and decreasing the training time remarkably compared to the Conventional SVR.

Nguyen and Huu (2015) compared the performance of three machine learning methods, namely, LASSO (Least Absolute Shrinkage and Selection Operator), RF, SVR, in daily water level forecasting. Continuous daily water level data from the Mekong River is utilized as input for the models for 1994 to 2003. The results revealed that the SVR model provides accurate results based on the Mekong River Commission requirements.

Alvisi et al. (2006) compared ANN with fuzzy logic in water level forecasting Reno River, Italy. In the models' construction, the same datasets with varying spatial and temporal information have been used to analyze the model's performance under different information levels. The results showed that the fuzzy logic approach performs better with a limited number of variables and IF-THEN logic statement, which links the input and output variables, and ANN shows better performance with detailed information and greater reliability.

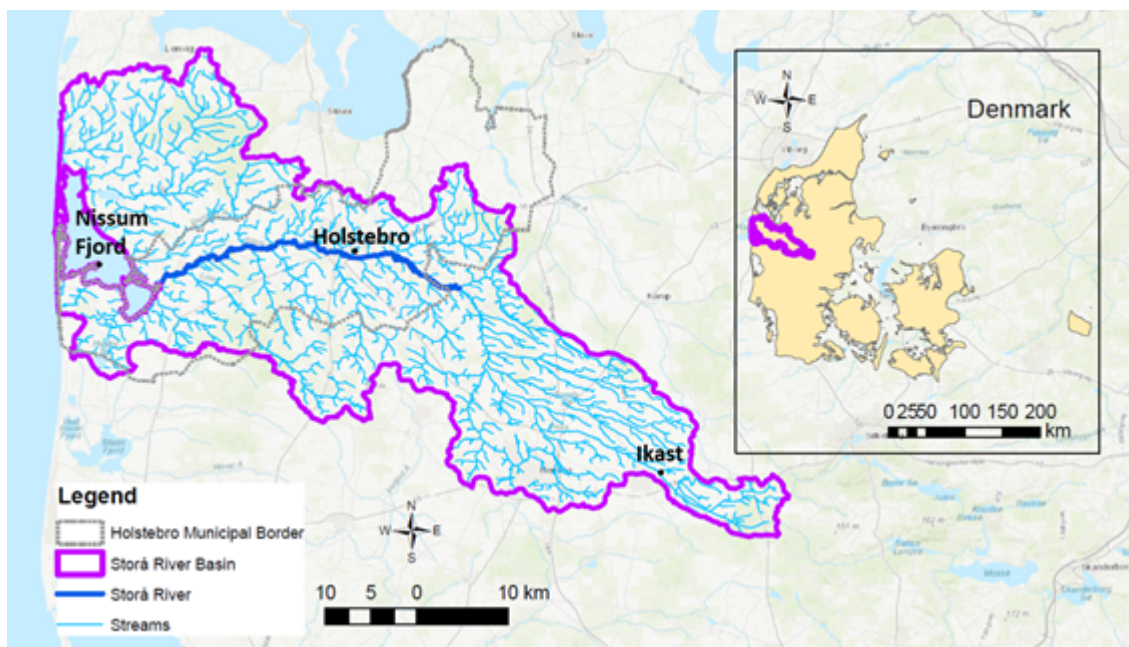
## Chapter 3. Case Study

*This chapter presents information about the site and gives an overview of the data utilized in this case study. In the site overview section, some statistics about the case study area are presented. An orientation of the topographic information of the area is provided. An overview of the case study data was provided in discussing the technical descriptions from the sources of the inputs to the methodology. Brief information about the data is provided in this section, including location of the station, source, unit, availability, frequency of the data.*

### 3.1 Site Overview

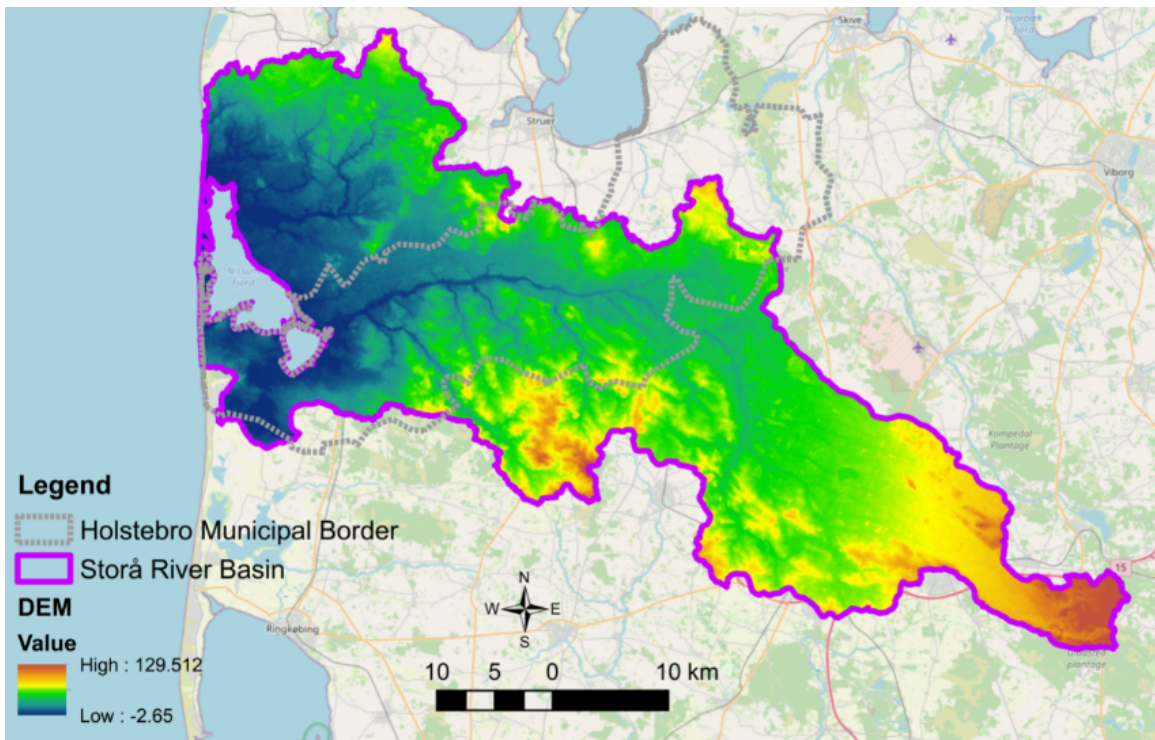
Storå River is the second largest river in Denmark, spanning 104 kilometers. The river originates from the south-east of Ikast, crosses the Jutland peninsula to the north-west through the Holstebro municipality before finishing its course in the Nissum Fjord.

Overview of the site area is presented in Figure 3. The catchment area of Storå River in Holstebro municipality is approximately 825 km<sup>2</sup> and only the part of the Storå River that lies within the borders of Holstebro municipality is visualized. (Holstebro Kommune, 2011)



**Figure 3** | Overview of the Case Study Area: Storå River Basin, Denmark.

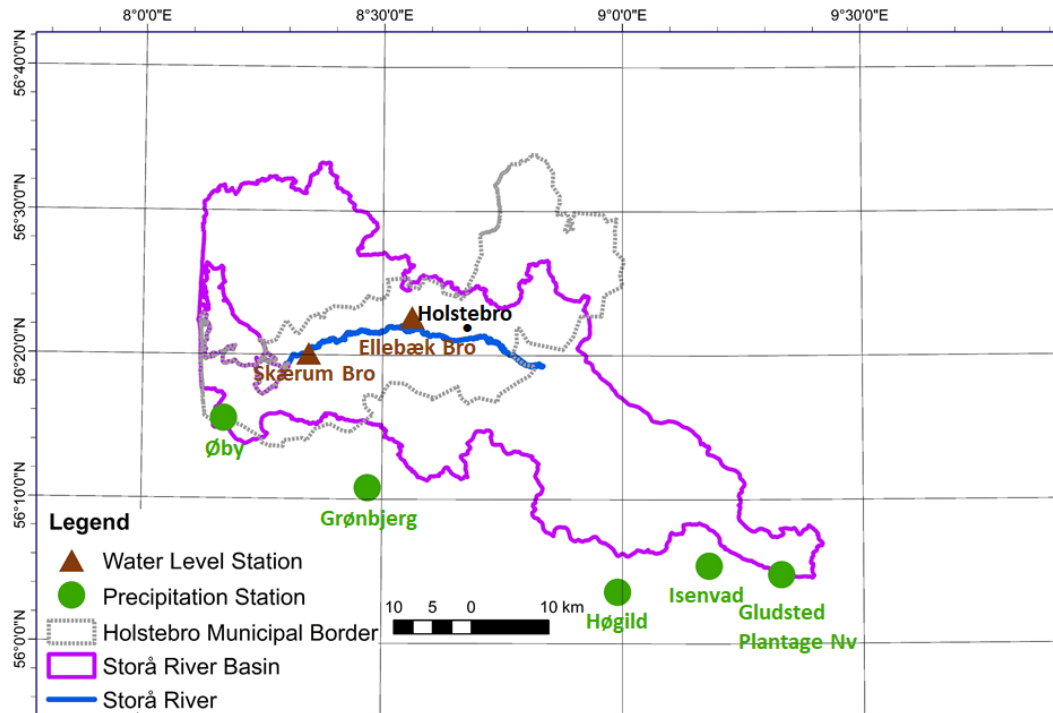
The European Union Digital Elevation Model (EU-DEM) is downloaded for E40°N30° through Copernicus Land Monitoring Service and zoomed in the case study area as presented in Figure 4. The EU-DEM is a hybrid Digital Elevation Model (DEM) product based on SRTM (Shuttle Radar Topography Mission) and ASTER-GDEM (Advanced Space-borne Thermal Emission and Reflection Radiometer - Global Digital Elevation Model) data, merged using a weighted averaging approach (EEA, 2017). Based on the topographic map with a spatial resolution of 25 m in Figure 4, the area lies in between Stora River Basin and Holstebro Municipal Border has low elevation, even reaching below sea level in the mouth of the river.



**Figure 4** | *Topography of the Case Study Area: Storå River Basin, Denmark.*

## 3.2 Data Overview

In this research, historical precipitation, forecasted precipitation, historical water level, relative soil moisture contents coming from the MIKE 11-NAM (NedborAfstørnings Model-Rainfall - Runoff Model) model, and simulated discharge coming from hydrologic model for the area are used to forecast water level in Skærum Bro Station. Spatial orientation of the station network is presented in Figure 5.



**Figure 5** | *Location of the Stations.*

For historical precipitation, data is collected from DMI (Danish Meteorological Institute) for five different stations, namely, Isenvad, Grønbjerg, Øby, Høgild, and Gludsted Plantage Nv. The frequency of the historical precipitation data is one hour, and data availability is presented in Table 1.

**Table 1 | Data Information, Availability and Frequency**

Data				Availability		Frequency
Name / Coordinates	Parameter	Unit	Source*	Start	End	
Isenvad	Observed Precipitation	mm	DMI	05-01-10 5:00	12-10-20 23:00	hourly
Grønbjerg	Observed Precipitation	mm	DMI	04-01-10 15:00	12-10-20 23:00	hourly
Øby	Observed Precipitation	mm	DMI	02-01-10 4:00	12-10-20 23:00	hourly
Høgild	Observed Precipitation	mm	DMI	20-12-11 22:00	12-10-20 22:00	hourly
Gludsted Plantage Nv	Observed Precipitation	mm	DMI	30-10-13 8:00	12-10-20 22:00	hourly
56.5°N/8°E/56°N/9°E	Forecasted Precipitation	mm	ECMWF	02-01-07 12:00	31-12-19 12:00	12-hour
Skærum Bro	Observed Water Level	m	MST	23-12-97 0:00	30-07-20 1:00	15-min
Ellebæk Bro	Observed Water Level	m	MST	24-12-97 0:00	12-10-20 7:00	15-min
Skærum Bro	Simulated Relative Moisture Content, L	%	DHI	02-01-07 12:00	31-12-19 12:00	2-day
Skærum Bro	Simulated Relative Moisture Content, U	%	DHI	02-01-07 12:00	31-12-19 12:00	2-day
Skærum Bro	Simulated Discharge	m <sup>3</sup> /s	DHI	01-01-11 0:00	01-01-21 12:00	12-hour

\* DMI: Danish Meteorological Institute, ECMWF: European Centre for Medium-Range Weather Forecasts, MST: Danish Environmental Protection Agency, DHI: Danish Hydraulic Institute

The forecasted precipitation data used in this research work retrieved from TIGGE (THORPEX Interactive Grand Global Ensemble) which has been developed as a part of THORPEX (THE Observing-system Research and Predictability EXperiment ) programme. TIGGE offers ensemble forecast data coming from 13 worldwide numerical weather prediction (NWP) centres. Due to its public availability and extensiveness, it is commonly used for non-profit research purposes (Park et al., 2008). For this research work, total precipitation data coming from ECMWF (European Centre for Medium-Range Weather Forecasts) with a 48-hour lead time is downloaded. The time interval is selected as 12 hours. Data represents a control forecast which has unperturbed initial conditions at the surface for the bounding coordinates of 56.5°N 8.0°E 56.0°N 9.0°E. This area represents a broader area when compared with the case study area presented in Figure 6. The grid resolution of the forecasted data is 0.5°x0.5° which is approximately 55kmx55km. At the end of 2016, a considerable portion of 2017 (8 months) was missing from the ECMWF system. This discontinuity in the data sets a limitation for the study.

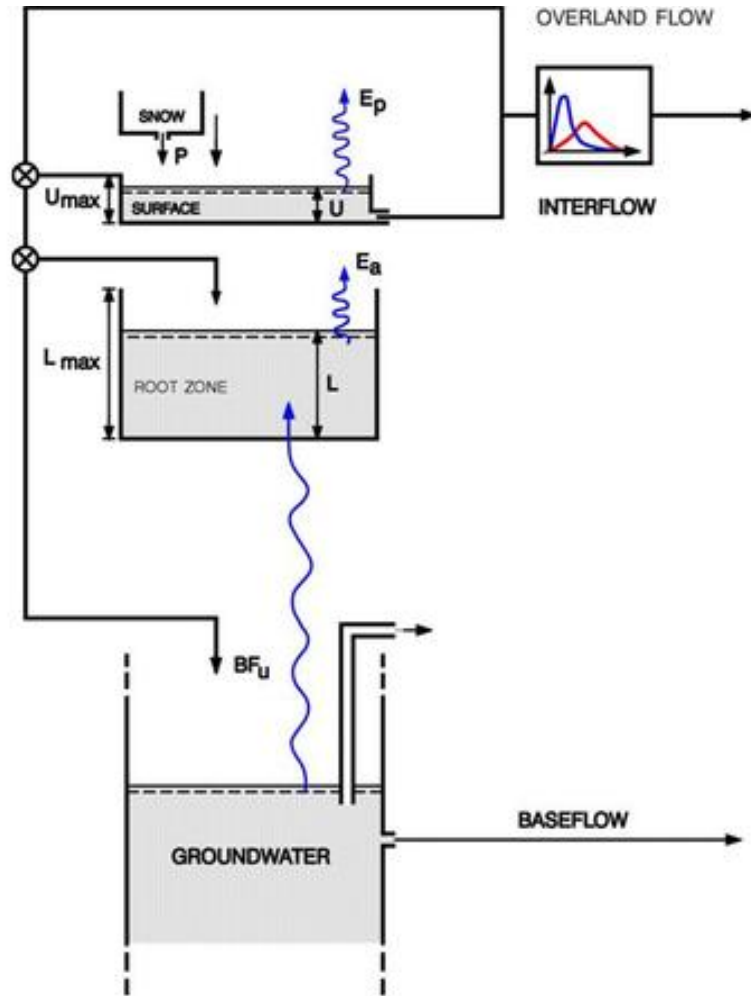
For water level, data is collected at Skærum Bro Station from DEPA (Danish Environmental Protection Agency) with a time interval of 15 minute. Although Hostebro city center is hazardous in terms of fluvial flooding, water level data from the stations in the city center was not considered for input selection due to two limitations. First, the city's sewage system influences the water level in the Storå River. Second, periodical water management activities through a gated structure in the Vandkraft Lake which is located upstream of the city center, alter the water level in the Storå River. The recorded water levels in the city center would be impractical to use as input since the natural

hydrologic process is disturbed by these two interventions. The location of Vandkraft Lake is presented in Figure 6. On the other hand, Skærum Bro Station is reasonably far from the abovementioned disturbances which makes it a suitable candidate to perform water level predictions for this study. Additionally, the water level from Ellebæk Bro Station is selected as an accompaniment in order to grasp an idea about the upstream tributary water level condition. Contribution of snowmelt was not considered in this research because there is no mountain in the area that can form a snow storage.



**Figure 6** | *Location of Vandkraft Lake. Alteration of water level measurements at Storebro Station due to water level measurement through the gated structure at the Vandkraft Lake.*

Since antecedent soil moisture or wetness situation of a soil is a distinctive parameter in describing the catchment's pace of generating subsurface, surface, and base flows from the precipitation (Casper et al., 2007 and Bronstert et al., 2012), relative soil moisture data is included in this research. The data is collected from the Rainfall-Runoff Model for Denmark, created using MIKE 11 modeling package - NAM modules and prepared by DHI (Danish Hydraulic Institute) (DHI, 1999). Relative soil moisture content,  $L$  represents water content in the root zone divided by maximum water content in the root zone, and relative soil moisture content,  $U$  represents surface water storage divided by maximum surface water storage. The structure NAM model showing the conceptual idea behind soil moisture contents  $U$  and  $L$  is presented in Figure 7.



**Figure 7** | *Structure of the NAM model*

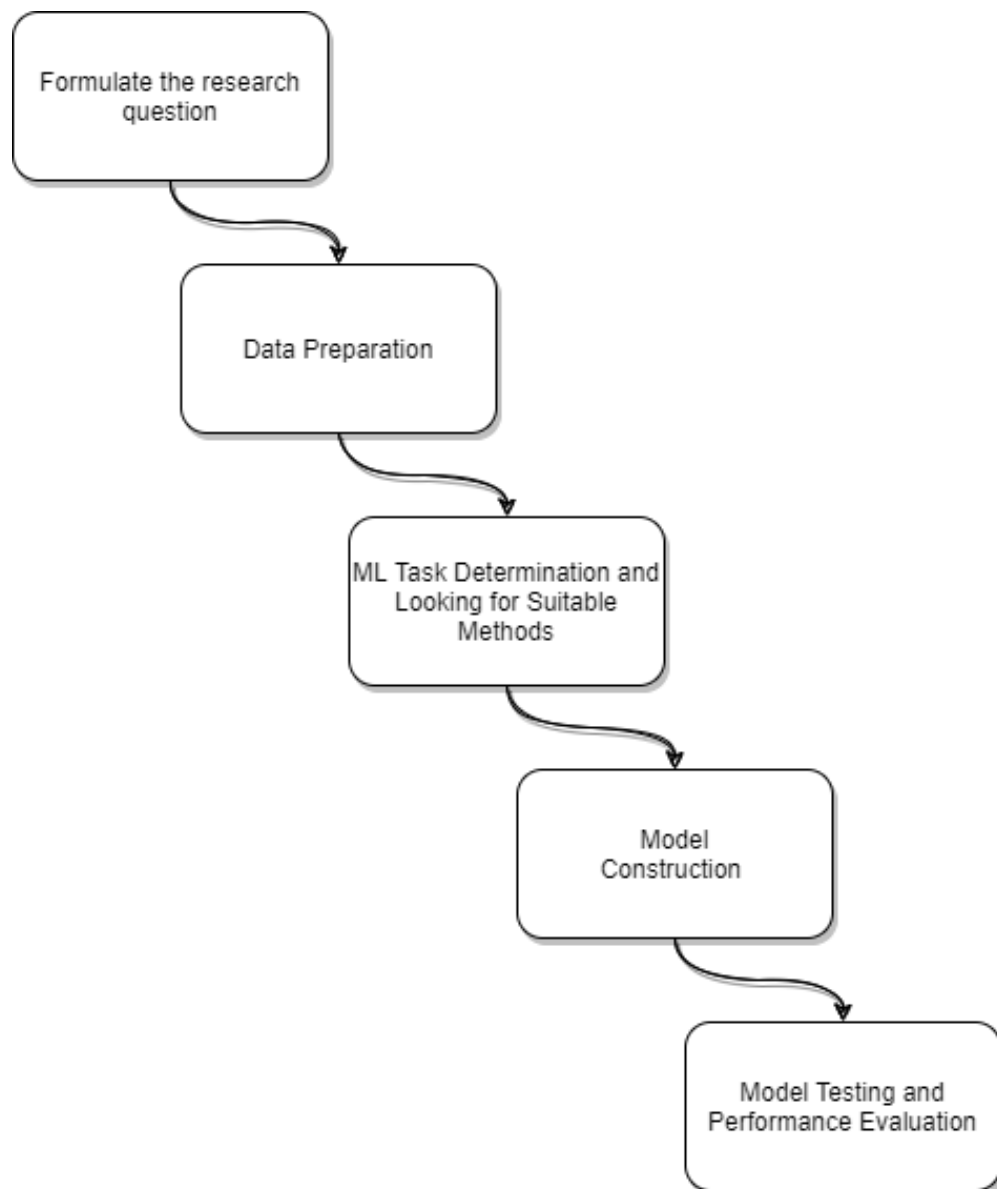
Finally, simulated discharge is used as another parameter in this research. The data is collected from an hydrologic model for the area with a 6-hour interval. Summary of the data used in this research is presented in Table 1.

## Chapter 4. Research Methodology

*This chapter revolves around a discussion on the research methodology and corresponding theory behind each procedure. First, the importance of data preparation and the steps of preparing data will give context to initializing this study. Afterwards, the discussion will continue with a theoretical background of machine learning models that were utilized in this research. Furthermore, the evaluation criteria and ways to improve machine learning model's performance were addressed. Ultimately, the purpose of this chapter is to provide the foundational description of the approach to answer the proposed research questions.*

### 4.1 Methodology Schematic

Formulating the research questions forms a backbone of any good research (Ratan et al., 2019). The main objectives and research questions were already defined in Chapter 1. Defining a research question gives an idea about the required data. Thus, the research will be continued with data preparation. This step includes acquisition and visualisation of data, preprocessing of data, feature selection, feature transformation and scaling, and data splitting. Afterwards, based on the collected data, identifying the machine learning task i.e. supervised or unsupervised and suitable machine learning algorithms will be addressed. Additionally, theoretical information about those algorithms will be provided. Model construction will then take place using training data. This will be an iterative process with the use of validation data. After the model construction is concluded, the model testing and performance evaluation will take place in order to investigate the prediction power of the model that was built. Finally, the formulated research questions will be answered with the prediction coming from the machine learning models. The theoretically explained research methodology is briefly visualized in Figure 8.



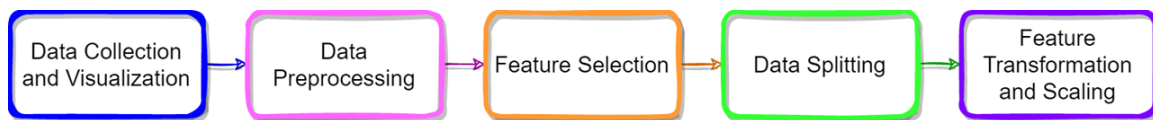
**Figure 8** | *Overview of Methodology*

## 4.2 Data Preparation

Data preparation is the primary step for any machine learning model. In basic terms, data preparation includes sets of procedures starting from how to get required data for the articulated problem up until creating ready-to-use data by the selected machine learning algorithms.

Machine learning models depend profoundly on the data. Data makes the training of a model, thus the whole machine learning concept possible. However, data often comes up with some missing values, quality issues, errors, or other flaws. Unless these issues are resolved, the predictions of machine learning algorithms might be useless, even misleading. Additionally, storing data may vary among organizations. Thus, formatting data may be required to make them all consistent with each other. In this sense, data preparation is vital in order to achieve accurate model outcomes. It helps to examine the data in meaningful ways and to extract the useful information. Data preparation enables further improvements of a model's performance as well (Wu et al., 2008). If the data does not align with the requirements of machine learning algorithms or contains erroneous information, it would cause communication problems during training of machine learning algorithms and lead to failure or impractical results. With proper data preparation, a machine learning model is more likely to generate remarkably better results than a model with no or poor data preparation.

For this research, water level, observed precipitation, forecasted precipitation, simulated soil moisture content, and simulated discharge data were preferred to use as described in Chapter 3.2. The procedural workflow that is planned to pursue for data preparation is presented in Figure 9.



**Figure 9** | *Data Preparation Workflow*

In order to avoid ambiguity in terms, data refers to the information collected through different organizations that consist of dependent (y) and independent (X) variables. When they are put in together they form a dataset (X, y). Here X represents the features and y represents the target variable that is being tested during an experiment. In this research the aim is to predict the water level in Storå River, thus, water level coming from Skærum Bro Station is the target variable. Rest of the data is considered as feature variables.

#### **4.2.1 Data Collection and Visualization**

Data collection and visualization require a properly articulated problem statement. It is not possible to guess which data is required without the established problem. After the problem and target variable needed to be predicted are defined, looking for data and going through different data collection mechanisms takes place. It is important to remember, data storing among different organizations may vary, and in the end, the gathered data might have inconsistent formatting. These formatting issues demand the

installation of different libraries depending on the selected programming language to import data. The imported data constitutes the "raw data". It is not feasible for a human being to draw inferences or grasp the underlying distribution in raw data by just looking at the numbers in this day and age. Data visualization helps identify anomalies, missing data, hidden patterns, cyclic representation, and other useful features that are potentially helpful in improving data quality and thus predicting performance of the machine learning algorithm. In this perspective, data visualization can be thought of as the bridge between data collection and data preprocessing.

#### **4.2.2 Data Preprocessing**

Data preprocessing refers to the interventions in raw data to convert it into a clean dataset that can be ready to use by a machine learning algorithm for training. The dataset provided at the beginning might require some work regarding unit conversions issues, the presence of outliers, noisy data and duplicated indices, improper formatting, dealing with missing or null values, encoding the categorical data, and more. The preprocessing step includes identifying and detecting inaccurate data and deleting, modifying, or changing it to improve the efficiency and prediction accuracy of the machine learning model.

Among all, missing data poses the biggest threat for this research due to incomplete datasets. One way to handle missing data is to remove all the missing values. Many machine learning algorithms can deal with missing data, but this does not mean that they should. Removing missing values can cause valuable information loss and jeopardize the deduction power of machine learning algorithms. Trying to find ways to handle missing data would be a better approach to proceed. There are many different ways to address this problem. Since the implementation of missing data techniques is not an objective of this research, some fast imputation techniques can be considered such as imputing missing values using mean, median, the most frequent value, and zero value. However, imputation of missing precipitation data should be considered separately because precipitation is not continuous both spatially and temporally by nature (Hema and Kant, 2017). In literature, there are several studies for imputation of missing precipitation data either on a monthly (Kajornrit et al., 2012) or daily (Tang et al., 2009; Ly et al., 2011; Lee and Kang, 2015) basis. However, hourly precipitation observations exhibit very high variation, which is even referred to as random behavior (Hema and Kant, 2017). Therefore, it is quite a challenge to determine the correct precipitation patterns for any length of missing data even for a single point considering the presence of dry days.

### 4.2.3 Feature Selection

Having access to an abundance of data creates a challenge for machine learning practitioners to extract important information. Feature selection is one of the most critical steps when building a machine learning model. There can be numerous data, but not all of them would be useful for the model. In fact, in general, adding so many variables increases the overall complexity, training time, computational cost, and a chance of overfitting the model while decreasing the overall accuracy. In basic terms, feature selection reduces the redundant or relatively less important features, making the machine learning model learn from more relevant features and contributes more to the model's performance with less resources. In the classification of feature selection methods, several different ways can be utilized. The most pronounced ones can be summarized into four main categories: Filter, Wrapper, Embedded and Hybrid (Hoque et al., 2014).

Filter method chooses a subset of features based on their performance in various statistical tests without help from a machine learning algorithm. There are many filter methods widely used in the literature performing different tasks like classification, regression, or clustering. (Jovic et al., 2015). Filter methods are generally considered as superior to other methods due to their computational speed, statistical robustness, simplicity, and cost effectiveness (Guyon and Elisseeff, 2003; Yu and Liu, 2003). Pearson's product moment correlation (for short: Pearson's correlation), information gain, chi-square test, fisher score are some of the common filter methods for feature selection. These methods are not applied on the test dataset.

Pearson's correlation is a statistical analysis that measures the amount of linearity among each independent feature with a target variable for prediction. The definition of Pearson's correlation coefficient involves division of covariance to standard deviation of two random variables. If  $x$  and  $y$  are considered as two random variables, then the formula for calculating the Pearson's correlation coefficient can be presented as:

$$r = r_{xy} = \frac{Cov(x, y)}{S_x \times S_y} \quad (4.1)$$

where  $S$  represents standard deviation for variable  $x$  and  $y$  separately,  $Cov(x,y)$  represents the covariance among the given variables, and  $r_{xy}$  represents Pearson's correlation coefficient.  $r$  takes values in between -1 and +1. Values getting closer to -1 represents negative correlation which means if one value decreases the other one would increase, those getting close to +1 represents positive correlation which means if one variable increases the other one would increase as well, and those close to 0 means no correlation in between variables (Kumar and Chong, 2018).

Mutual information is another statistical analysis that can be used in feature selection based on the filter method. It originates from information theory and is widely

used to apprehend the relevance and redundancy in between feature and target variables. Unlike the correlation coefficient which does not allow the demonstration of dependencies, the mutual information is sensitive about it. Specifically, mutual information is an amount of information obtained about one random variable by observing the other random variable. As another definition, it is the reduction in the uncertainty of one random variable because of a knowledge about the other. The mathematical formulation for two random variables  $x$  and  $y$  can be presented as following:

$$I(x; y) = \sum p(x, y) \log \frac{p(x, y)}{p(x)p(y)} \quad (4.2)$$

where  $I(x, y)$  is the mutual information between  $x$  and  $y$ ,  $p(x, y)$  is the joint probability distribution function of  $x$  and  $y$ , and  $p(x)$  and  $p(y)$  are the marginal probability distribution functions for  $x$  and  $y$ . In the concept of entropy which represented unpredictability of a random variable, mutual information can be also presented as:

$$I(x; y) = H(x) - H(x | y) \quad (4.3)$$

where  $H(x)$  is marginal entropy,  $H(x|y)$  is conditional entropy. This represents the unpredictability of a random variable  $x$  decreased by observing  $y$  (Kraskov et al., 2014; Hoque et al., 2014; Zeng et al., 2014).

Wrapper method selects feature subsets by using the prediction performance of a learning algorithm. Compared to filter methods, wrapper methods are much slower and computationally expensive yet find feature subsets better fitted to the predetermined machine learning algorithm that leads supreme learning performance (Guyon and Elisseeff, 2003; Yu and Liu, 2003; Hoque et al., 2014; Jovic et al., 2015).

The embedded method performs feature selection during the training of a model and is usually particular to a given learning algorithm. In other words, the embedded method does not differentiate learning from feature selection. This distinctive feature of embedded methods differentiates it from filter and wrapper feature selection (Guyon and Elisseeff, 2003; Lal et al., 2006; Hoque et al., 2014).

Hybrid method is a combination of filter and wrapper methods. Main idea is exploiting the best properties of methods: High accuracy comes from the wrapper method and high efficiency comes from the filter method. Hybrid method utilizes the filter

method to create a list of features in the ranked order, and utilizes the wrapper method after to create nested subset of previously listed features using machine learning algorithm (Hoque et al., 2014; Jovic et al., 2015; Ben Brahim and Limam, 2016).

#### 4.2.4 Feature Transformation and Scaling

Depending on the data sets, the range of values might be very distinct for every feature. Machine learning algorithms focus only on the numbers and having much higher values in one feature column compared to others may create some communication problems for the machine learning algorithm. Scaling the features and guaranteeing the machine learning algorithm treats all features fairly, data scaling plays an essential role. Moreover, different scales can create a problem when doing analyses, e.g. if plotting different features in the same plot. So for visualization and interpretive data analyses, scaling is also relevant. Another reason to scale is for interpretive purposes, so that weight magnitudes are related to feature importances and feature importances can thus be investigated by looking at weight magnitudes.

Standardization and normalization are the most popular techniques in data scaling. Standardization refers to transforming the features by removing the mean and scaling it to unit-variance through dividing it to standard deviation. It can also be called Z-score normalization. Standardized value of sample  $x$  can be calculated by the following formula:

$$x_s = \frac{x - \mu}{\sigma} \quad (4.4)$$

where  $\mu$  refers to the mean of the feature and  $\sigma$  refers to the standard deviation. In general, tree based algorithms such as Random Forest and Gradient Boosting are scale-invariant. Standardization takes place independently for each feature by computing the related statistical component on the training dataset. It helps to reduce the outlier effect.

Normalization is another commonly used scaling technique which refers to transforming the features by scaling each feature to the desired range. Bounding the values between (0,1) which is a special case of min-max scaling is the typically used version. The formulation of min-max scaling is presented as:

$$x_s = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (4.5)$$

where  $x_{\min}$  and  $x_{\max}$  refers to minimum and maximum values of the feature column. It can also be performed for different intervals. Limiting the data in a small fixed range might smoothen the effect of outliers (Jiawei et al., 2011; Alshdaifat, 2014). The formula for any arbitrary interval (a,b) is presented as:

$$x_s = a + \frac{(x - x_{\min})(b - a)}{x_{\max} - x_{\min}} \quad (4.6)$$

In machine learning and statistics many real time series might include numerical features that follow a distribution far from normal. This situation may complicate the next steps of analysis (Raymaekers and Rousseeuw, 2021). The way to deal with that kind of data is by applying some transformation techniques to boost normality. Transformations toward the normal distribution help to decrease the size of outliers which make training easier. It is possible to transform the time series to follow normal distribution through power transformations like Box-Cox (Box and Cox, 1964) and Yeo-Johnson (Yeo and Johnson, 2000) which are commonly used for improving normality. Depending on the skewness of the data log or square root transformations can also be utilized. Transforming data to follow normal distribution makes it possible to achieve better machine learning performance in practice, yet it may not be possible in every occasion.

In this research, an experiment was conducted to understand the effect of feature transformations and scaling techniques. Thus, at first, no scaling and scaling through normalization and standardization applied to the original data. Later, power transformations were applied in order to obtain the Gaussian (normal, bell-curve) or Gaussian-like distribution of the features and again subjected to no scaling and scaling through normalization and standardization. At the end, machine learning models trained with these six data sets and the mean absolute errors were compared in order to decide which feature transformation and scaling technique will be selected moving forward.

#### 4.2.5 Data Split

In developing a machine learning algorithm it is a common practice to split the dataset into training, validation and test sets. It is important to identify each set separately in the context of machine learning. The training set represents the amount of data that is utilized to train the selected machine learning algorithm. It is done by learning from the historical data and estimating the parameter of the machine learning algorithm in order to predict well when the machine learning model is encountered with the data never seen before. The validation set represents the amount of data used for tuning hyperparameters.

The test set, on the other hand, is used for evaluation of the algorithm. It is only used once after the model is completely trained and validated (Lazzeri, 2020).

In data splitting it is important to consider physical principles such as hydrologic year and seasonality and statistical properties such as mean, standard deviation, minimum and maximum values in data splitting. If the model is constructed without exposing the whole range of testing data set, it is expected to have poor outcomes (Wu et al., 2008) and that's why data splitting requires special attention.

Performance of the machine learning model is directly correlated with its ability of making reliable predictions on unseen test data. Overfitting poses a threat along this way. It stems from modelling the inherent noise in the training dataset more than revealing the relationship among features and target variables. In other words, the model has learned too much redundant information from the training dataset and it fails with unseen test data. Cross-validation is one of the most used data sampling methods to train the models and to tune the hyperparameters in order to prevent overfitting. There are several cross validation methods including k-folds cross validation. In this method the training data is randomly partitioned into k folds almost equally. In each iteration (k-1) folds are used to train the machine learning model and the remaining one fold is utilized for validation. The process is repeated k times till each fold is used exactly once as validation data. The average validation error is used to describe overall performance of the model (Zhou et al., 2017; Berrar, 2018; Bi et al., 2021).

In this research, at first, the data set is planned to divide into three groups: training, validation, and test sets while trying to keep the time series nature of the data intact. Later, together with the randomized search approach which is used for the hyperparameter tuning, the cross validation method is introduced using 5-fold to improve the performance.

## 4.3 Machine Learning

The distinction in machine learning algorithms is drawn in between supervised and unsupervised learning. Although semi-supervised, and reinforcement learning also exist in the area, most articles only mention supervised and unsupervised learning algorithms. In supervised learning, it is required to use a labeled dataset that supervises the algorithm's training by revealing the underlying relationships among feature and target variables. With this supervision, the algorithm is capable of predicting the target variable when unforeseen data is presented to the algorithm. Supervised learning can be successful in overcoming real-world computational problems, widely used in regression and classification. Unlike supervised learning, unsupervised learning does not rely on labeled data yet finds patterns and similarities within unlabeled data without external supervision. This algorithm is efficient when looking for unknown relationships between observations of features. Clustering, density estimation, finding association rules, anomaly detection are most commonly used unsupervised learning tasks (Alloghani et al., 2020; Sarker, 2021).

In this research, the machine learning task is identified as a supervised learning algorithm since the training dataset contains input linked with correct output data. As suitable models it is decided to start with the linear regression algorithm due to its simplicity. Since this project is trying to predict water level and employing these predictions in flood warning and early warning systems it is important to have a machine learning algorithm that can handle the outliers. Besides, after a single model, it would be stimulating to deploy an ensemble model. In this perspective Random Forest Regression is selected as the second machine learning algorithm. In order to understand how it affects building one tree at a time by learning from the previous one instead of building each tree independently, Gradient Boosting Regression is selected as the third machine learning algorithm. Finally, a Feed Forward Neural Network is chosen to implement state-of-art machine learning algorithms in this research work.

### 4.3.1 Linear Regression

Linear regression is one of the most popular machine learning algorithms that fall under supervised learning. It is a very simple algorithm that deserves the right attention because many problems can be solved with this model, even intrinsically nonlinear ones (Bonaccorso, 2017). Linear regression is a statistical approach and performs regression tasks. It assumes a linear relationship among independent (x, one or more) and dependent (y) variables. They can also be referred to as feature and target variables, respectively. Depending on the number of independent variables, linear regression can be investigated under two main categories: Simple Linear Regression (SLR) and Multiple Linear Regression (MLR). The mathematical representation of a simple linear regression equation is presented below.

$$y = \beta_0 + \beta_1 x + \varepsilon \quad (4.7)$$

As represented in the equation 4.7, Simple Linear Regression uses one independent variable presented with x to predict the numeric value of a dependent variable presented with y. The intercepts presented with  $\beta_0$  and  $\beta_1$  and the error term presented with  $\varepsilon$ .

Multiple Linear Regression, as an extension of Simple Linear Regression, uses more than one independent variable to predict the numeric value of the dependent variable. The mathematical representation is presented as:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_n x_n + \varepsilon \quad (4.8)$$

where x represents independent variable, y represents dependent variable,  $\beta$  represents the regression coefficients and  $\varepsilon$  represents the error term. Linear regression algorithm assumes the error term (residuals) follow normal distribution (Williams et al., 2013; Sarker, 2021).

### 4.3.2 Random Forest

Random Forest is another most prevalent supervised learning algorithm based on building decision trees with ensemble methods. In decision trees, a single tree is not accurate by itself because it suffers from a bias-variance trade-off. In other words, decision trees are able to grow a complex model with low bias, yet they would change a lot with training on different data. In order to keep low bias trees and reduce this high variance associated with prediction, building ensemble models comes into picture. Ensemble model combines predictions from different machine learning algorithms and takes average to reach reduced variance ergo more accurate predictions than any individual algorithm can give. However, it is not possible to build ensemble models for Decision Trees using different training sets, since there is only one training set. The way to get multiple training sets from a single one is using a statistical technique called bootstrapping. Bootstrapping refers to resampling the training set with replacement. Replacement is an utterly critical task because no replacement would lead identical samples of the training set. By resampling some observations in the original training dataset might be repeated in the new ones. Combination of bootstrap with Decision Trees is called Bagging. Bagging (or Boosting AGGregatING) is the process of obtaining bootstrapped samples from the original training set, building low biased - high variance decision trees for each of the obtained samples and finally aggregating predictions from all of these trees created. If the task is regression, aggregating means taking the average of predictions acquired from the trees. On the other hand, in classification tasks, aggregating means taking the majority vote. Although averaging the predictions reduces the variance for the final prediction, if the individual predictions are highly related with each other, taking the average would not make any difference and all the effort would be wasted. In bagging, trees incline to look similar to each other which causes generation of related predictions. In order to avoid these predictions, Random Forest was introduced by Breiman in 2001. The basic idea is decorrelating the trees generated through Bagging by forcing every tree to use a random subset of features during the split. In this sense, the Random Forest algorithm follows the same procedure in Bagging with a small deviation. The Random Forest algorithm can produce accurate results, have low sensitivity to multicollinearity, relatively stable for missing data, outliers and noise.. As a limitation, high demand in terms of time and computational resources can be shown. Compared to Decision Trees, Random Forests are more resistant to overfitting, yet this can still be a problem (Efron and Tibshirani, 1993; Breiman, 2001; Hastie et al., 2001; Sutton, 2005; Zhang and Lu, 2012; Prasad, 2006; Chen et al., 2020).

### 4.3.3 Gradient Boosting

The boosting concept is originating from the idea of boosting the accuracy of models with limited performance (i.e. weak learners) by correcting the predecessor model's prediction sequentially in an ensemble model. Originally the boosting algorithm was presented by Kearns in 1988 and over time the popularity accelerated. The evolution of boosting algorithms is explained in general terms by Mayr et al. in 2014. The algorithm itself was introduced in 1999 by Friedman. The learning in the Gradient Boosting algorithm happens by optimizing the loss function through the steepest gradient descent. There are also other loss functions, yet the gradient descent is one of the most

popular algorithms to perform optimization (Ruder, 2016). It uses the loss function of the predecessor model as an input to the next model and the procedure goes on in this sequence till either the loss function reaches zero or the stopping criteria are met.

Gradient Boosting algorithm shares some common properties with Random Forest algorithm like both algorithms are based on ensemble learnings and are using decision trees as the weak learner. The difference stems from how trees are built and the aggregation of the results. In the Gradient Boosting algorithm, decision trees are built one at a time in order to improve pitfalls and optimize advantages from the predecessor model. On the other hand, in the Random Forest algorithm, decision trees are built independently and the results are aggregated at the end of the process by averaging the predictions for regression and taking the majority vote of the predictions for classification tasks as described in Chapter 4.3.2. The Gradient Boosting algorithm aggregates the results en route. In addition to the differences, the Gradient Boosting can use other weak learners than decision tree (Papacharalampous et al., 2019)

#### **4.3.4 Artificial Neural Network**

The artificial neural networks are mathematical models that use nonlinear computational methods inspired by the functioning of the biological brain and the nervous system. The network takes an input and passes it through multiple layers of neurons and produces the output. A neuron is a basic unit in an artificial neural network for computation that resembles the biological neurons. Each neuron in the neural network receives the multiple weighted inputs through synaptic connections, calculates the weighted sum, applies either linear or nonlinear activation function and returns it to the next layer. The input to a neuron can be directly coming from the training set as a feature or the previous layer's output as well. The artificial neural networks are composed of three layers, namely, input, hidden, and output. Based on the model complexity there can be several hidden layers (Sazli, 2006).

One of the main factors that affects the performance of the artificial neural networks in the selection of activation function which introduces the nonlinearity to the neural network. This function determines whether a neuron should be activated or not after calculating weighted sum and further adding bias with it. There are several activation functions such as Sigmoid, ReLu (rectified linear unit), Leaky ReLu, ELU, tanh, softmax, softplus, linear that influence the network's prediction ability (Chollet, 2015; Feng and Lu, 2019).

Depending on the type of connections, artificial neural networks divide into two main categories, Feed Forward Neural Networks and Recurrent Neural Networks. If the flow of the information is only in forward direction, with no feedback from the output neuron to the input neuron, it is called Feed Forward Neural Network. On the other hand, if there is such feedback from the output neuron to the input neuron, it is called Recurrent Neural Network. Feed Forward Neural Networks can be divided further into two categories depending on the number of layers, either single-layer or multi-layer. In the

single-layer Feed Forward Neural Networks there are no hidden layers. Since there is no computation performed in the input layer, these networks are called single-layer. On the other hand in multi-layer Feed Forward Neural Networks there is at least one hidden layer in between input and the output layers (Sazli, 2006).

In this research, a Feed Forward Neural Network is harnessed to forecast water level in the Stora River. In building the model TensorFlow library which was utilized. TensorFlow is an open-source software library which can be employed across large-scale heterogeneous systems. In order to enable fast experimentation, Keras was developed. Keras is a deep learning application programming interface, running on top of the machine learning platform TensorFlow. There are two model types available in Keras, the Sequential model and the Functional API. The Sequential model represents a linear stack of layers which makes it comfortable to utilize in building vanilla Feed Forward Neural Networks. Functional API, on the other hand, can be used in building more complex models (Abadi et al., 2015; Chollet, 2015). The Sequential model is employed for water level forecasting. As an optimizer Adam is selected. Adam is a gradient descent optimization algorithm which is known as one of the most popular optimizers and commonly implemented in neural networks. It is experimentally proven that the Adam optimizer is faster than any other optimizers (Kingma and Ba, 2015; Bock et al., 2018). As a loss function the mean absolute error is used in order to maintain consistency in the research and found out as the recommended loss function (Qi et al., 2020; Jierula et al., 2021). Moreover early stopping criteria which monitor the validation loss is introduced in order to prevent overfitting. The early stopping mechanism monitors validation loss after each training cycle, and when the validation result stops improving it finalizes the training depending on the predefined patience value. For activation function, batch size, epochs, and learning rate several inputs were used in order to tune hyperparameters through the random search method.

#### **4.3.5 Evaluation Criteria**

There are several different statistical measures that have been adopted in evaluation of machine learning algorithms including coefficient of determination ( $R^2$ ), root mean square error (RMSE), mean absolute error (MAE), mean square error (MSE), correlation coefficient (CC), mean absolute percentage error (MAPE) etc. Using various statistical measures instead of a single criterion is required in order to evaluate different aspects of the model performance (Richter et al., 2011; Cheng et al., 2016). RMSE and MAE are frequently used evaluation criteria in the field of hydrology. These metrics measure the efficiency for the machine learning models in the same unit as the target variable. This usually provides more information about the efficiency of the machine learning model than relative errors or goodness-of-fit measures. Either RMSE or MAE are typically recommended to use as absolute error indicators, yet it is better to use them both since the degree to which RMSE exceeds MAE indicates the extent of the outliers (Legates and McCabe, 1999; Harmel et al., 2014). On the other hand, the accuracy of machine learning models are reporting commonly through coefficient of determination ( $R^2$ ) which allow comparability in the literature (Conrads and Roehl, 2007; Richter et al., 2011).

In this research coefficient of determination ( $R^2$ ), root mean square error (RMSE), mean absolute error (MAE) are selected as the evaluation criteria. The mathematical formulations of the  $R^2$  for observed and predicted values is presented as:

$$R^2 = 1 - \frac{RSS}{TSS} \quad (4.9)$$

where RSS represents the sum of squares of residuals which is a calculated squared sum of observed value subtracted by the predicted value. TSS stands for total sum of squares which is a calculated squared sum of a value subtracted by the observed value.

The mathematical formulations of the RMSE is presented as:

$$RMSE = \sqrt{\sum_{i=1}^n \frac{(\hat{y}_i - y_i)^2}{n}} \quad (4.10)$$

where  $\hat{y}_i$  represents the predicted variables whereas  $y_i$  represents the observed values and  $n$  stands for number of observations. The mathematical formulations of the MAE is presented as:

$$MAE = \frac{1}{n} \sum_{i=1}^n |\hat{y}_i - y_i| \quad (4.11)$$

### 4.3.6 Improving Models

There are various ways to improve machine learning models' performance. One of them is introducing more data to machine learning models. The machine learning models require a large amount of true training data. That's why in case the available data increases, machine learning predictions get better, as it allows the data to tell itself rather than depending on assumptions and weak correlations (Hughes, 2019).

Another way of improving the machine learning model's predictive performance is improving the quality of data. There are multiple factors that influence the data quality such as consistency, completeness, accuracy, integrity, presence of outliers, dimensionality reduction, feature selection etc. (Gudivada et al., 2017). Poor data quality can be a threat for the machine learning model's predictions.

Using k-fold cross-validation sets also helps to improve the performance of the models by reducing risk of overfitting to training data and improving chances of good performance on unseen data. In k-fold cross validation, the data is split into approximately equal size of k folds and the machine learning model is trained k times in everytime leaving one fold out to use in computing the prediction error (Borra and Di Ciaccio, 2010; Asrol et al., 2021).

Hyperparameter tuning can also be included in this list. Machine learning models consist of various hyperparameters that define characteristics of a model and help to improve the performance of it for any given problem. Finding the best combination of these parameters might be a challenging task. There are mainly two kinds of hyperparameter tuning techniques: manual search and automatic search. Manual search heavily depends on intuition and experience of the expert since it consists of trying the hyperparameter sets by hand. This technique requires professional knowledge and practical expertise which makes it harder to be used by laymen. In order to overcome this difficulty automatic search techniques have been proposed. Grid Search and Random Search are the most pronounced methods. In grid search algorithm hyperparameter optimization is performed through exhaustive searching which refers to training the machine learning model with every possible combination of values of hyperparameters on the training set and evaluating the performance on the validation set. Efficiency of this algorithm depends on the number of hyperparameters being tuned and their range of the values, the more the values lower the efficiency. This problem is solved through the random search algorithm. According to this algorithm, the hyperparameter optimization practice is performed on only a few hyperparameters that matter in a random combination of ranges. Compared to the grid search algorithm, random search is more effective in a high-dimensional space (Wu et al. 2019)

In this research random search is applied together with k-fold cross validation using RandomizeSearchCV in the python environment to tune the hyperparameters. 10 iterations and 5 fold is selected due to time constraints. Iterations refers to the number of parameter settings that are sampled through param\_distribution. As scoring more than one metrics is used defined as in the evaluation criteria and as refit the mean absolute

error is assigned. Refit allows to return the best estimator from the random search and predict directly on this RandomizedSearchCV instance (SKlearn, n.d.).

Random Forest Regression is able to grow very complex decision trees. As a result of this, it is expected to have overfitting in the training set. In order to prevent this problem some hyperparameters were used to decrease complexity. Selected hyperparameters to be tuned in this research are `max_depth`, `max_features`, `n_estimators`, `ccp_alpha`, and `min_samples_split` for the Random Forest Regression model. `Max_depth` represents the depth of each tree in the Random Forest. Each decision tree is able to grow to the largest extent possible. The deeper the tree gets, it gathers more information, the model becomes more complex, and overfitting risk increases. Tuning the `max_depth` parameter and finding the optimum value help to reduce the growth of trees. `Max_features` represents the number of random feature subsets considered before the best split. `N_estimators` represent the number of trees in the forest. Random Forest algorithm consists of multiple decision trees. Increasing the number of the trees helps decrease the bias as explained in chapter 4.3.3. However, it is not possible to keep increasing these trees due to time and resource constraints. In this context, finding an optimum `n_estimators` is required. Another way of controlling the size of a tree by pruning the nodes by `ccp_alpha`. High values might cause information loss, on the other hand, low values may not prevent overfitting. `Min_samples_split` represents the minimum number of samples needed for split. If the minimum number of samples cannot be guaranteed, training will stop.

Gradient Boosting Regression is also a tree based ensemble learning algorithm, thus it shares several hyperparameters with Random Forest, especially tree related ones. In this research, `max_depth`, `min_samples_split`, `max_features` and `learning_rate` were used as hyperparameters. Apart from the latter, the same parameters are used and explained for the Random Forest model. `Learning_rate` refers to how fast the error is corrected from one decision tree to another. Lower values would make the Gradient Boosting model more robust due to giving more time for the model to learn, yet lowering the learning rate requires more decision trees which would increase computation time and increase the chance of overfitting.

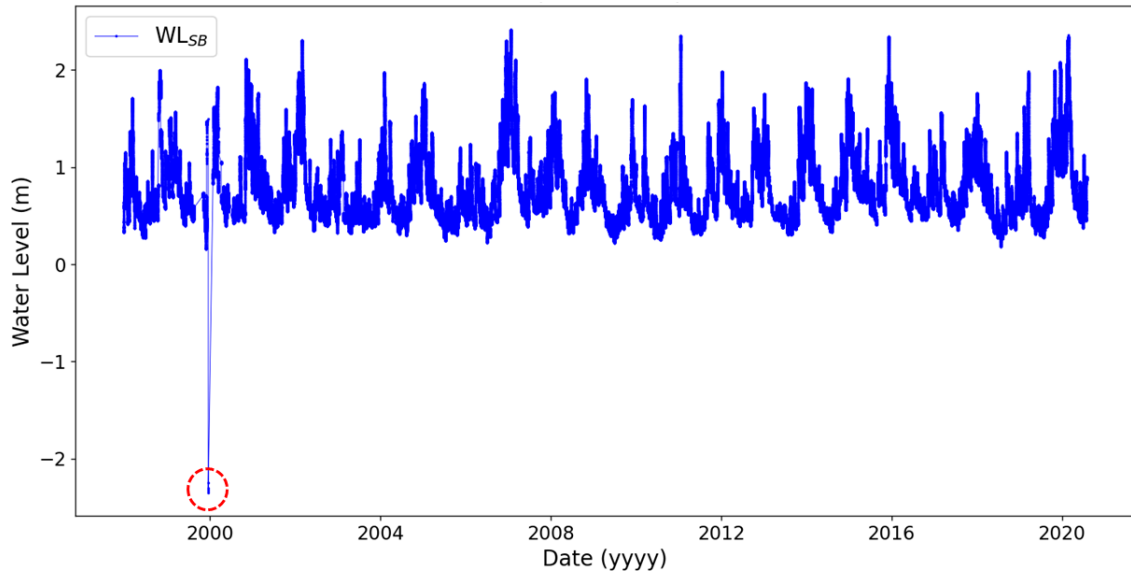
## Chapter 5. Data Analysis

*This chapter, the data was prepared to be used in the machine learning algorithms. At first data analysis results were presented for data visualization and preprocessing through missing data imputation, outlier removal, format modification, and duplicated index elimination. Thereafter, Pearson's correlation and mutual information analyses were applied to preprocessed data in order to select the most relevant features that would fulfill the research objectives. After the feature selection process, the persistence model was introduced. The feature sets were then created based on the outcome of the filter methods. Afterwards the data split was done by considering the limitations on the forecasted precipitation data. Finally, feature transformation and scaling techniques were applied.*

### 5.1 Data Visualization and Preprocessing

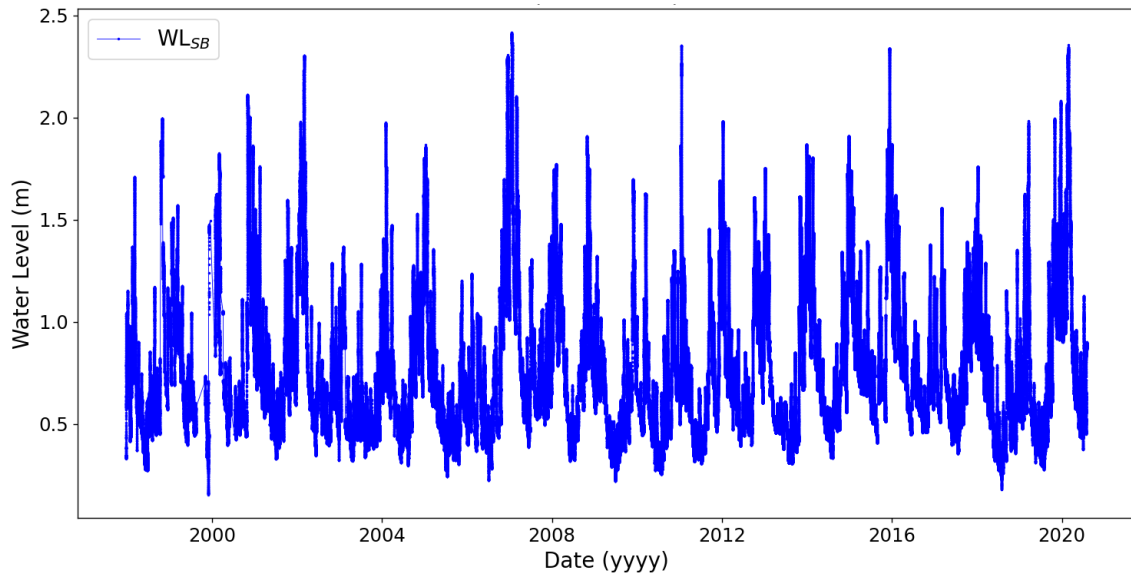
The Python environment is utilized for data analysis in this research. Python is one of the most frequently used programming languages in both data analysis and machine learning. For data analysis, Pandas library is used. It stands for "Python Data Analysis Library" and is widely used in data analysis and manipulation due to its convenience in working with tabulated data. Together with Matplotlib and Seaborn libraries, Pandas offers a great variety of visualization for tabulated data. Other than aforementioned libraries, NumPy, SciPy, Scikit-Learn, were also utilized moving forward.

After the objective was defined, required data was collected. Data analysis started with the water level parameter at Skærum Bro and Ellebæk Bro stations. There was no redundant data in the datasets; they only consist of water level measurement recorded in meters by 15-minute time intervals. Note that water level measurements included elevation of the river bed in the Storå River. Following the initial visualization, five data points were detected as outliers for Skærum Bro Station, presented with the red circle in Figure 10.



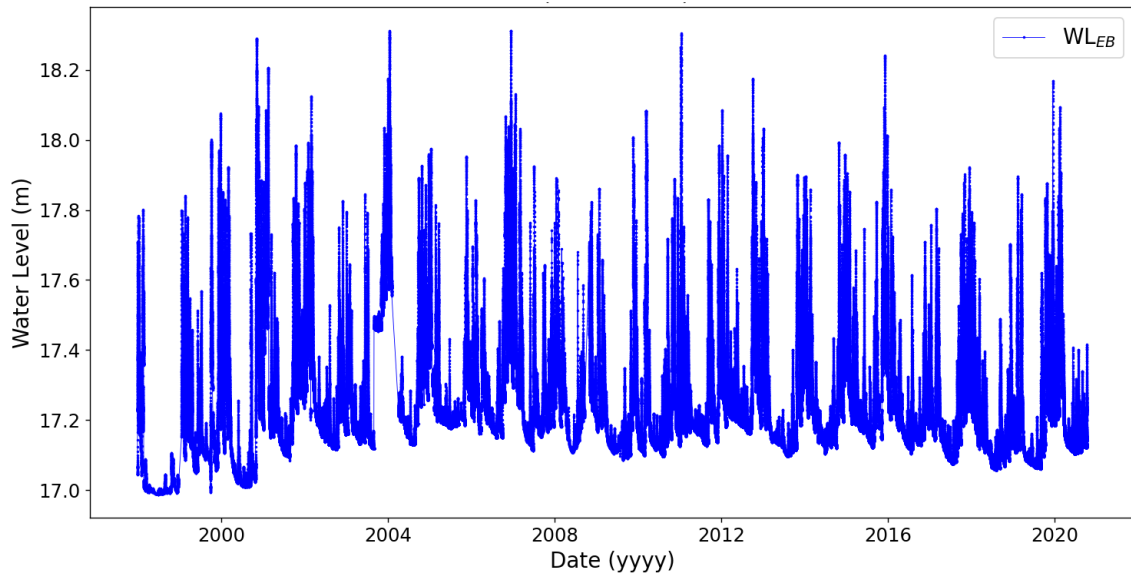
**Figure 10** | *Storå River water level measurements at Skærum Bro Station by 15-min intervals with detected outliers (1997 - 2020).*

Outlier removal was performed for these five data points. In order not to lose peak value information, further outlier removal was not performed. The datasets are presented in Figure 11 and 12 for water level stations at Skærum Bro and Ellebæk Bro, respectively. The minimum water level recorded in Storå River was 0.157m on 1999-11-26 at 22:00:00 and the maximum water level was recorded as 2.411m on 2007-01-21 at 10:00:00. The water level in the Storå River was remarkably variable throughout a year. Yearly difference between maximum and minimum water level varies between 1.045m and 2.074m. In general maximum water levels were observed in winter season whereas the minimum water levels were observed in summer season.



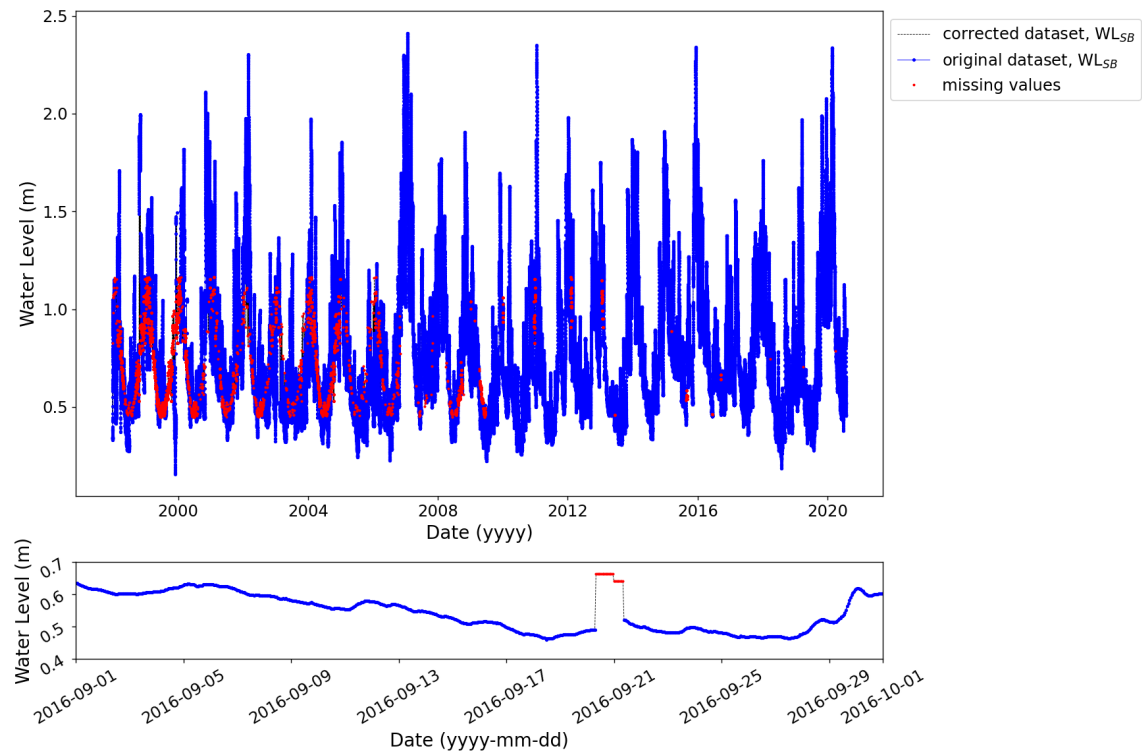
**Figure 11** | *Storå River water level measurements at Skærum Bro Station by 15-min intervals after removal of outliers (1997 - 2020).*

The water level in Ellebæk Brook also showed variation throughout a year, yet this variation was smaller than Storå River. The average difference in maximum and minimum water levels for Ellebæk Brook was 0.970 m. On the other hand, it was 1.600 m for the Storå River. There might be several reasons behind this difference. First, Ellebæk Brook is a much smaller stream than Storå River. The variation in Ellebæk Brook is only about the stream itself. However, Storå River is fed by various different small or big water sources and variation in those might create a cascade effect for Storå River. Another reason is the difference in deposition of sediments for both water bodies. Deposition refers to the settlement of material being transported inside the river and it occurs when the river loses its energy. This effect is mostly observed in the mouth of a river where the journey of a river ends and the energy drops. In this perspective, Skærum Bro Station in terms of the location is more prone to deposition than Ellebæk Bro Station. Some factors like seasonal changes, climate variability, or human activities can affect the amount of deposition thus, water level variations. Other than aforementioned reasons, change in biomass and some anthropological factors might help to explain the recorded water level variation differences for Skærum Bro and Ellebæk Bro stations.



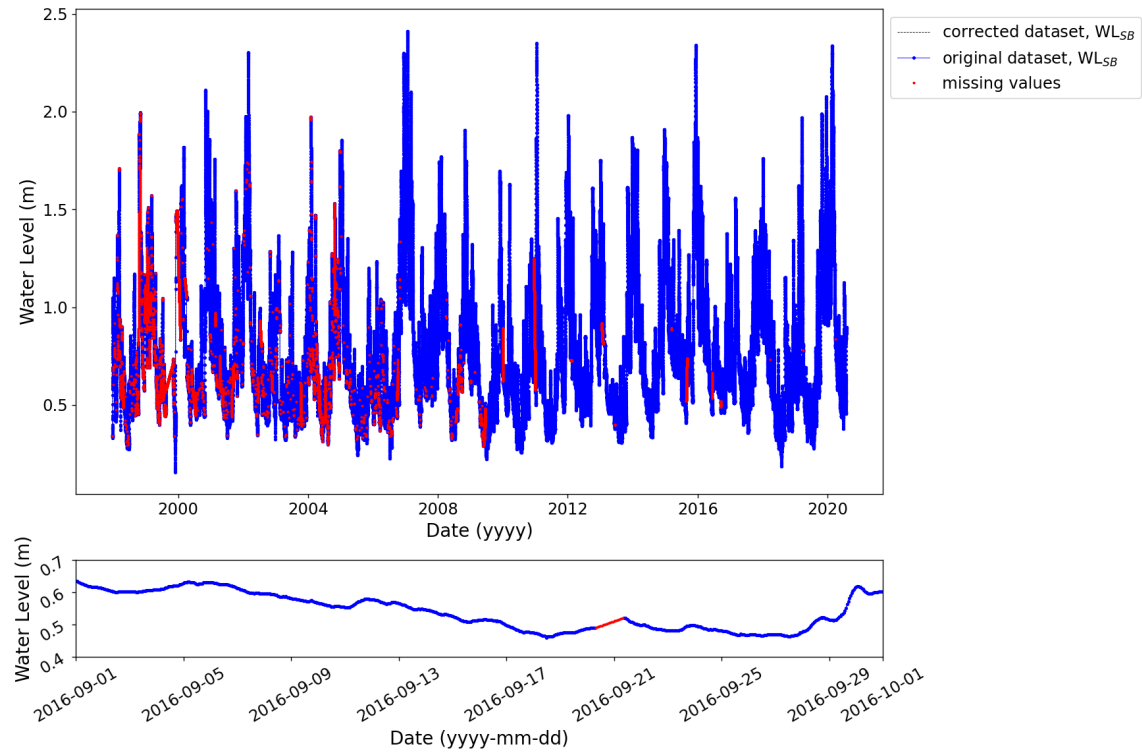
**Figure 12** | *Ellebæk Brook water level measurements at Ellebæk Bro Station by 15-min intervals (1997 - 2020).*

To make the dataset consistent with other variables in terms of frequency, resampling the water level data over one hour was performed by calculating the average water level in an hour for both Skærum Bro and Ellebæk Bro stations. During the missing data control, 14,661 data points were detected in the Skærum Bro water level dataset. In imputing missing values some fast techniques were considered as discussed in theory in section 4.2.2. Although these fast techniques could achieve filling in the data gaps reasonably, it was foreseen that the techniques would create abrupt changes and interfere with continuity of the time series, thus mis-presenting the physical process behind the water flowing through the stream. To illustrate the validity of this argument, the missing hourly values were filled with the daily mean water level value derived from yearly mean values between 1997 to 2020 and the results are presented in Figure 13. Zoomed representation introduced in the second plot shows an interruption in the water level measurement presented with the blue line for two days of missing value gap. An artificially created peak by the mean value imputation technique justifies the disqualifications of the aforementioned techniques moving forward.



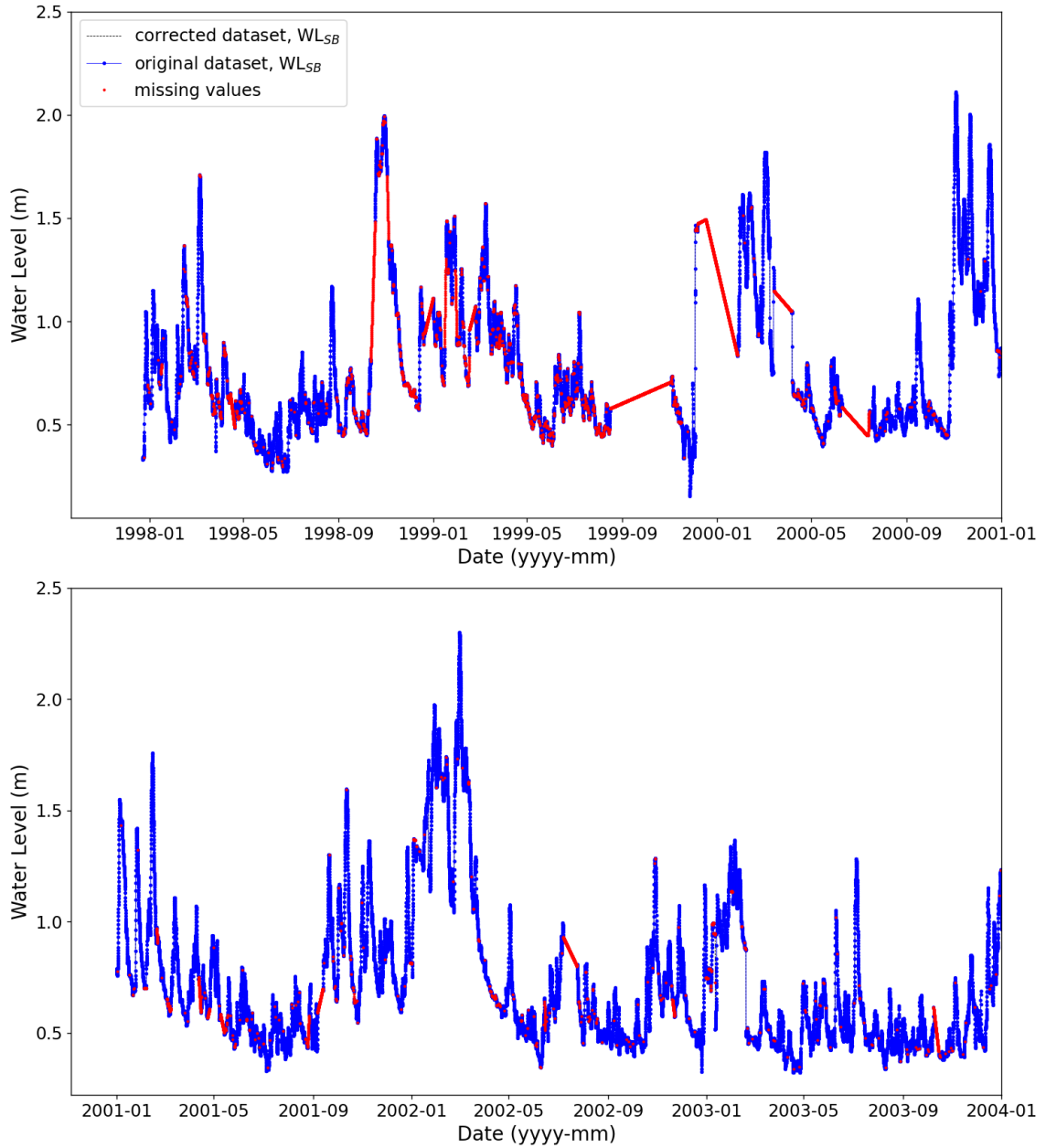
**Figure 13** | *Missing value imputation using the mean value technique for Storå River hourly water level measurements at Skærum Bro Station (1997 - 2020).*

In order to avoid the problems described preceding, the linear interpolation technique was used in imputing missing water level values. This technique imputes the missing data by connecting dots as a straight line, thus, no discrepancy would occur. The corrected dataset for Skærum Bro Station based on this intervention is presented in Figure 14. In the zoomed plot, the same time period is presented as above to illustrate the ability of this imputation technique to capture the physical process of the water flow.



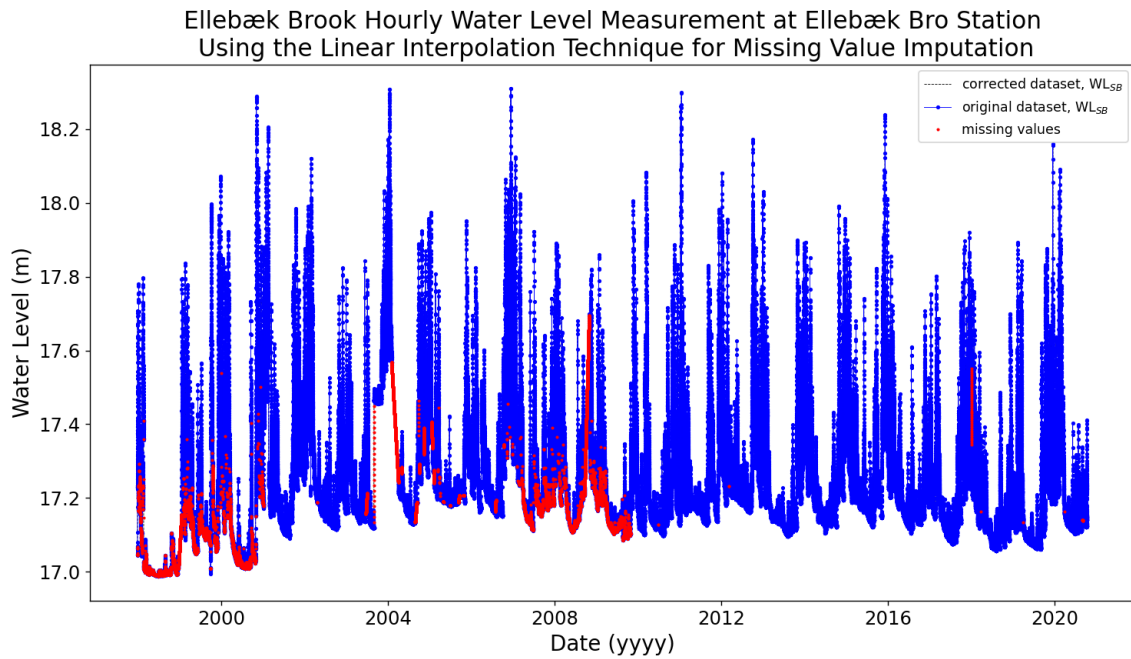
**Figure 14** | *Missing value imputation using the linear interpolation technique for Storå River hourly water level measurements at Skærum Bro Station (1997 - 2020).*

Worth to mention, missing data in approximately the first quarter of the time series was quite significant, it can be understood how dense the red dots are, and interpolation of these gaps created unrealistic results as presented in Figure 15. Specifically, water level values in October, 1999 and January, 2000 may not be accurate due to the large gap in the dataset. In fact, intuitively it was expected to see some peaks for those periods considering the behavior of water level around these months in the whole dataset. Therefore, even linear interpolation technique was not the best fit for missing data imputation for water level in the Storå River. However, the starting dates across different parameters' time series varies as presented in Table 1, and 2013 was the limiting start date for all parameters. Effectively, the unrealistic missing data imputation for the water level at Skærum Bro Station does not create a problem for the dates before 2013. Thus continuing with linear interpolation technique was practicable for this case for the years in between 2013 and 2020.



**Figure 15** | *Missing value imputation using the linear interpolation technique for Storå River hourly water level measurements at Skærum Bro Station.*

Missing value imputation for water level from Ellebæk Bro Station was pursued using the same technique. When the concentration of red dots was analyzed, there was a significant amount of data loss at the beginning of the time series and around 2008. Linear interpolation technique again created some unrealistic results for those periods. However, as explained above it is not a concern for this research. For this data set, the missing value imputation with linear interpolation technique can be used for the years in between 2013 and 2020.



**Figure 16** | *Missing value imputation using the linear interpolation technique for Ellebæk Brook hourly water level measurements at Ellebæk Bro Station (1997 - 2020).*

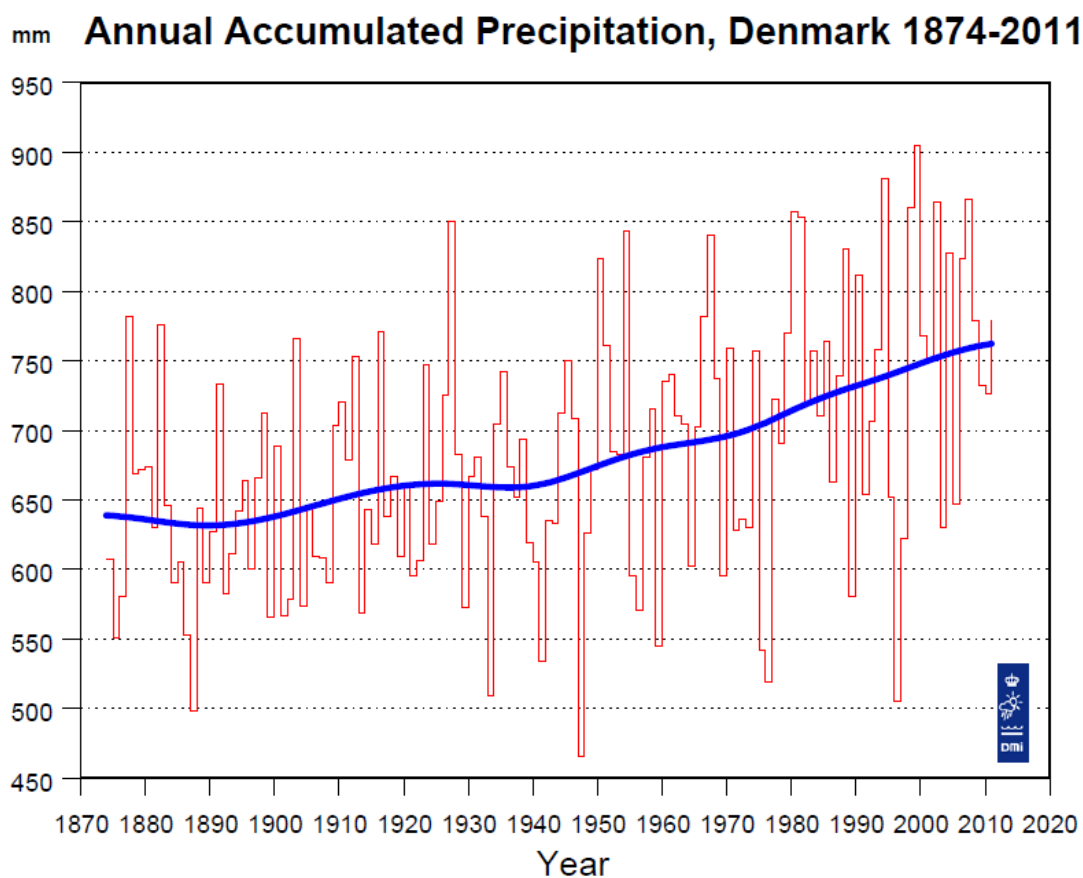
It was moved forward with precipitation data analysis. The historical precipitation dataset includes five different categories demonstrated in columns. Only data in columns 'timeObserved', 'stationID', and 'value' were kept; the rest were found redundant and removed immediately. The 'timeObserved' column presents the date of data measured in unix timestamp. Storing data in unix timestamp format is handy for computer systems because it eliminates confusion over timezones. However, it is not easy for a human being to understand the time from a big integer while there is a more communicable way. To put it another way, "July 12, 2021 2:46:04 PM Central European Summer Time" is much more understandable than the unix timestamp version, 1626101164. That's why the unix timestamp in 'timeObserved' was converted to datetime. The 'stationID' column consists of five different precipitation stations: Isenvad, Grønbjerg, Øby, Høgild, and Gludsted Plantage Nv. The data was reformatted by grouping stations separately. The 'value' column presents the precipitation value in millimeters (mm), and the column was renamed according to the abovementioned precipitation station's name.

Initial visualization of each station can be found in Figure 17 for the raw data. Despite the preprocessing done by the organisation there was still need for further processing because of gaps and erroneous information in the data sets as presented in the Figure 17.



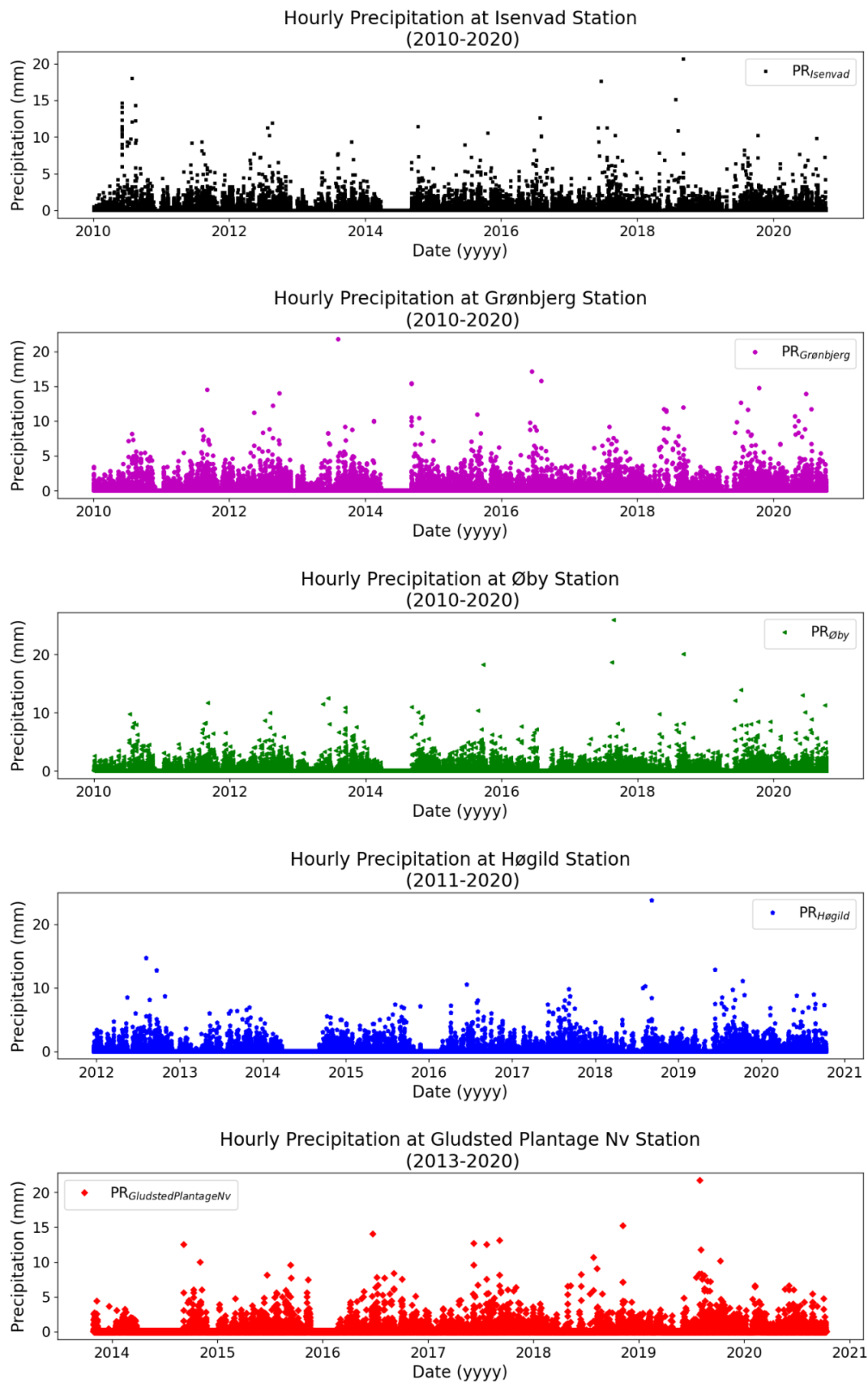
**Figure 17** | *Historical precipitation data for given precipitation stations*

Missing data control concluded with a total of 82,022 data points for Isenvad, 80,619 data points for Grønbjerg, 81,867 data points for Øby, 66931 for Høgild, and 52703 for Gludsted Plantage Nv which contributes 87%, 85%, 87%, 87%, and %86 of data missing among the whole dataset, respectively. As described in Chapter 4.2.2 imputation of missing data, especially precipitation data is quite a difficult task due to the spatial and temporal nature of precipitation. Nevertheless, some attempts in imputing missing values using mean, minimum and zero values have been executed. Using mean and minimum values ignored the dry day concept completely and caused annual accumulated precipitation up to 7000 mm. When annual accumulated precipitation values taken from DMI's technical report for 1874-2011 presented in Figure 18 were considered as a reference, 7000 mm was approximately ten times greater than the actual annual accumulated precipitation values in Denmark. Thus, these imputing techniques were impractical.



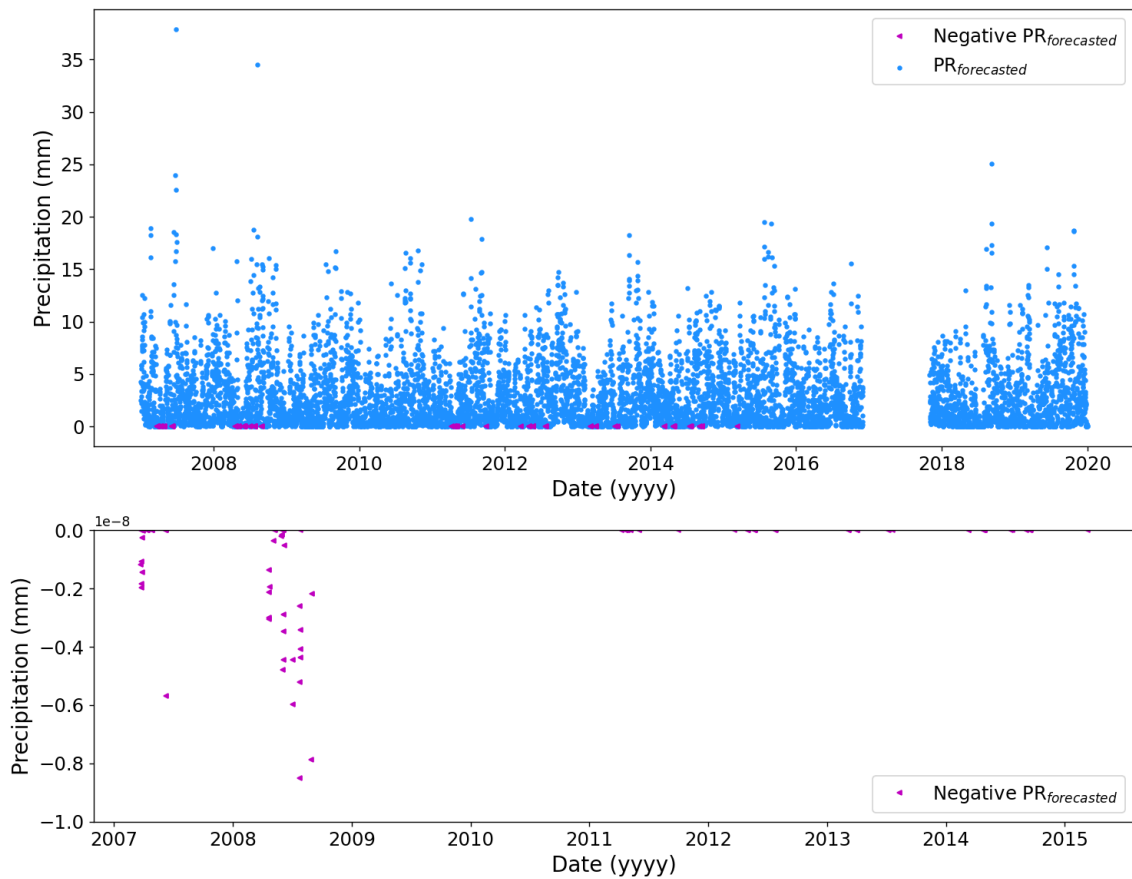
**Figure 18** | *Annual accumulated precipitation, Denmark (1874 - 2011)*

On the other hand, imputing missing values with zero gave acceptable annual precipitation. Although filling missing data with zero does not accurately represent the reality, especially for years 2014 and 2016 where significant gaps were displayed in the data, it was determined to proceed with this technique, considering the limitations and the scope of this project. Apart from gaps, some outliers were detected for the stations Øby and Høgild. Øby Station recorded 70.6 mm precipitation on 2016-10-13 at 08:00:00. Examining the preceding and following precipitation values together with statistics of the time series this value was assigned as outlier and removed. In the same way, an outlier is detected in Høgild Station with 407 mm precipitation on 2020-08-16 at 11:00:00 and removed. For the other stations there were no obvious outliers. The removal of outlier data was conducted after careful consideration such that peak precipitation data was not lost in the process. In duplicated index check, multiple data points were detected for every precipitation station and cleaned from the datasets. The final version is presented in Figure 19.



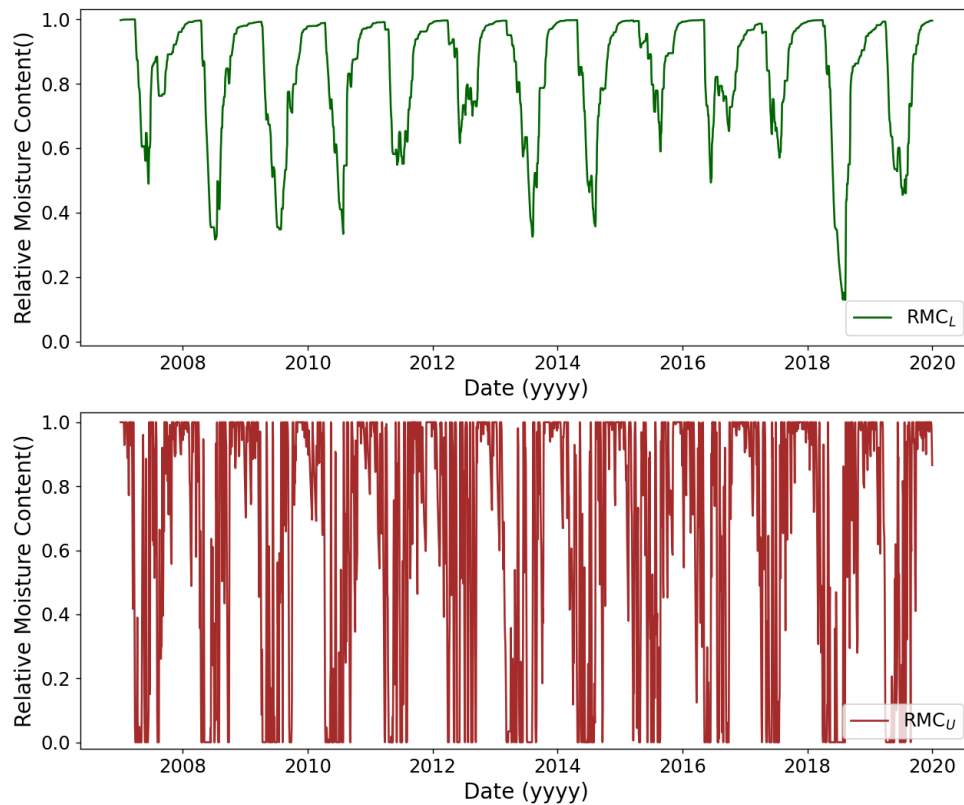
**Figure 19** | *Historical precipitation data for give precipitation stations after missing value imputation*

The ECMWF precipitation forecast data was downloaded through the python environment in GRIB data format. It is a file format for gridded data in order to store and transport the information. This format is widely used for meteorological data. However, all the other data collected by the given stations was arranged in points spatially. Thus, conversion was required for forecasted precipitation data. For this purpose, a tool called Panoply developed by NASA was used in conversion. Panoply allows a user to visualize the stored data in the GRIB file as well as extract the data in the form of a map and tabular data. After the data was extracted in tabular format some preprocessing was required. Precipitation value is assigned in every corner of the gridded data thus, first the sum of these total precipitation values were calculated to have one value that represents the whole grid. After that, the time series index was created manually to resolve the difference in downloaded time steps. Afterwards, the negative data points were detected as visualized in Figure 20, and replaced with zero.



**Figure 20** | *Forecasted precipitation data (2007 - 2019).*

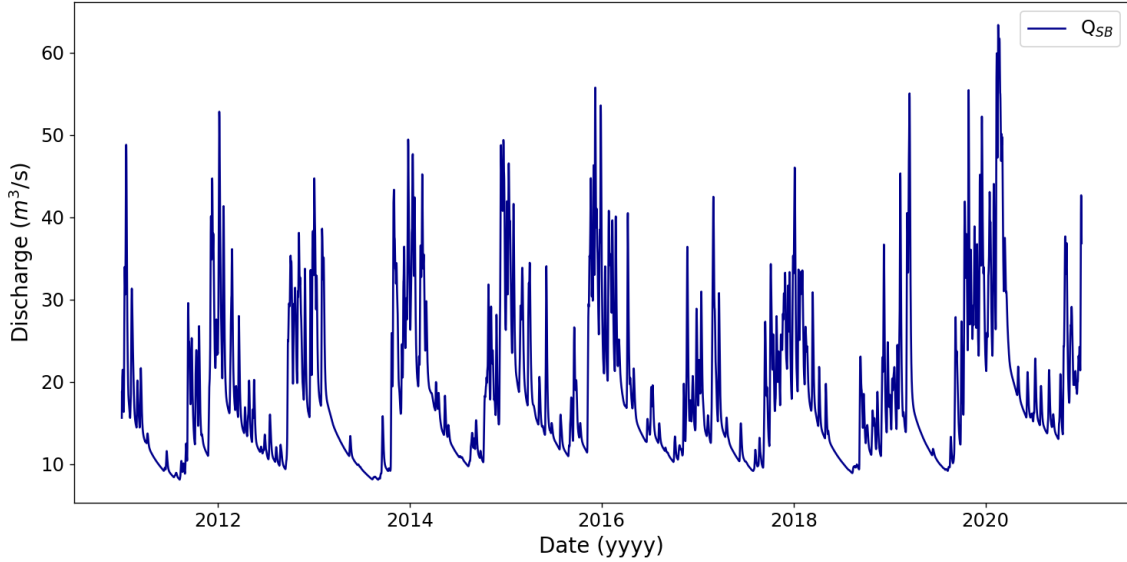
Relative moisture content came from simulation results from the MIKE 11-NAM models. Therefore, no further processing was applied to this data set. L represents the moisture in the root zone whereas U represents top soil moisture. Figure 21 presents the soil moisture content for the aforementioned soil zones. The deeper location of the root zone soil moisture was reflected in the longer duration between maximum of 0.9997 and minimum of 0.1283 relative moisture content measurements. In contrast to the root zone measurements, the top soil measurement shows more frequency cyclical durations showing that the relative moisture content values are heavily influenced by factors such as evapotranspiration and direct infiltration. Root zone was inclined to have high soil moisture content at the end of winter - beginning of spring seasons and low soil moisture content during summer season.



**Figure 21** | *Relative soil moisture content simulation at Skærum Bro Station by 2-day intervals (2007 - 2019).*

The discharge time series presented in Figure 22 was derived from an hydrologic model for the area. Similar to the relative moisture content data, the discharge data was a simulation result, and no further data and time series processing was applied to the data set. The maximum discharge value of 63.36 m<sup>3</sup>/s was simulated to occur in 2020-02-18 at 12:00:00 while the minimum discharge was simulated with 8.08 m<sup>3</sup>/s during 2013-09-02 at 12:00:00 among the whole dataset. When it comes to seasonal analysis,

the maximum discharge was simulated to occur during the winter season while a minimum discharge of the same year occurs during the fall season for the whole dataset, except in 2019, maximum discharge was simulated at the end of fall season. The seasonal variation in hydrometeorological characteristics of the study area is thus reflected in the discharge time series presented in Figure 22.



**Figure 22** | *Storå River discharge simulation at Skærum Bro Station by 12-hour intervals (2011 - 2021).*

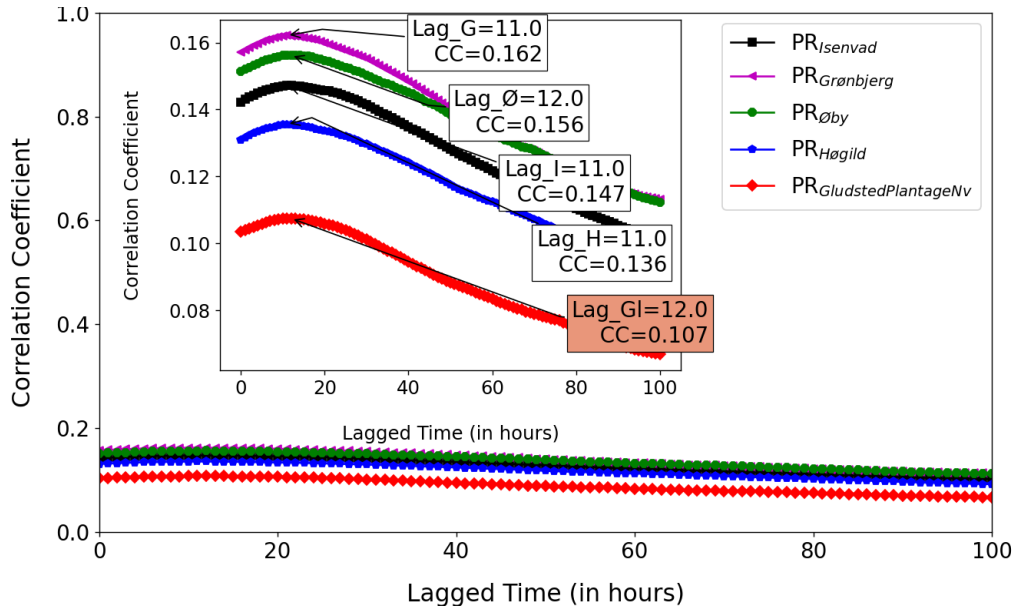
## 5.2 Feature Selection

In feature selection filter methods were utilized in this part. It was planned to use the wrapper method in addition to the filter methods after machine learning models start to train. The application of hybrid method will be discussed in the next chapter.

### 5.2.1 Correlation Analysis

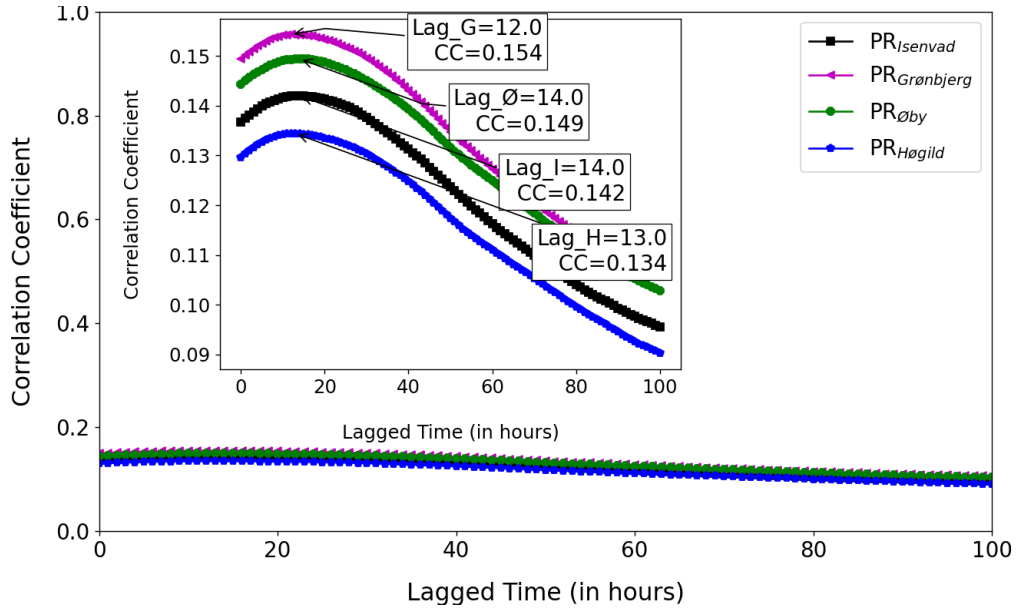
Pearson's correlation coefficient was utilized for feature selection in predicting the target variable of  $WL_{SB}(t+48)$ . The analysis was conducted through lagging the time series to analyse the relationship and sequential dependencies of the target variable and potential features. The lag refers to shifting the time series. The goal was to identify optimal lag which represents the largest absolute value of Pearson's correlation coefficient. 100 was selected as an arbitrary number for lagging due to its repetitiveness in the literature. Having absolute value of the correlation results closer 1 means the feature has potential to be useful for predictive purposes in machine learning.

In terms of data availability, limiting factors were defined as the observed precipitation data for the start and simulated relative soil moisture and forecasted precipitation data for the end (Table 1). Thus, time-lagged cross correlation analyses among  $WL_{SB}(t+48)$  and precipitation values from Isenvad, Grønbjerg, Øby, Høgild, and Gludsted Plantage Nv stations were conducted separately. The combined results are presented in Figure 23. According to the visualization, Grønbjerg Station resulted in the highest correlation coefficient with 0.162 at 11 hours prior shift. The stations Øby, Isenvad, Høgild, and Gludsted Plantage Nv were ordered by decreasing correlation coefficients of 0.156, 0.147, 0.136, and 0.107 with a lag of 12, 11, 11, and 12 hours, respectively. It was revealed that Gludsted Plantage Nv Station has relatively lower correlation than others and can be removed in moving forward.



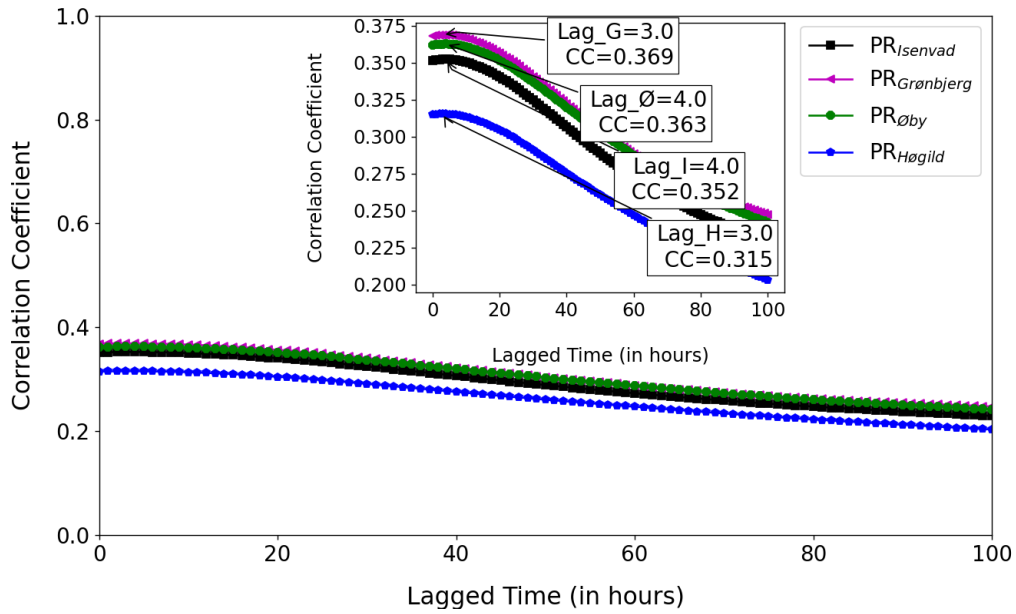
**Figure 23** | Correlation analysis between water level at Skærum Bro ( $WL_{SB}(t+48)$ ) and given precipitation stations.

After the removal of Gludsted Plantage Nv Station, the common starting date was updated to 20-12-2011. The lag-time correlation analysis was repeated for each station one more time, separately. Visualization of the correlation coefficients and lag times for precipitation stations with the revised time series is presented in Figure 24. The correlation results were slightly decreased for each station. However, considering the enlarged time length gives more information that can be deployed by machine learning algorithms, it was decided to continue with an updated time interval. Moreover, the correlation results for precipitation stations either for enlarged time length or for the original length can be considered as weak or poor correlation.



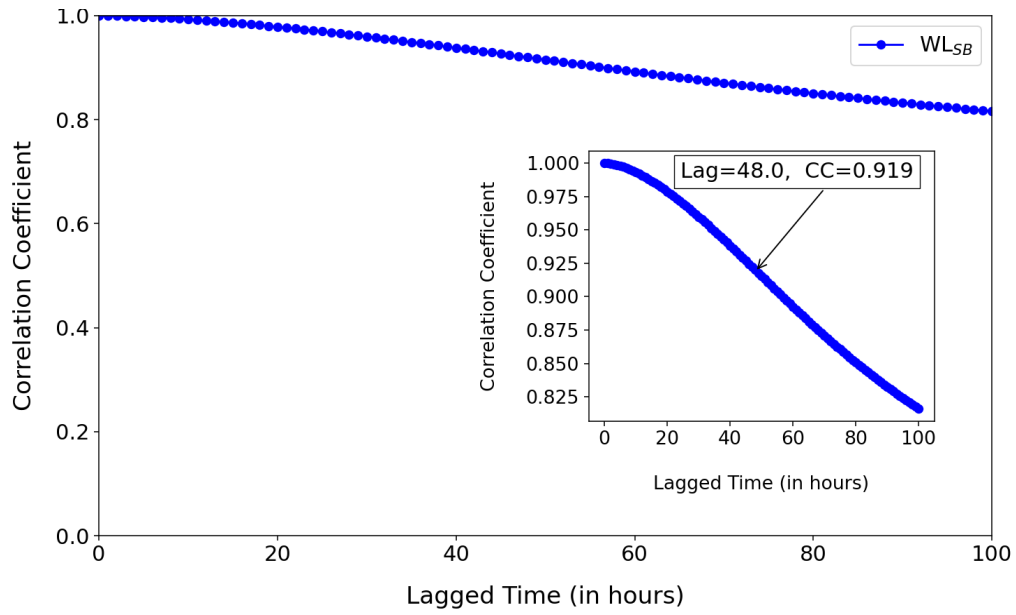
**Figure 24** | Correlation analysis between water level Skærum Bro ( $WL_{SB}(t+48)$ ) and given precipitation stations after removal of Gludsted Plantage Nv Station.

Considering longer duration precipitation estimation contains less errors than hourly estimations (Hema and Kant, 2017), a 24-hour moving window was used to obtain accumulated precipitation, which might give a better idea about water level change in the Storå River than the ones recorded hourly. In fact, accumulated precipitation for all four stations had a higher correlation coefficient as presented in Figure 25. The highest correlation coefficient result was observed in Grønbjerg Station with 0.369 and it followed with 0.363 in Øby Station and 0.352 in Isenvad Station. Høgild Station shows the lowest correlation with 0.315, yet it was decided to keep the station because the correlation coefficient is relatively higher.



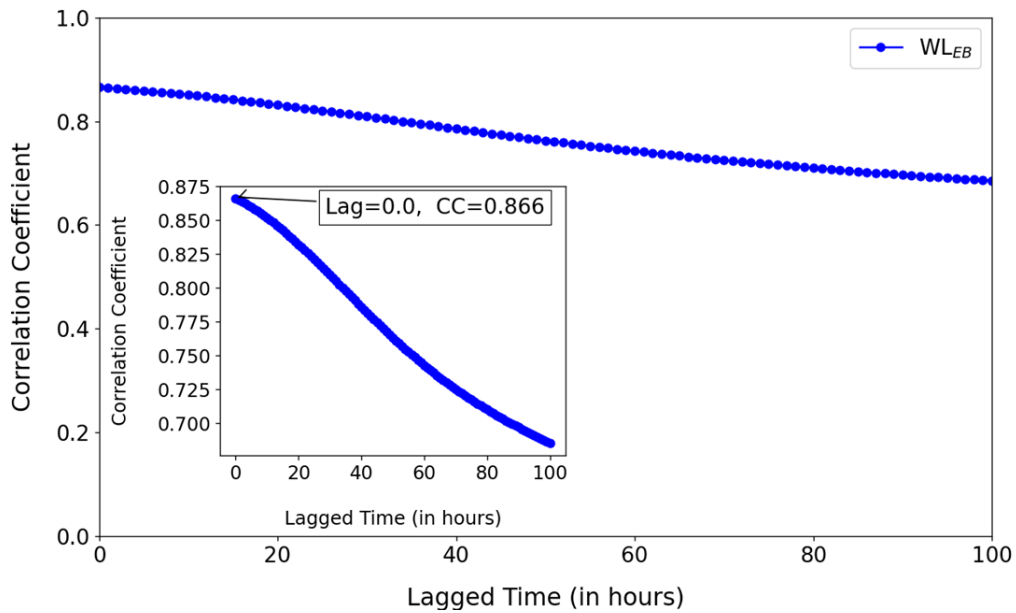
**Figure 25** | Correlation analysis between water level at Skærum Bro ( $WL_{SB}(t+48)$ ) and given precipitation stations for 24-hour accumulated precipitation.

Autocorrelation is a special case of cross correlation which refers to the correlation of a variable with itself at different times. It was used to assess the linear dependency of consecutive water level observations at the Skærum Bro Station. The aim was to understand whether two water level observations having 48 hours lag in between are correlated or not. The autocorrelation result for observed water level at Skærum Bro Station is presented in Figure 26. The correlation result among current and 48 hour previous water level was obtained as 0.92. This shows that the water level at Skærum Bro is highly correlated with itself even after 48-hours.



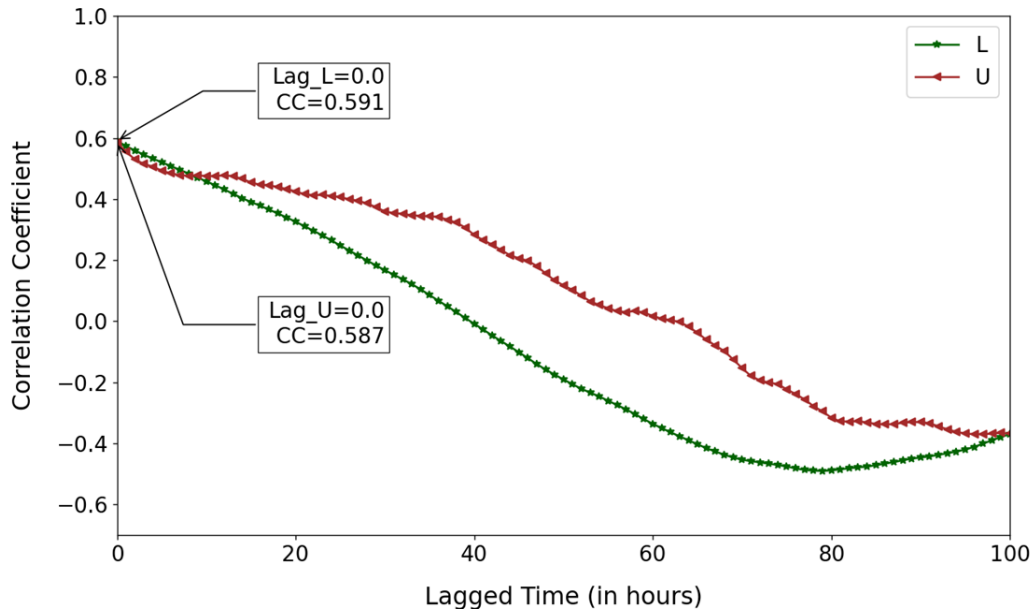
**Figure 26** | Autocorrelation for water level at Skærum Bro Station ( $WL_{SB}(t-Lag)$ ).

Cross correlation analysis was continued with water level measurements from Skærum Bro and Ellebæk Bro stations. The result is presented in Figure 27. The highest correlation was observed with 0.87 at time equal to zero.



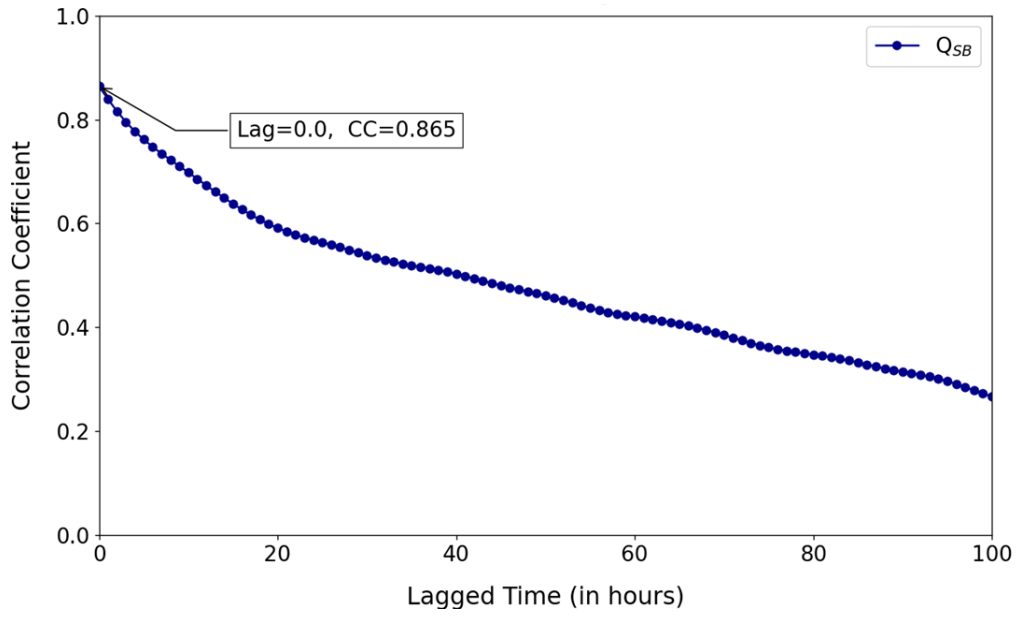
**Figure 27** | Correlation analysis between water level at Skærum Bro Station ( $WL_{SB}(t+48)$ ) and water level at Ellebæk Bro Station ( $WL_{EB}(t-Lag)$ ).

Afterwards, the correlation among water level at Skærum Bro Station and simulated soil moisture content for both root zone and surface based on the same station was analyzed. The result of correlation analysis is presented in Figure 28. According to results, root zone soil moisture content was slightly higher correlated with the target variable, 0.591 than soil moisture content on the surface, 0.587.



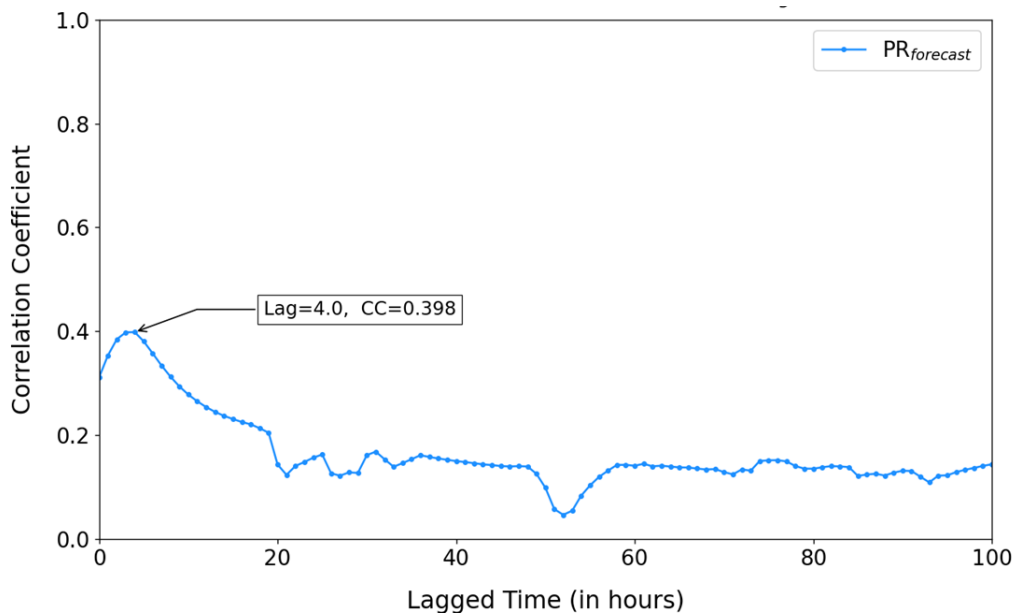
**Figure 28** | Correlation analysis between water level at Skærum Bro Station ( $WL_{SB}(t+48)$ ) and simulated soil moisture content for root zone,  $L$  and surface,  $U$  components ( $RMC(t-Lag)$ ).

Correlation analysis between simulated discharge and the target variable is presented in Figure 29. The correlation coefficient of 0.865 represents a significant correlation.



**Figure 29** | Correlation analysis between water level at Skærum Bro Station ( $WL_{SB}(t+48)$ ) and simulated discharge ( $Q_{SB}(t-Lag)$ ) at Skærum Bro Station.

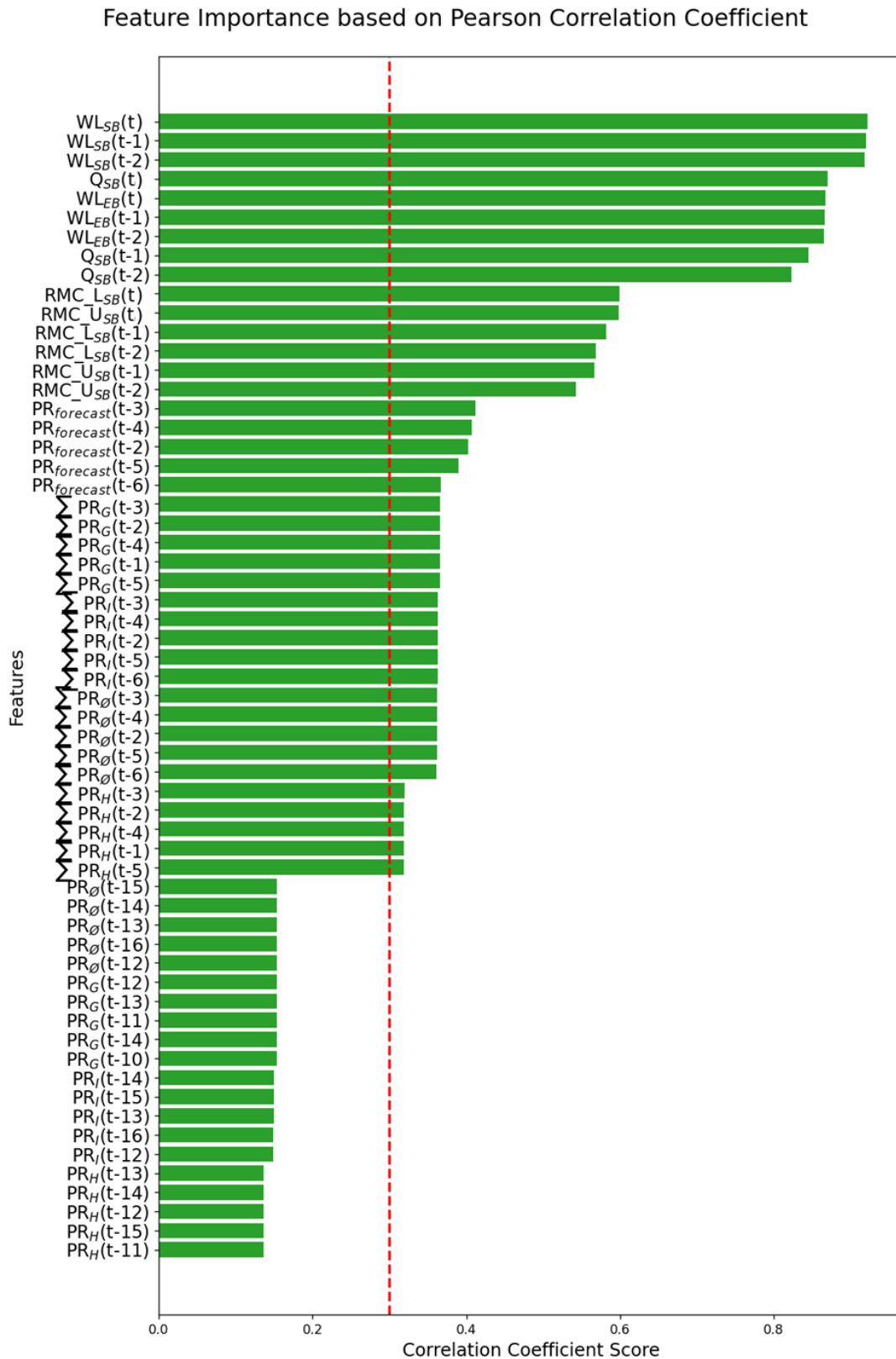
Correlation analysis between forecasted precipitation and the target variable is presented in Figure 30. The highest correlation was observed as 0.398 with a lag of 4 hours. The variations stem from the nonlinear nature of precipitation.



**Figure 30** | Correlation analysis between water level at Skærum Bro Station ( $WL_{SB}(t+48)$ ) and forecasted precipitation ( $PR_{forecast}(t-Lag)$ ).

Based on the given Pearson's correlation results, features were selected for the best correlation, two hour prior and two hour posterior time steps. For highest correlation achieved at  $t$  equal to zero, no posterior time step was considered due to unavailable information. To summarize which features are selected, relative importances is presented in Figure 31. As it can be seen from the distribution, the direct jump displayed after accumulated precipitation features. Since hourly precipitation features had weak correlations, it was decided to use only 24-hour accumulated precipitation for given stations. The red dotted line represents a significant threshold in feature importances.

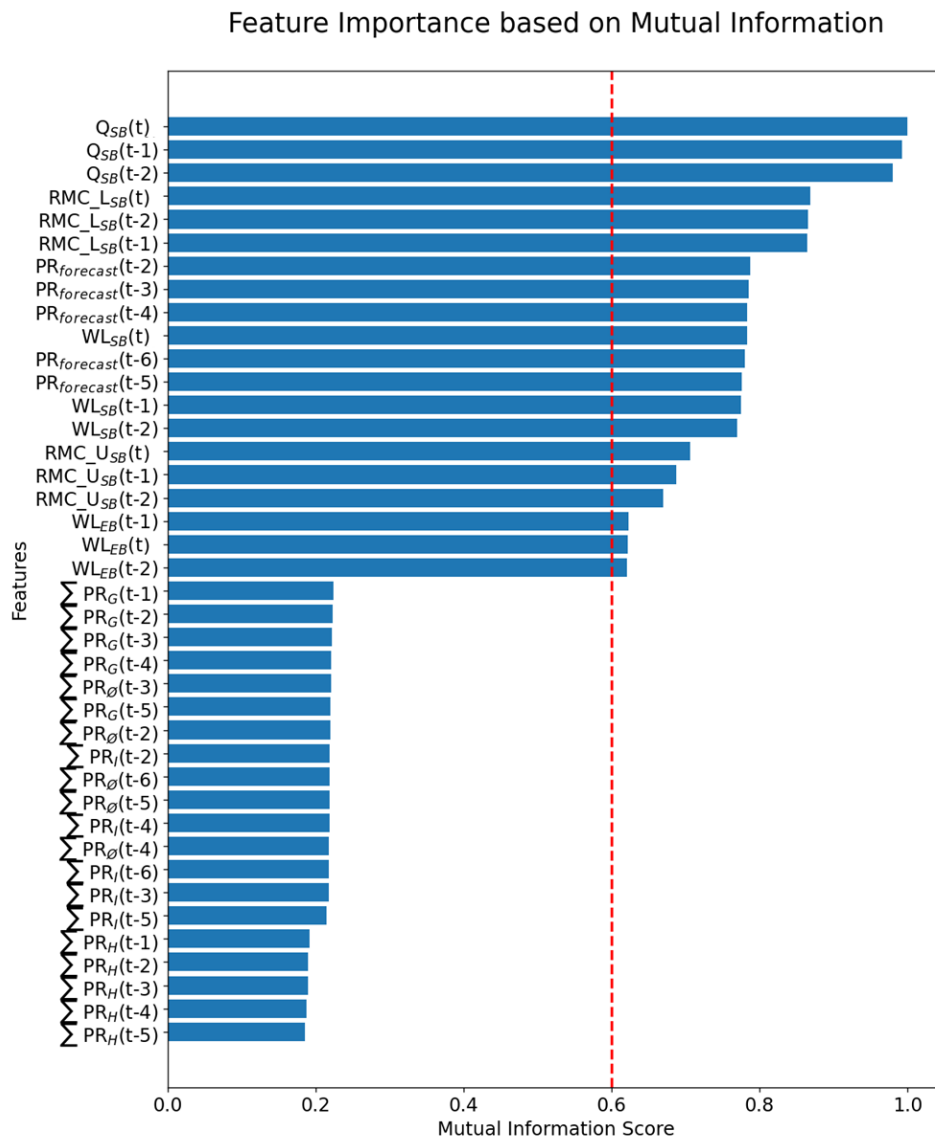
The results constitute the preliminary feature elimination based on filter method. It helps to measure the strength of linearity for the given features and target variable. These results can be used for any machine learning algorithm. After the first elimination based on the correlation analysis, feature elimination was carried on with the mutual information analysis in the next section.



**Figure 31** | *Feature importance based on Pearson's Correlation Coefficient.*

### 5.2.2 Mutual Information

Mutual information is a measure of dependence. The feature importance based on mutual information among the target variable and the selected features after correlation analysis is presented in Figure 32. The highest mutual information score was observed for simulated discharge at Skærum Bro Station. It followed by root zone soil moisture content, forecasted precipitation, water level at Skærum Bro Station, surface soil moisture content, water level at Skærum Bro Station, respectively. The sum sign represents 24-hour accumulated precipitation which was calculated by moving windows. For the given stations those features had the lowest score among all features and were removed as the study advanced. The red dotted line represents a significant threshold in feature importances.



**Figure 32** | *Feature Importance based on mutual information.*

### 5.2.3 Persistence Model

Creating a basic benchmark model is beneficial for machine learning tasks. It provides machine learning practitioners to understand the impact of an intervention which should be abandoned or enhanced moving forward. In this research, the persistence model was considered as a benchmark model. Persistence method constitutes one of the easiest methods in predicting future behavior. The main feature of the persistence model is that future values of any time series are calculated based on the stationarity assumption. In other words, nothing will change between the current time  $t$  and the forecast time  $(t+T)_{\text{forecast}}$  (Paulescu et al., 2021).

### 5.2.4 Feature Sets

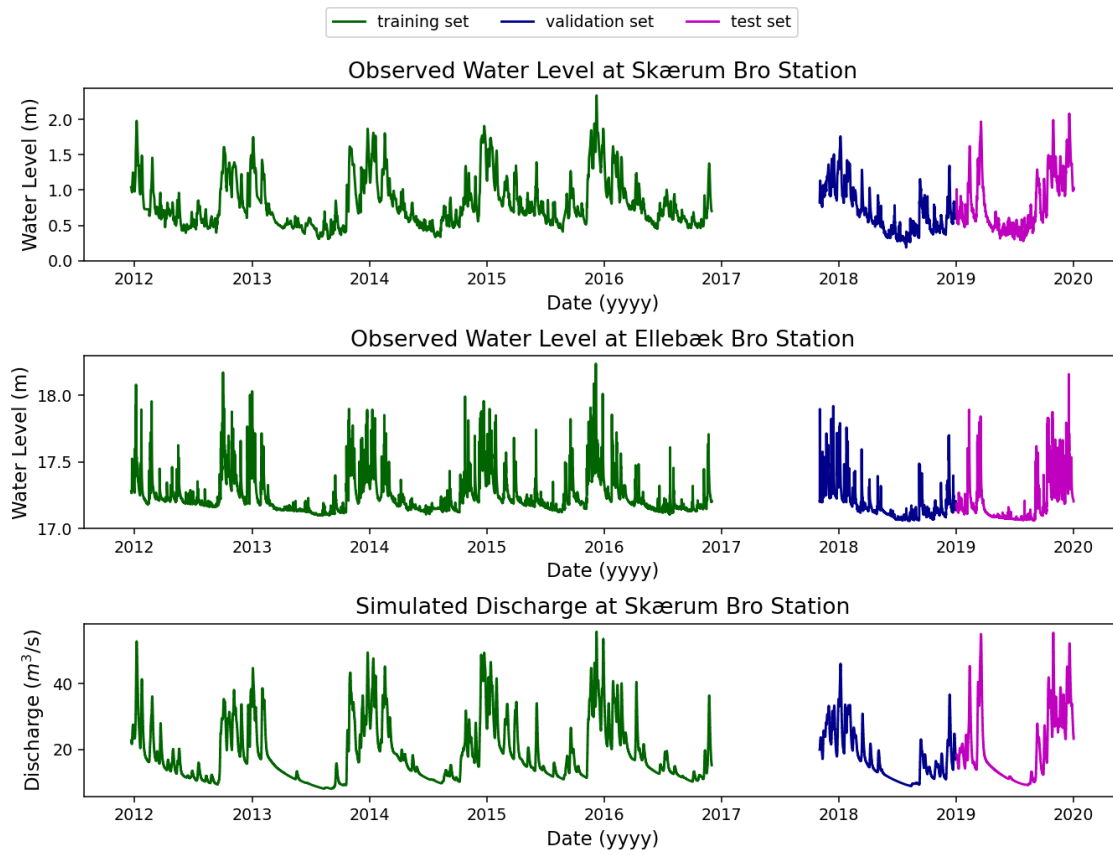
Feature sets were created in order to understand how features affect performance. These sets were created by considering the feature selection methods. The final features that were included in the machine learning models were discussed in the previous section. Thus, the first feature set consists of the simulated discharge at Skærum Bro for three time steps. The explicit features included in the feature set are presented in Table 2. Second feature set was created as an addition of the feature that has the second highest relative importance based on Figure 32 which is the simulated relative moisture content at the root zone to the first feature set. Third feature set was created by addition of forecasted precipitation features to the second feature set. Although  $WL_{\text{SB}}(t)$  had a slightly higher score than  $PR_{\text{forecasted}}(t-4)$  and  $PR_{\text{forecasted}}(t-5)$ , in order to keep the addition to sets constrained with only one information  $WL_{\text{SB}}(t)$  was not included in the third feature set. Water level at Skærum Bro, soil moisture content on the surface, and water level at Skærum Bro were added in order to create the fourth, the fifth, and the sixth feature sets, respectively.

**Table 2** | *Feature sets number and description.*

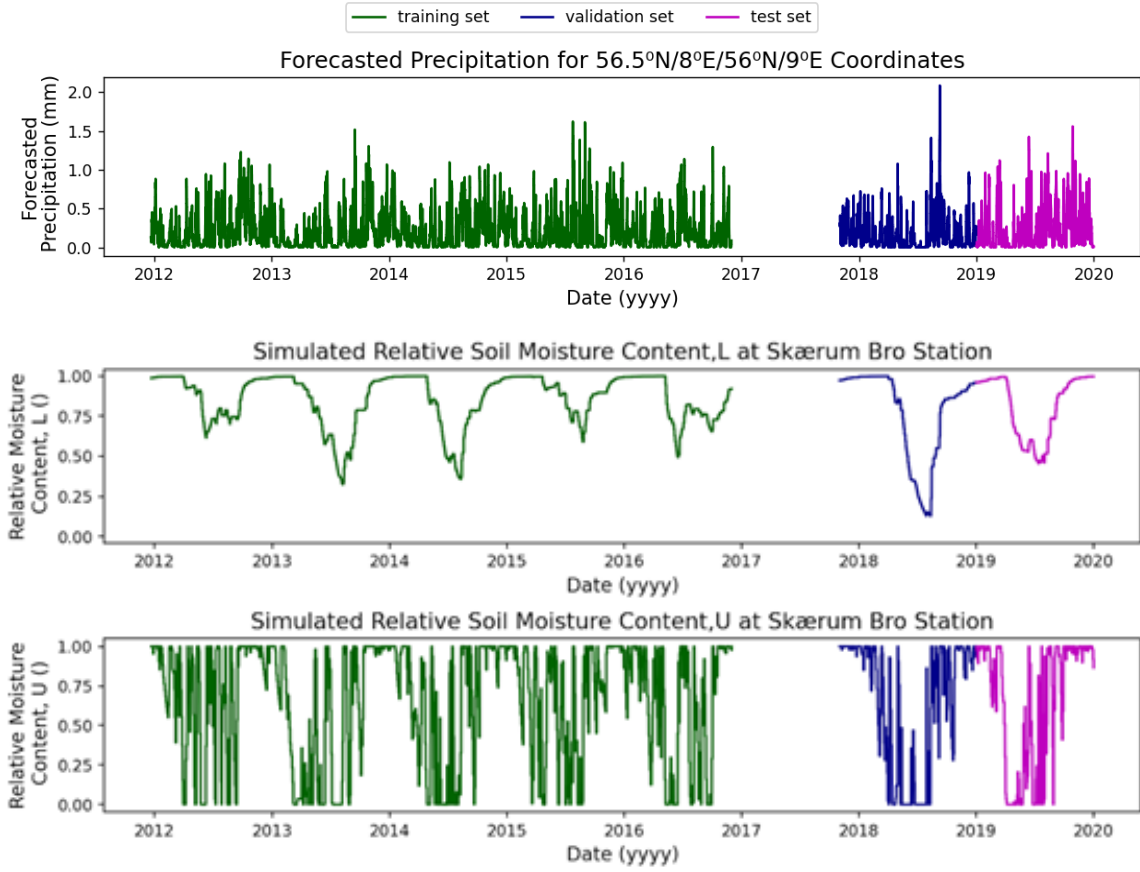
Feature Sets	
Number	Description
1	$Q_{SB}(t) + Q_{SB}(t-1) + Q_{SB}(t-2)$
2	$Q_{SB}(t) + Q_{SB}(t-1) + Q_{SB}(t-2) + RMC\_L(t) + RMC\_L(t-1) + RMC\_L(t-2)$
3	$Q_{SB}(t) + Q_{SB}(t-1) + Q_{SB}(t-2) + RMC\_L(t) + RMC\_L(t-1) + RMC\_L(t-2) + PR_{forecast}(t-2) + PR_{forecast}(t-3) + PR_{forecast}(t-4) + PR_{forecast}(t-5) + PR_{forecast}(t-6)$
4	$Q_{SB}(t) + Q_{SB}(t-1) + Q_{SB}(t-2) + RMC\_L(t) + RMC\_L(t-1) + RMC\_L(t-2) + PR_{forecast}(t-2) + PR_{forecast}(t-3) + PR_{forecast}(t-4) + PR_{forecast}(t-5) + PR_{forecast}(t-6) + WL_{SB}(t) + WL_{SB}(t-1) + WL_{SB}(t-2)$
5	$Q_{SB}(t) + Q_{SB}(t-1) + Q_{SB}(t-2) + RMC\_L(t) + RMC\_L(t-1) + RMC\_L(t-2) + PR_{forecast}(t-2) + PR_{forecast}(t-3) + PR_{forecast}(t-4) + PR_{forecast}(t-5) + PR_{forecast}(t-6) + WL_{SB}(t) + WL_{SB}(t-1) + WL_{SB}(t-2) + RMC\_U(t) + RMC\_U(t-1) + RMC\_U(t-2)$
6	$Q_{SB}(t) + Q_{SB}(t-1) + Q_{SB}(t-2) + RMC\_L(t) + RMC\_L(t-1) + RMC\_L(t-2) + PR_{forecast}(t-2) + PR_{forecast}(t-3) + PR_{forecast}(t-4) + PR_{forecast}(t-5) + PR_{forecast}(t-6) + WL_{SB}(t) + WL_{SB}(t-1) + WL_{SB}(t-2) + RMC\_U(t) + RMC\_U(t-1) + RMC\_U(t-2) + WL_{EB}(t) + WL_{EB}(t-1) + WL_{EB}(t-2)$

### 5.3 Data Split

Dividing the data set into training, validation, and test sets were done manually due to TIGGE-ECMWF data limitation for dates between the end of 2016 and 2017. The visualisation of data splits for each feature is presented in Figure 39 and 40. It was aimed to clarify the range, time span, and relatedness of variables for all three sets. In Table 5, the statistical details of each feature are tabulated. The minimum, maximum and mean for all features and for all three sets were quite close to each other with some differences.



**Figure 39** | *Data splits for the features of observed water level at both Skærum Bro and Ellebæk Bro stations, and simulated discharge at Skærum Bro Station for training (green), validation (dark blue), and test (pink) sets.*



**Figure 40** | *Data splits for the features of forecasted precipitation and simulated relative soil moisture contents L and U at Skærum Bro Station for training (green), validation (dark blue), and test (pink) sets.*

In the next chapter model improvement part, 5-fold cross validation was introduced as data splitting. The new training set used in the 5-fold cross validation was created by combining the datasets that were originally decided as training and validation.

**Table 5** | *Statistical details of selected features for training, validation, and test sets.*

Variables	Unit	Training Set			
		Mean	Std Dev	Min	Max
$Q_{SB}(t)$	$m^3/s$	19.06	9.27	8.08	55.75
$RMC_L(t)$	%	0.86	0.16	0.33	1.00
$PR_{forecasted}(t-4)$	mm	2.66	3.04	0.00	19.47
$WL_{SB}(t)$	m	0.81	0.35	0.31	2
$RMC_U(t)$	%	0.65	0.38	0	1
$WL_{EB}(t)$	m	17.25	0.14	17	18
$WL_{EB}(t+48)$	m	0.81	0.35	0	2.34

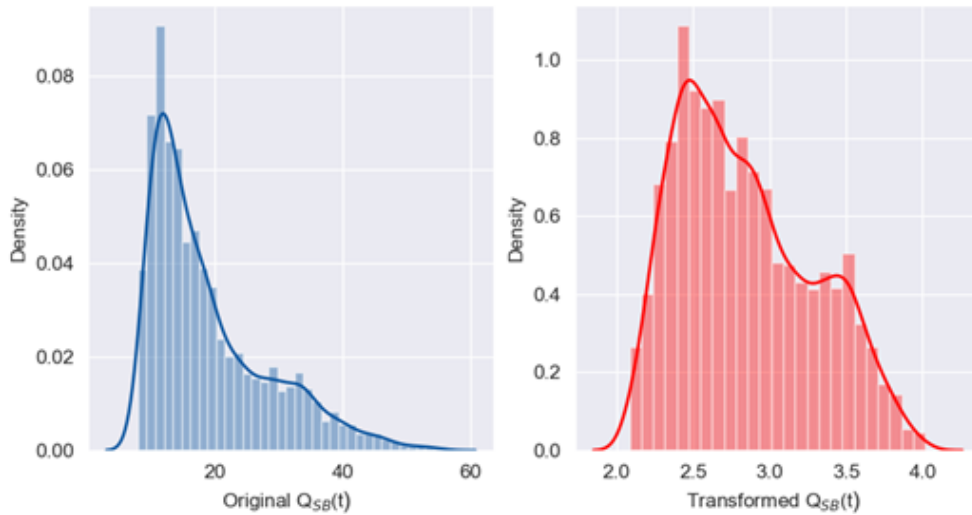
Variables	Unit	Validation Set			
		Mean	Std Dev	Min	Max
$Q_{SB}(t)$	$m^3/s$	17.66	7.46	8.89	46.03
$RMC_L(t)$	%	0.78	0.28	0.13	1.00
$PR_{forecasted}(t-4)$	mm	2.14	2.81	0.00	25.04
$WL_{SB}(t)$	m	0.69	0.32	0.19	2
$RMC_U(t)$	%	0.64	0.41	0	1
$WL_{EB}(t)$	m	17.18	0.12	17	18
$WL_{EB}(t+48)$	m	0.69	0.32	0	1.76

Variables	Unit	Test Set			
		Mean	Std Dev	Min	Max
$Q_{SB}(t)$	$m^3/s$	20.40	10.82	9.16	55.44
$RMC_L(t)$	%	0.82	0.19	0.45	1.00
$PR_{forecasted}(t-4)$	mm	3.09	3.49	0.00	18.72
$WL_{SB}(t)$	m	0.81	0.41	0.28	2
$RMC_U(t)$	%	0.66	0.40	0	1
$WL_{EB}(t)$	m	17.20	0.16	17	18
$WL_{EB}(t+48)$	m	0.81	0.41	0	2.08

## 5.4 Feature Transformation and Scaling

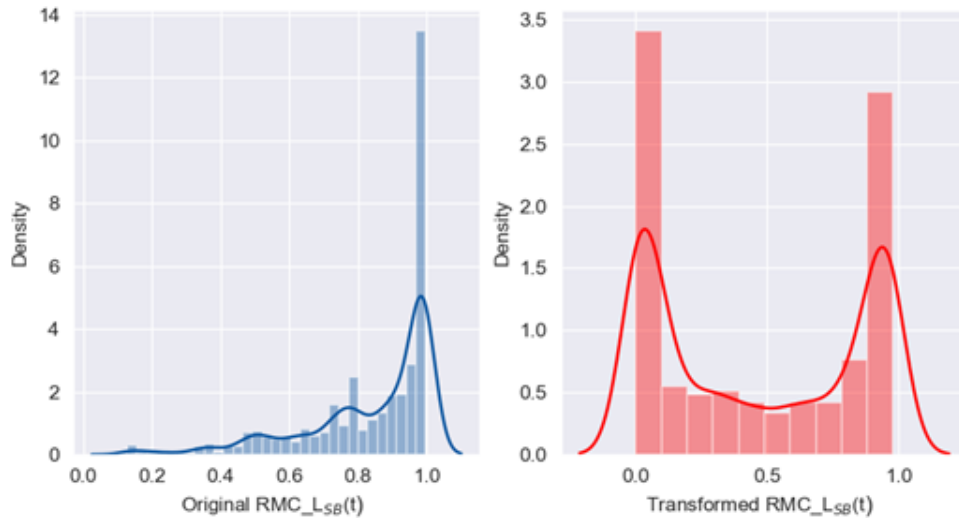
Feature transformation and scaling were experimented in order to understand their effect on machine learning models' performance. For the power transformations, first the skewness of each feature was measured. Based on the degree of skewness either a Box-cox, Square-Root, or Log transformation technique was implemented. Box-cox transformation can be applied to features with both positive and negative skewness. As a rule of thumb, when the skewness is in between  $(-0.5, 0.5)$ , it can be considered as a fairly symmetrical distribution (Droutsas et al., 2020). Thus, the threshold value was assigned as 0.5. If the absolute value of a feature skewness was greater than threshold and the skewness was positive, it was considered as a positively skewed feature and log transformation applied. If the absolute value of a feature skewness was greater than threshold and the skewness was smaller than zero, it was considered as a negative skewed feature and exponential transformation was used. The Box-Cox transformation was applied for both positive and negative skewness and the technique which gives minimum skewness was selected.

The original and post-transformation distribution plots are presented below. Since there were more than one features for the same variable which represents just shifted versions of each other, and they all share the almost exact statistics, only one of them was visualized. The histogram together with probability density function for the  $Q_{SB}(t)$  time series for original data on the left and transformed data on the right are presented in Figure 33. Original data exhibits positive skewed distribution with most data accumulated around left tail and longer right tail distribution. After transformation, heavily accumulated data on the left tail was distributed in the middle and the right tail was getting shortened. The skewness was decreased from 1.21 to 0.47.



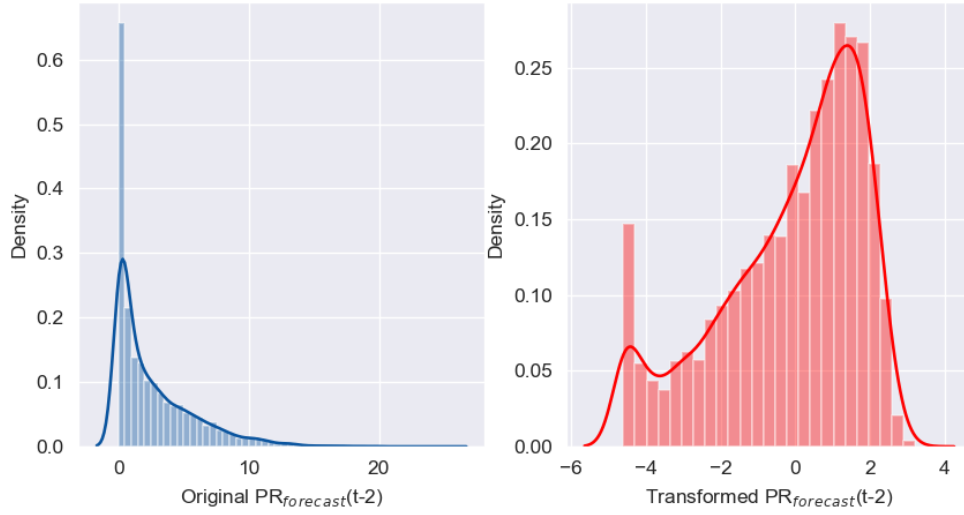
**Figure 33** | *Histogram and probability density function for  $Q_{SB}(t)$  original and post-transformation.*

The histogram and probability density function for  $RMC_{L_{SB}}(t)$  time series are presented in Figure 34 for both original and transformed data. The original data displays negative skewness with longer left tail distribution and accumulated data on the right tail. After transformation the skewness was significantly decreased, from -1.41 to 0.07. Although the distribution did not resemble a bell shaped curve, the distribution was approximately normal.



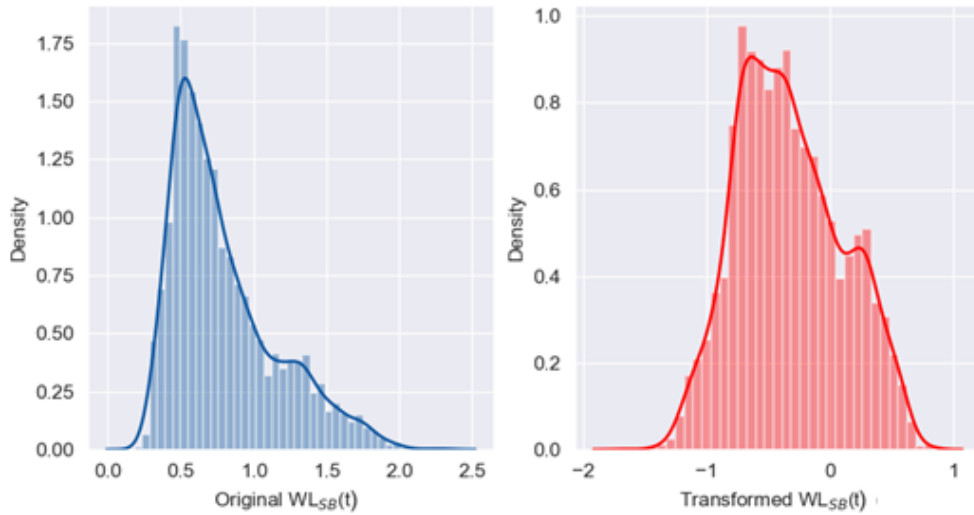
**Figure 34** | *Histogram and probability density function for  $RMC_{L_{SB}}(t)$  original and post-transformation.*

The histogram and probability density function for  $PR_{forecast}(t-2)$  time series are displayed below in Figure 35. The original data distribution was positive skewed with 1.65 skewness. After transformation, the distribution became moderately negative skewed with -0.79 skewness.



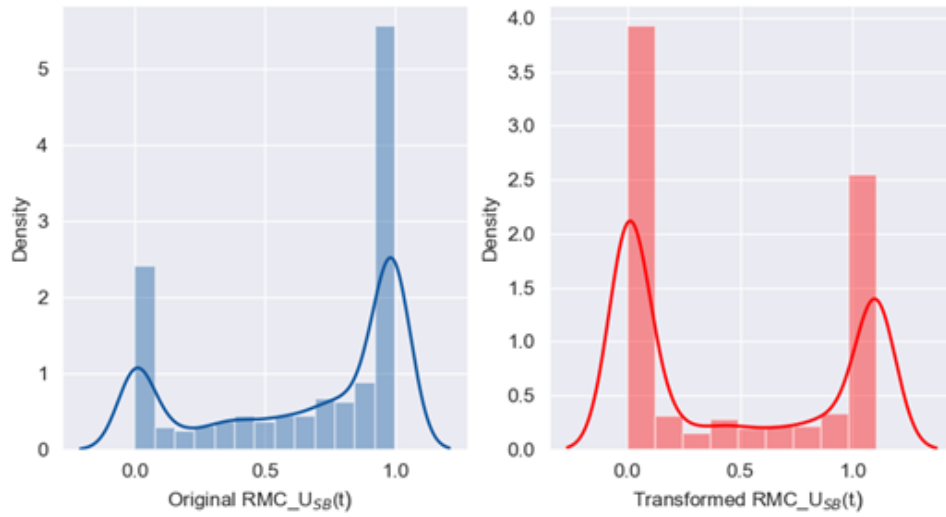
**Figure 35** | *Histogram and probability density function for  $PR_{forecast}(t-2)$  original and post-transformation.*

The histogram together with probability density function for the  $WL_{SB}(t)$  time series are plotted as below in Figure 36. The original data exhibited positive skewness and after transformation it became almost normally distributed data.



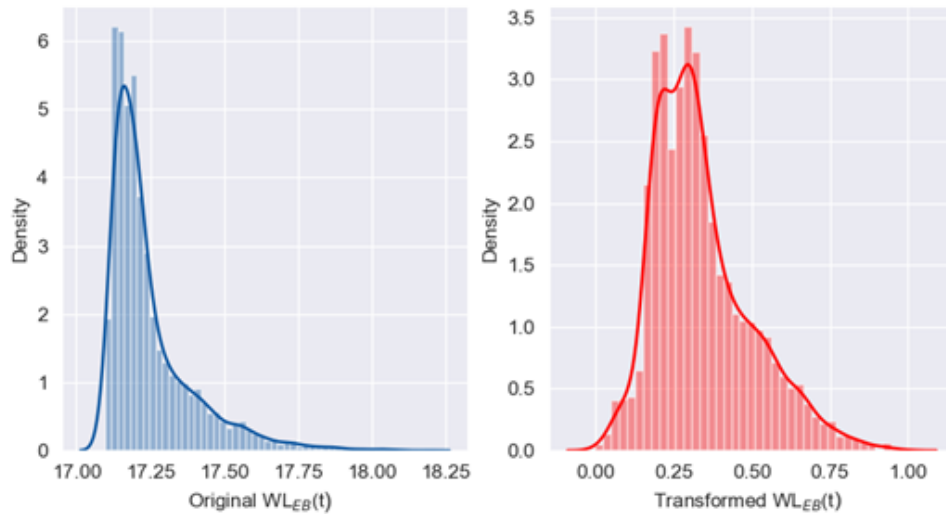
**Figure 36** | *Histogram and probability density function for  $WL_{SB}(t)$  original and post-transformation.*

The histogram and probability density function for  $RMC_{U_{SB}}(t)$  time series are presented for both original and transformed data in Figure 37. The original data displays negative skewness with -0.69 skewness. After transformation, the skewness significantly decreased to 0.32. Although the distribution still does not look like a bell shaped curve, the distribution was approximately normal.



**Figure 37** | *Histogram and probability density function for  $RMC_{U_{SB}}(t)$  original and post-transformation.*

The histogram together with probability density function for the  $WL_{EB}(t)$  time series for the original data on the left and transformed data on the right are presented in Figure 38. Original data exhibits positive skewed distribution with 1.91 skewness. After transformation the skewness was decreased to 0.93.



**Figure 38** | *Histogram and probability density function for  $WL_{EB}(t)$  original and post-transformation.*

Although not all features are turned into Gaussian distributions, they seemed Gaussian-like distributions and it was found to be adequate to move forward with the scaling techniques. The tabulated form of skewness results for transformed features and raw version are presented as a summary in Table 3.

**Table 3** | *Skewness results original and post-transformation.*

Feature	Skewness	
	Before Transformation	After Transformation
$Q_{SB}(t)$	1.21	0.47
$Q_{SB}(t-1)$	1.21	0.47
$Q_{SB}(t-2)$	1.21	0.47
$RMC_{L_{SB}}(t)$	-1.41	0.07
$RMC_{L_{SB}}(t-1)$	-1.41	0.07
$RMC_{L_{SB}}(t-2)$	-1.41	0.07
$PR_{forecast}(t-2)$	1.65	-0.79
$PR_{forecast}(t-3)$	1.65	-0.79
$PR_{forecast}(t-4)$	1.65	-0.79
$PR_{forecast}(t-5)$	1.65	-0.79
$WL_{SB}(t)$	1.06	0.25
$WL_{SB}(t-1)$	1.06	0.24
$WL_{SB}(t-2)$	1.06	0.24
$RMC_{U_{SB}}(t)$	-0.69	0.33
$RMC_{U_{SB}}(t-1)$	-0.69	0.32
$RMC_{U_{SB}}(t-2)$	-0.69	0.32
$WL_{EB}(t)$	1.91	0.93
$WL_{EB}(t-1)$	1.89	0.92
$WL_{EB}(t-2)$	1.87	0.92

## Chapter 6. Results and Discussion

*This chapter presents the results of the research study in line with fulfilling the research objectives presented in Chapter 1. This chapter was designed in five sections revolving around the assessment of feature transformation and scaling, the selection of feature sets, the model improvement, introducing the artificial neural network, and the overall assessment of the models. It is important to mention, in the first three parts of this chapter the analyses were conducted using Multiple Linear Regression, Random Forest Regression, Gradient Boosting Regression. The created Gaussian-like dataset and the original dataset in the data analysis chapter were subjected to no-scaling and scaling through normalization and standardization in order to investigate the effect of the feature transformation and scaling on machine learning models' performance on the first part. Afterwards, in the second part the performance of the machine learning models were examined with different feature sets. In the third part, the improvement of machine learning models through hyperparameter tuning and the implementation of a hybrid method in feature selection was analyzed. The Feed-Forward Neural Network model did not include the first three parts due to time limitation. It only included the last part where all four machine learning models were compared and assessed in terms of water level prediction accuracy.*

### 6.1 Part 1: Assessment of Feature Transformation and Scaling

There is no fixed solution to choose which scaling technique to use in the data preparation part. Therefore, in order to decide the scaling technique to be implemented, an experiment was conducted using the original and Gaussian-like distributed data that created Chapter 5.4. For both datasets three cases were considered based on no scaling and scaling through normalization and standardization. In python, the MinMaxScaler library was used for normalization and the StandardScaler library was used for standardization. Machine learning models were trained using the training dataset created in Chapter 5.3 using the all features and validation results for mean absolute error terms are presented in Table 4, original data on the top and transformed data on the bottom.

**Table 4** | Mean absolute error from given machine learning models for original data (on top) and for Gaussian-like distributed data (on bottom) considering no scaling, normalization, and standardization. (The lower error values are highlighted for each model).

MAE for original data (m)			
Model	No Scaling	Normalized	Standardized
LR	0.0882	0.0874	0.0874
RF	0.0975	0.0975	0.0976
GB	0.0923	0.0923	0.0924

MAE for Gussian-like distributed data (m)			
Model	No Scaling	Normalized	Standardized
LR	0.0899	0.0898	0.0898
RF	0.1025	0.1027	0.1028
GB	0.0948	0.0949	0.0949

The mean absolute error terms for the Random Forest Regression and the Gradient Boosting Regression models displayed in Table 4 were presented after hyperparameter tuning in order to obtain more realistic results after decreasing the overfitting. Tree based algorithms are known as scale invariant, thus, expected behavior of Random Forest and Gradient Boosting was no change in the error term for normalization and standardization. However there were slight differences for no scaling and scaling through normalization and standardization for Random Forest and Gradient Boosting models. This can be explained by hyperparameter tuning for those models. Later, the transformation techniques were implemented and as presented in Table 4 on the bottom, the mean absolute error term increased. This shows transforming the data to follow Gaussian-like distribution was not a sound intervention in this case. The uniform distribution exists in nature so does skewed distribution as well. It is not always possible to transform data to follow normal distribution and expect it to perform better with machine learning algorithms as presented here. Based on the results of this procedure, normalization techniques on raw data were to be implemented moving forward.

## 6.2 Part 2: Assessment of Different Feature Sets

In this part the training and the validation results obtained through training of the three machine learning models with predefined feature sets were investigated. Although three different evaluation criteria were presented, the mean absolute error term will be examined mostly due to its ability in communicating with early warning systems. In other words, knowing the error margin coming from machine learning predictions allows flood risk managers to have safe ground in their decision making process. As a first step, the results of machine learning models after each feature set were compared by the persistence model in order to understand if the machine learning models were able to beat a simple model or not.

Persistence model serves as the base model that shifts the water level at Skærum Bro Station 48-hour ahead of time and considers that measurement as the prediction and 48-hour previous measurement as the current water level. Thus, the error terms were calculated among these two values both coming from the historical measurements. In essence, this is just basic statistics in analysing trends and assuming the behavior for the desired time period would be the same as prior trend. This actually means an assumption that the conditions affecting water level in the Storå River are stationary.

The result of the persistence model is presented in Table 6. Without any machine learning model, persistence model was able to predict water level in the Storå River less than 10 cm error based on the validation set. Coefficient of determination for the train test was higher than the validation set. This shows at the beginning of the time series (2011-2016), water level recordings in two days were more correlated than the validation period (2017).

**Table 6** | *Persistence model prediction results for training and validation set.*

Evaluation Criteria	Train	Validation
MAE	0.090	0.098
RMSE	0.134	0.145
$R^2$	0.857	0.788

The first feature set contains only  $Q_{SB}(t)$ ,  $Q_{SB}(t-1)$  and  $Q_{SB}(t-2)$  features as an input to machine learning models as explicitly presented in Chapter 5.2.4. The evaluation results on both training and validation sets are presented in Table 7. For Random Forest and Gradient Boosting models predictions were improved through hyperparameter tuning, on the other hand in Multiple Linear Regression was stayed as it was since it does

not have hyperparameters to tune. In the Multiple Linear Regression model the mean absolute error was recorded 5.1 cm higher; in the Random Forest model the mean absolute error was recorded 7.2 cm higher; and in the Gradient Boosting model the mean absolute error was recorded 6.5 cm higher compared to the persistence model based on validation set. It can be concluded that the persistence model outperforms the prediction performance of machine learning models trained with the first feature set.

**Table 7 |** *Prediction results of the given machine learning models training on the first feature set. (Compared to the persistence model, mean absolute error values are highlighted for the underperformed machine learning models. (\*) represents hyperparameter tuning.)*

Feature Set	Evaluation Criteria	Train			Validation		
		MLR	RF*	GB*	MLR	RF*	GB*
1	MAE	0.100	0.112	0.093	0.149	0.170	0.163
	RMSE	0.134	0.154	0.127	0.177	0.205	0.197
	R <sup>2</sup>	0.857	0.810	0.872	0.687	0.580	0.609

The second feature set was created by adding simulated relative moisture content at the root zone to the existing feature set. The evaluation results on both training and validation sets are presented in Table 8. It can be seen from the error terms and coefficient of determination on the training set, both Multiple Linear Regression and Random Forest models were learned from the newly added features. Although the mean absolute error terms on the validation set were also improved slightly for these models, the results were still not good enough to beat the persistence model. On the other hand, the Gradient Boosting model performed worse than the previous model. The interpretation for this can be relative soil moisture content at the root zone did not contain useful information for the Gradient Boosting model.

**Table 8 |** Prediction results of the given machine learning models training on the second feature set. (Compared to the persistence model, mean absolute error values are highlighted for the underperformed machine learning models. (\*) represents hyperparameter tuning.)

Feature Set	Evaluation Criteria	Train			Validation		
		MLR	RF*	GB*	MLR	RF*	GB*
2	MAE	0.096	0.100	0.091	0.145	0.160	0.167
	RMSE	0.131	0.138	0.126	0.174	0.196	0.201
	R <sup>2</sup>	0.864	0.848	0.872	0.697	0.615	0.596

The third feature set was created by the addition of forecasted precipitation retrieved from TIGGE data to the second feature set. The evaluation results on both training and validation sets are presented in Table 9. The mean absolute error terms decreased for all machine learning models both on training and validation sets. 3.4 cm improvement in the mean absolute error term was observed in the Multiple Linear Regression model. This improvement was 4.4 cm for the Random Forest model and 5.3 cm for the Gradient Boosting model. The forecasted precipitation was expected to be very important information to affect the water level which was confirmed by the results. However, even this much of a difference was not enough to surpass the persistence model's performance.

**Table 9 |** Prediction results of the given machine learning models training on the third feature set. (Compared to the persistence model, mean absolute error values are highlighted for the underperformed machine learning models. (\*) represents hyperparameter tuning.)

Feature Set	Evaluation Criteria	Train			Validation		
		MLR	RF*	GB*	MLR	RF*	GB*
3	MAE	0.077	0.065	0.064	0.111	0.116	0.114
	RMSE	0.106	0.094	0.089	0.139	0.145	0.138
	R <sup>2</sup>	0.909	0.930	0.937	0.806	0.788	0.808

The fourth feature set contains water level measurements coming from Skærum Bro Station in addition to what the third feature set had. The evaluation results on both training and validation sets are presented in Table 10. Error terms and coefficient of determination were improved for both the training and the validation sets compared to the previous models and finally outperformed the persistence model. The mean absolute error improvements on validation set were 2.4 cm, 2.5 cm, 2.8 cm for Multiple Linear Regression, Random Forest, and Gradient Boosting, respectively.

**Table 10** | *Prediction results of the given machine learning models training on the fourth feature set. (\*) represents hyperparameter tuning.)*

Feature Set	Evaluation Criteria	Train			Validation		
		MLR	RF*	GB*	MLR	RF*	GB*
4	MAE	0.062	0.042	0.045	0.087	0.091	0.086
	RMSE	0.090	0.063	0.065	0.121	0.125	0.117
	R <sup>2</sup>	0.935	0.969	0.967	0.853	0.842	0.864

In the fifth feature set, the relative soil moisture content at the top soil was included in the fourth feature set. The evaluation results on both training and validation sets are presented in Table 11. The improvement was only observed for the Random Forest model on validation set, whereas Multiple Linear Regression and Gradient Boosting models' mean absolute error terms on validation set were decreased by 2 mm compared to the previous feature set.

**Table 11** | *Prediction results of the given machine learning models training on the fifth feature set. (\*) represents hyperparameter tuning.)*

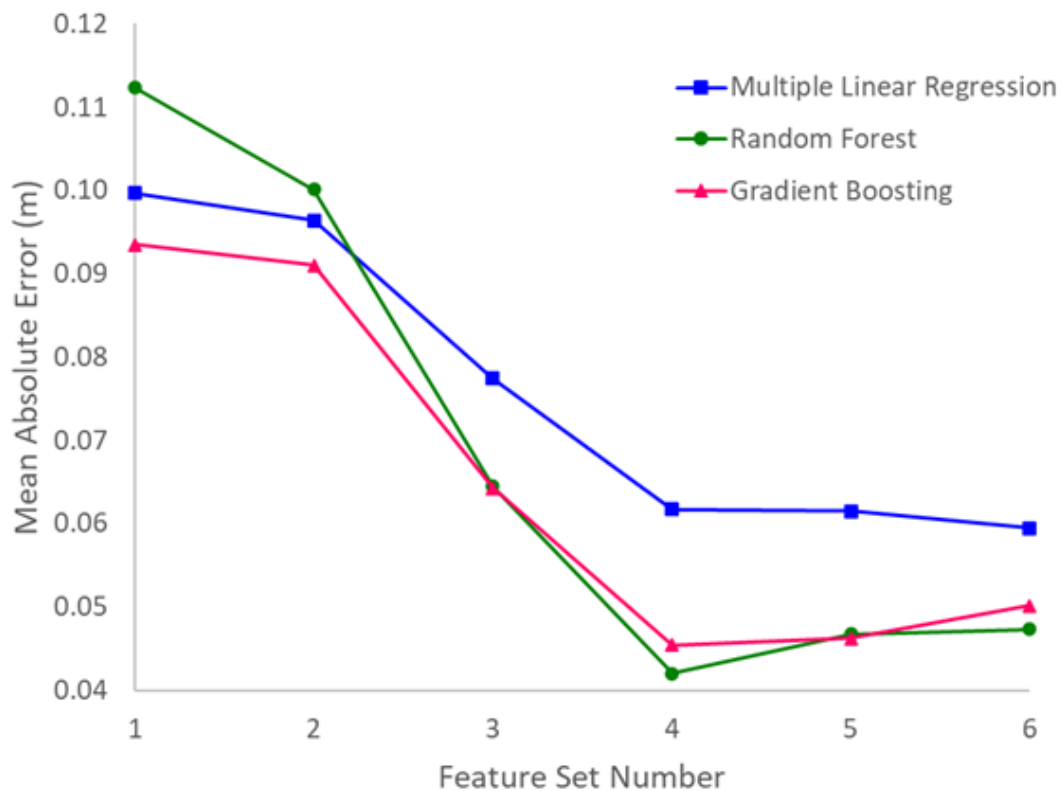
Feature Set	Evaluation Criteria	Train			Validation		
		MLR	RF*	GB*	MLR	RF*	GB*
5	MAE	0.061	0.047	0.046	0.089	0.089	0.088
	RMSE	0.090	0.085	0.067	0.122	0.124	0.118
	R <sup>2</sup>	0.936	0.952	0.965	0.851	0.845	0.859

In the sixth and the final feature set which includes all 19 features after the filter feature selection methods, water level coming from Ellebæk Bro station was added to the fifth feature set. The evaluation results on both training and validation sets are presented in Table 12. All three models performed better with the addition of this information. The improvement on mean absolute error terms were 7 mm, 1 mm and 4 mm for Multiple Linear Regression, Random Forest, and Gradient Boosting models respectively. It is important to note that the error terms in the training dataset for all three models were quite lower than the errors coming from the validation dataset. This is an indicator for overfitting.

**Table 12** | *Prediction results of the given machine learning models training on the sixth feature set. (\*) represents hyperparameter tuning.)*

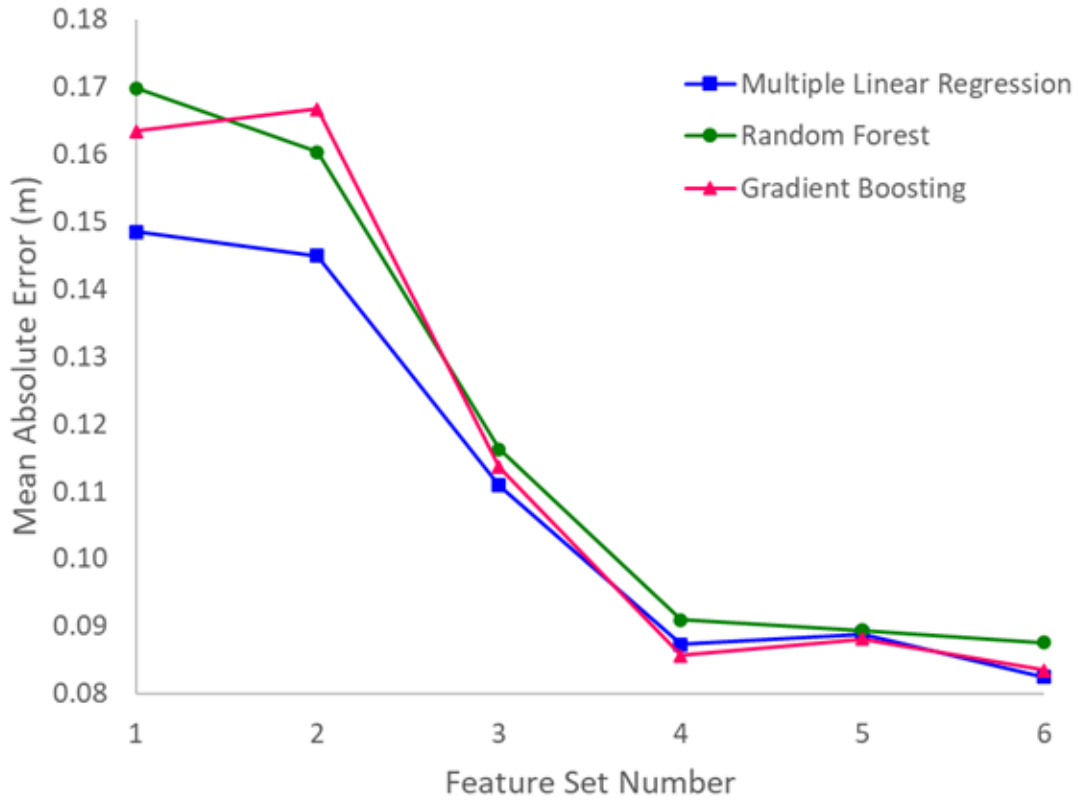
Feature Set	Evaluation Criteria	Train			Validation		
		MLR	RF*	GB*	MLR	RF*	GB*
6	MAE	0.059	0.047	0.050	0.082	0.088	0.084
	RMSE	0.086	0.071	0.072	0.116	0.119	0.113
	R <sup>2</sup>	0.940	0.959	0.959	0.864	0.859	0.872

To summarize the tabulated data presented above and visualize the influence of each feature set, a number of plots were introduced. Mean absolute error on the training set displayed in Figure 41 for all the feature sets. This plot shows a decreasing trend on the mean absolute error which can be explained by increasing information through added feature sets. A steep decrease in the error was observed for feature sets 3 and 4, which correspond to features for forecasted precipitation and water level measurements at Skærum Bro Station. Although these two features did not have the highest importance after the applied filter methods for feature selection, they had a remarkable contribution on the training performance. Another noticeable mean absolute error result was observed for the Random Forest model after training with the fifth feature set. Normally adding more data would not inhibit the training performance. However, if more features are added, it becomes more difficult to determine correct coefficients and increases the risk of overfitting.



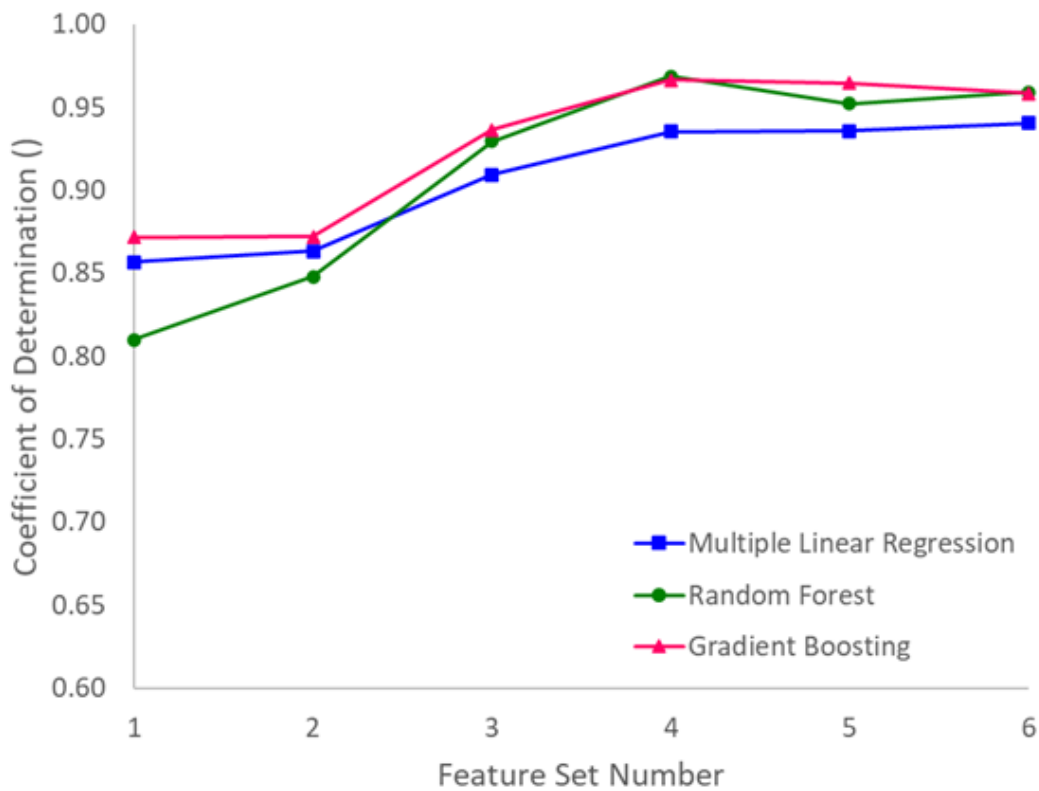
**Figure 41** | *Summary of mean absolute error on the training set for given feature sets.*

Mean absolute error for the validation set is displayed in Figure 42 for all feature sets. Water level predictions were improved as more features were introduced to the machine learning models. The error range was in between approximately 17 cm to 8 cm. Steep decrease in the error term occurred for feature sets 3 and 4 as observed in the training set presented above.



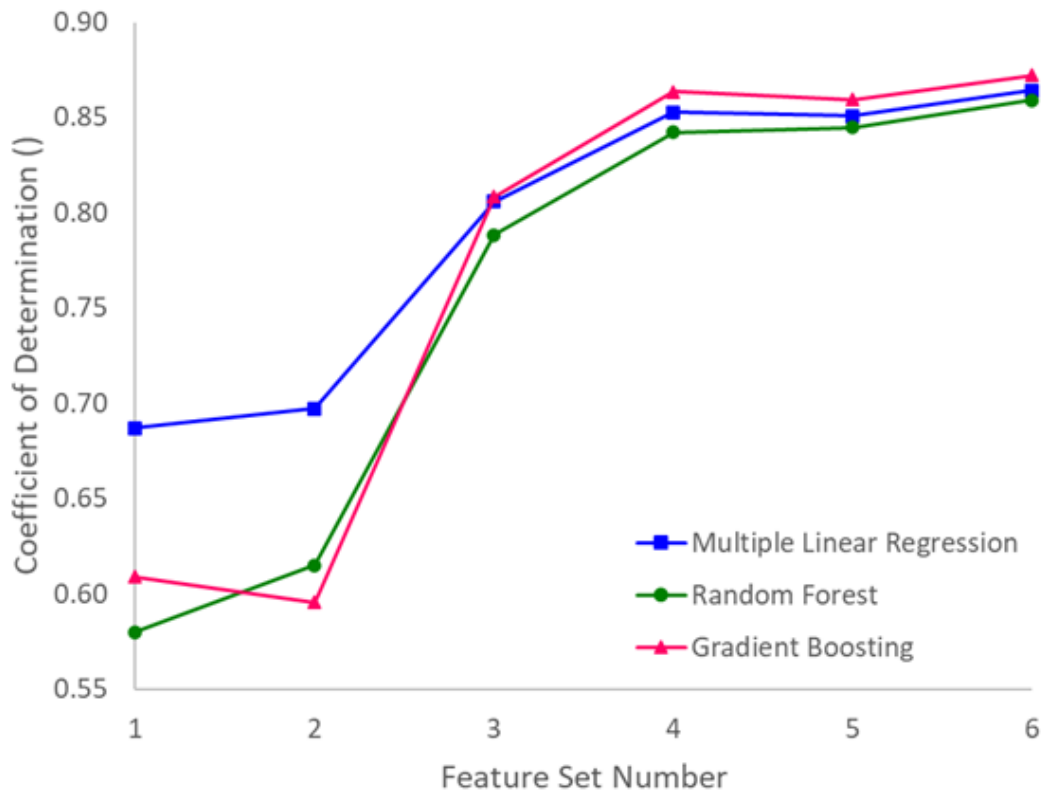
**Figure 42** | *Summary of mean absolute error on the validation set for given feature sets.*

Coefficient of determination plot is presented in Figure 43 for the training set. It showed an increasing trend for all machine learning models. For the fifth feature set a slight decrease is observed in the coefficient of determination value for the Random Forest model. On the other hand, the change for Gradient Boosting and Multiple Linear Regression models were insignificant for the same feature set. This slight decrease in the coefficient of determination for Random Forest indicates either randomness of hyperparameter tuning due to randomized search or overfitting of the training data.



**Figure 43** | *Summary of coefficient of determination on the training set for given feature sets.*

Coefficient of determination for the validation set is presented in Figure 44. An increasing trend observed through the feature sets. For the second feature set, a slight decrease is observed in the coefficient of determination value for the Gradient Boosting model which can be explained by the randomness of the hyperparameter tuning process. Addition of the forecasted precipitation on the third feature set resulted with a significant increase in the coefficient of determination values for all machine learning models. It is important to note that with the first feature set consisting of only one type of feature, the machine learning models performed over 0.55 coefficient of determination value. This shows the machine learning models perform to some degree even with the limited data. The overall improvement observed for the Multiple Linear Regression model is 25%, for the Random Forest model is 48% , and for the Gradient Boosting model is 43%.



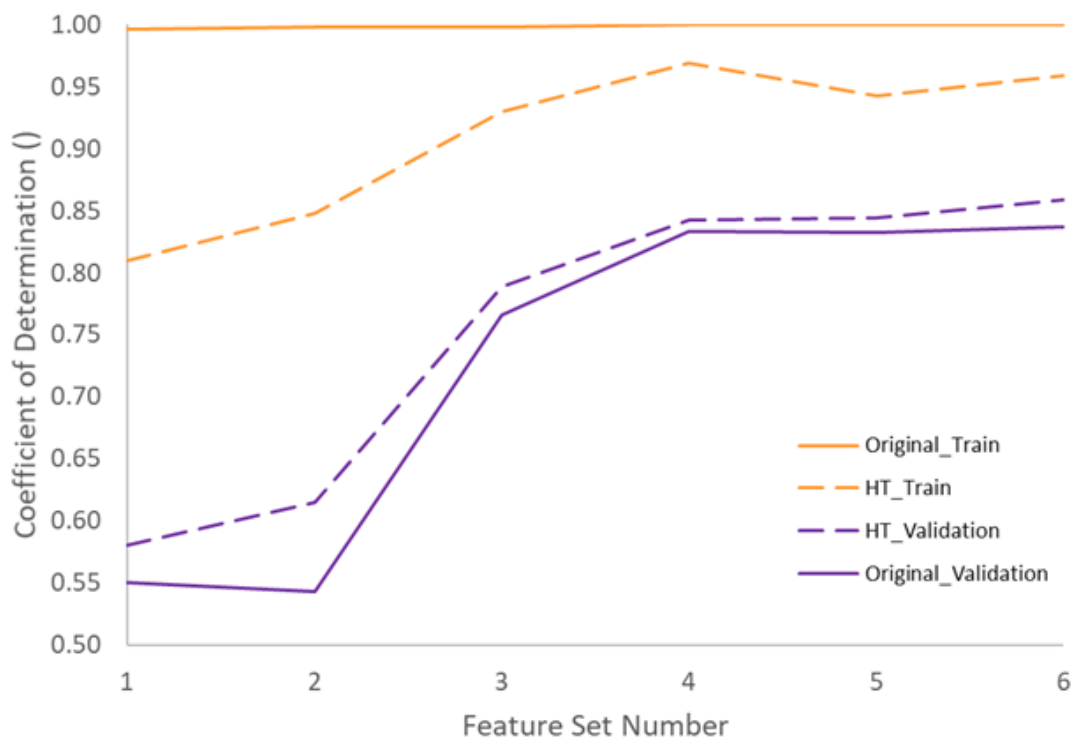
**Figure 44** | *Summary of coefficient of determination on the validation set for given feature sets.*

## 6.3 Part 3: Model Improvement

### 6.3.1 Effect of Hyperparameter Tuning

In the previous part, the results presented were already tuned for hyperparameters. In this section, the results without hyperparameter tuning were evaluated and the improvement in the prediction performances were discussed.

The effect of hyperparameter tuning for Random Forest Regression is presented in Figure 45. Training and validation sets were visualized with orange and purple colors, respectively. Original results presented with straight lines and results obtained after hyperparameter tuning presented with dotted lines. The model was trained by the sixth feature set.

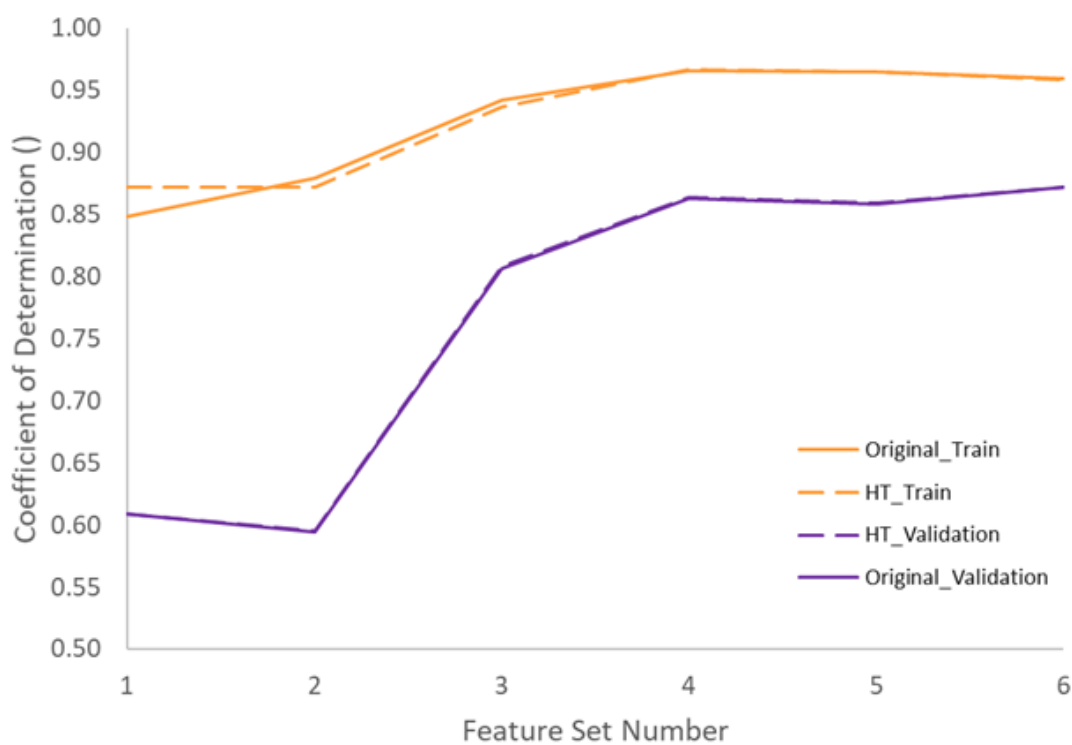


**Figure 45** | *Improving Random Forest Regression model through hyperparameter tuning. Original dataset represents the data without hyperparameter tuning, presented with a straight line and HT represents the hyperparameter tuned data, presented with a dotted line. Orange color stands for the training dataset and purple color stands for the validation dataset.*

The interpretation of the continuous orange line is the Random Forest Regression model overfitted for each and every feature set. In training models, the hyperparameters were used as it is without any alteration in the original Random Forest Regression model.

Since there were no `max_depth` or other parameters assigned, the model overfitted for each feature set training. This situation caused a lower coefficient of determination values for validation set as presented with the purple straight line. After hyperparameter tuning the results become more reasonable. The overall observed improvement for the validation set is 48%.

The effect of hyperparameter tuning was elaborated by showing the train and validation set results with and without hyperparameter tuning for the Gradient Boosting Regressor models trained by the feature set 6. The improvement for Gradient Boosting was minute for both training and validation sets. Only at the beginning, hyperparameter tuning improved the accuracy of the train set. Compared to Random Forest, Gradient Boosting is less likely to overfit on the training data.

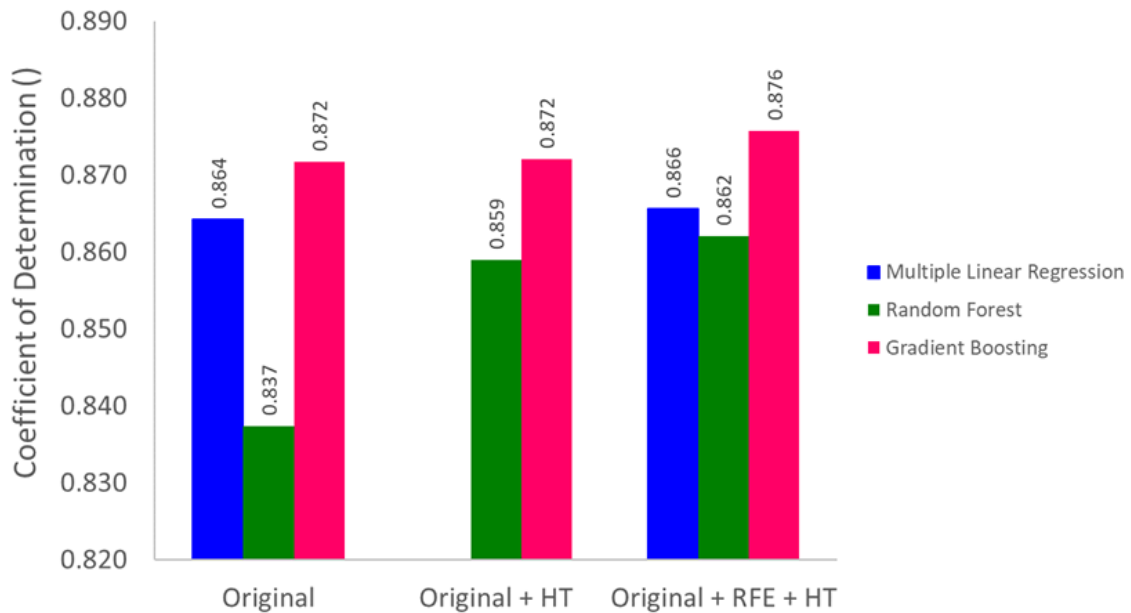


**Figure 46** | *Improving Gradient Boosting Regressor model through hyperparameter tuning. Original dataset represents the data without hyperparameter tuning, presented with a straight line and HT represents the hyperparameter tuned data, presented with a dotted line. Orange color stands for the training dataset and purple color stands for the validation dataset.*

### 6.3.2 Effect of Recursive Feature Elimination

Recursive feature elimination (RFE) is a wrapper feature selection method. When it is combined with previous correlation and mutual information analyses they form a hybrid method for feature selection. RFE was applied together with the machine learning algorithm and ranked the features based on their importances then recursively eliminated the features according to predefined step.

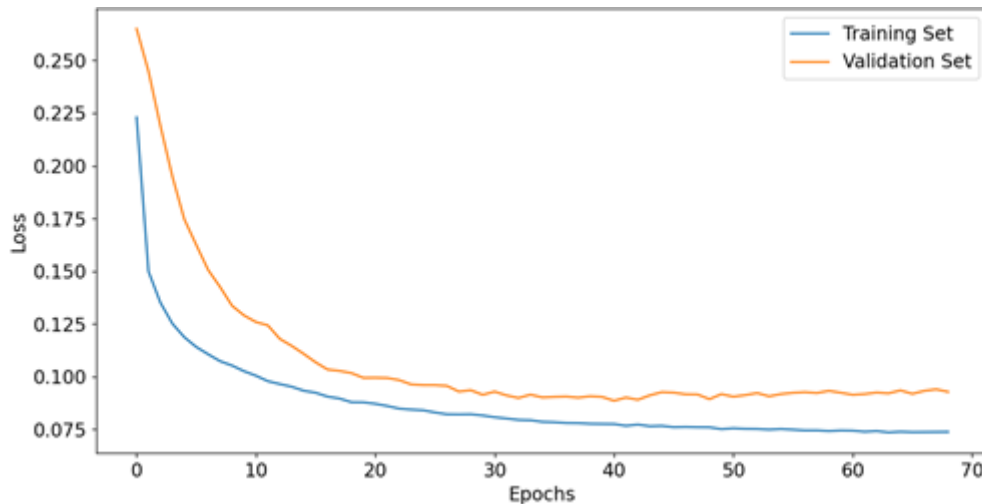
In this research RFECV class was utilized from sklearn library. The recursive elimination was conducted one feature at a time step considering coefficient of determination. As a minimum number of features, five features were selected. After the elimination, 14 features remained for Multiple Linear Regression. This number even further decreased for the Random Forest and the Gradient Boosting with 5 features for both models. The effect of improvement is demonstrated in Figure 47.



**Figure 47** | *Evaluation of model improvement on validation set for original, addition of hyperparameter tuning, and addition of recursive feature elimination presented separately.*

## 6.4 Part 4: Feed-Forward Neural Network

Feed Forward Neural Network was selected as the fourth machine learning model in predicting water level at the Storå River within the scope of the research. This model has its own section because it was not included in the feature set experiment due to time constraints. It was presented as another machine learning algorithm and used to compare performance with other machine learning models presented in the previous part. The model was built using TensorFlow and Keras libraries as mentioned in Chapter 4.3.4. The hyperparameter tuning was implemented for the parameters activation, batch size, epochs and learning rate. Keras tuner was utilized through random search in hyperparameter tuning. For activation function relu and tanh functions introduced for input and hidden layers. For the output layer linear function was used. For the batch size default, doubled and halved values were introduced to the random search algorithm. For the learning rate 0.01, 0.001, 0.0001 were used. For the epochs 50 and 100 introduced to the random search algorithm. Early stopping criteria was implemented monitoring the validation loss. For the best model, the model loss plot is displayed in Figure 48. The training was forced to stop before reaching 70 epochs due to the early stopping mechanism.



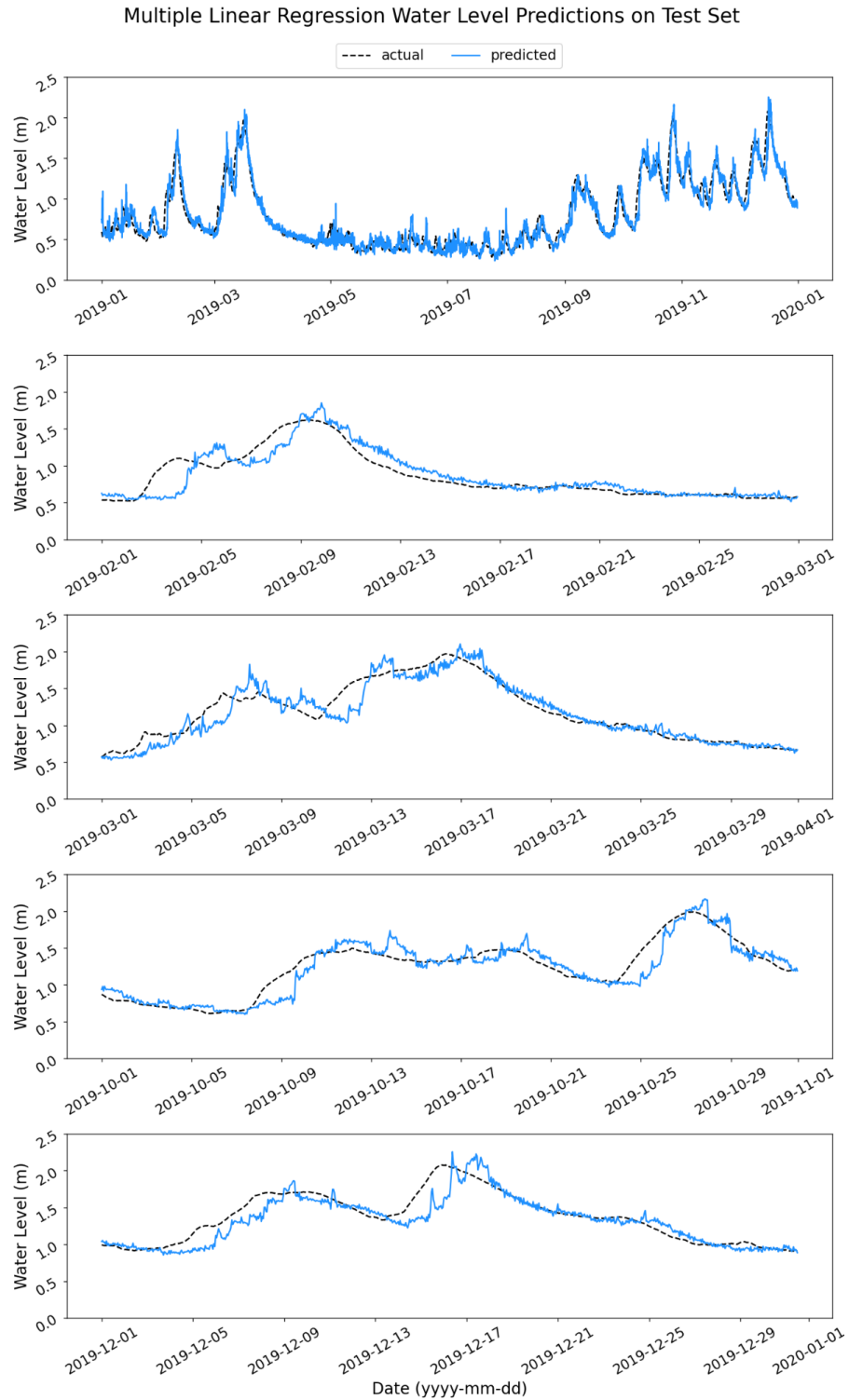
**Figure 48** | *Feed Forward Neural Network model loss*

The recursive feature elimination technique was not implemented for this model since the improvement was insignificant.

## 6.5 Part 5: Overall Assessment of Tested Methods

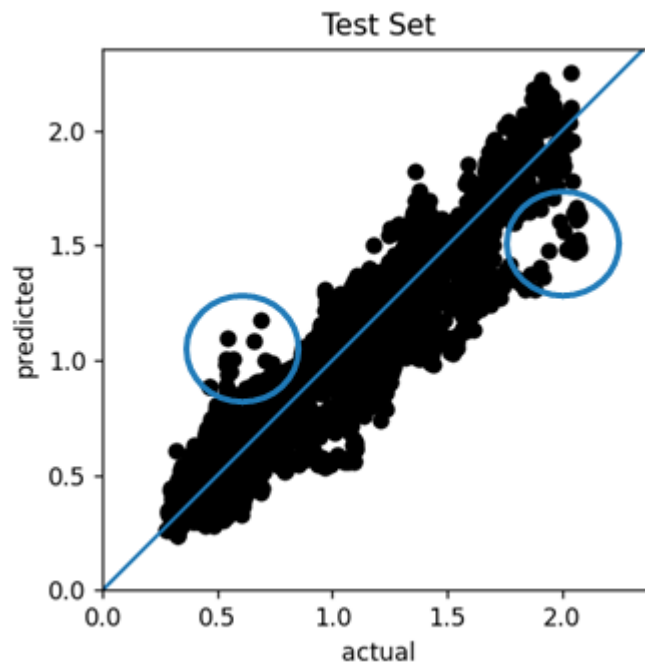
This section is dedicated to water level predictions of machine learning models: Multiple Linear Regression, Random Forest Regression, Gradient Boosting Regression and Feed Forward Neural Network. These models were then compared based on the evaluation criterias on the test set. The peak water levels observed in the time series were examined with special focus. According to the observed measures there were four peaks detected in the test set. The first peak was observed at the beginning of February at 1.621 m. The second peak is observed in the middle of March at 1.969 m. The third peak was observed towards the end of October at 1.99 m. Finally, the fourth peak was observed in the middle of December at 2.076 m. The machine learning models were examined based on how close their predictions to these peak values and the predefined evaluation criterias.

The water level prediction results from Multiple Linear Regression models with 48-hour lead time are presented in Figure 49. The first plot represents the result for the whole test set. There were four peaks present in the observed water level time series. In order to investigate the Multiple Linear Regression model's performance, all peaks were zoomed in and displayed separately. The actual water level was visualized with black dotted line and it followed a smooth curve since during preprocessing step water level data with 15-min interval converted into hourly mean data and the curve got smoothened. The peak predictions showed lag and were overestimated for all four zoomed-in plots. The predictions exhibited variations between peaks, and showed a smooth trend towards tail. The variations in the predicted water level can be explained by using 14 features during the training of the machine learning model. The mean absolute error was calculated as 8.244 cm, for validation it was 8.130 cm. The improvement in the mean absolute error according to the persistence model recorded as 3.992 cm. The calculated root mean square error reported as 11.602 cm for the test set, 11.580 cm for the validation set and the coefficient of determination reported as 0.921 for the test set, and 0.866 for the validation set.



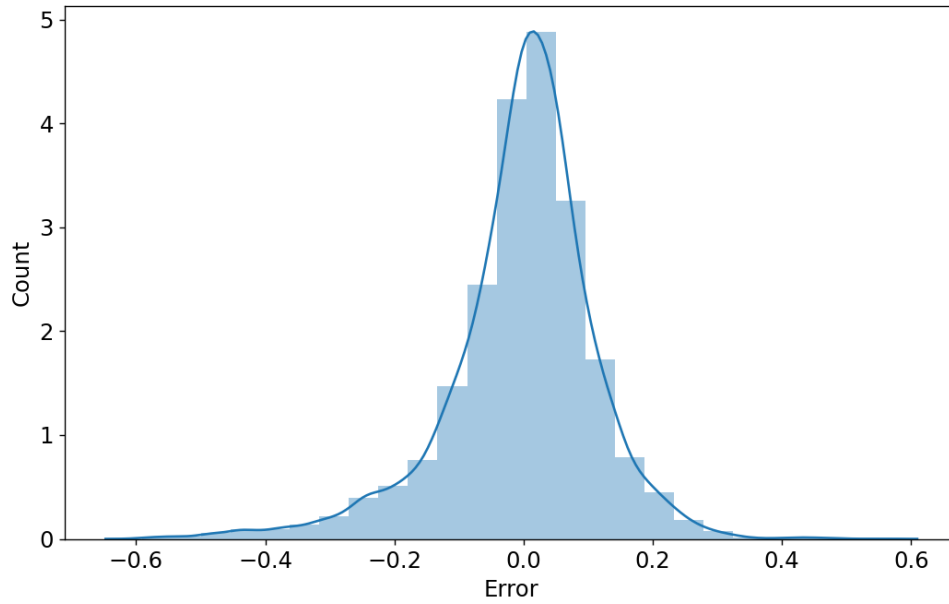
**Figure 49** | *Multiple Linear Regression water level prediction results on the test set with 48-hour lead time. First plot represents the whole test data set and the others are the zoomed-in versions representing the prediction performance of the machine learning model for peak values.*

The deviations from the diagonal for actual versus the predicted values in the Multiple Linear Regression model are presented in Figure 50. The blue line represents the diagonal which shows the perfect prediction. Depending on where the black dot lies, the prediction power of the Multiple Linear Regression model can be interpreted. For a good model it is expected to see symmetric scattered black points around the blue diagonal line. The plot shows overall good results with a narrow deviation band around the diagonal. For some low water levels overprediction was encircled on the left hand side and some high water level underpredictions were observed in the encirclement on the right hand side of the figure.



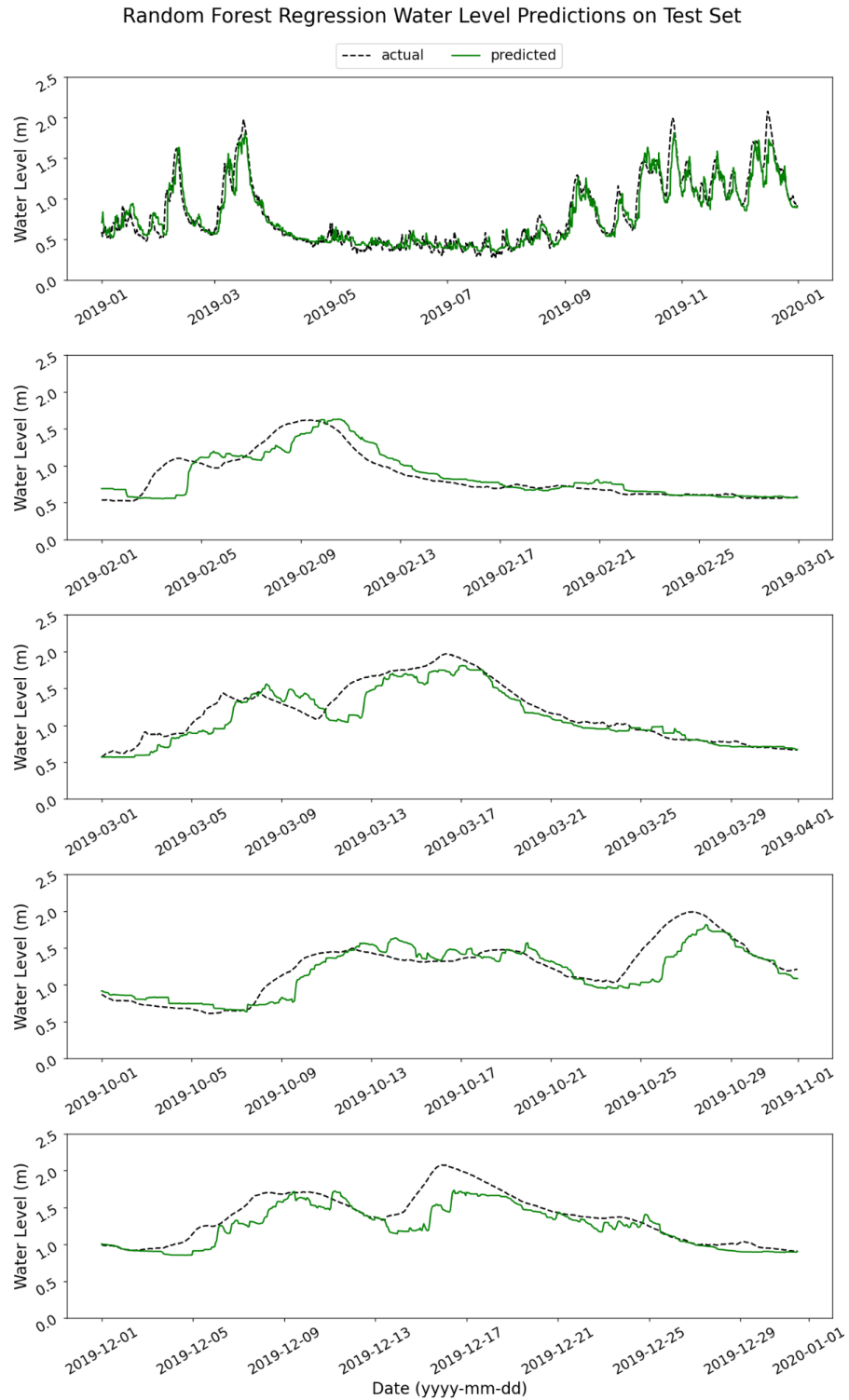
**Figure 50** | *Actual vs. predicted values on the test set for the Multiple Linear Regression model presented with the black dots. The diagonal is presented with a blue line at 45 degrees. The blue circles represent overfitting on the left side and underfitting on the right side.*

The error distribution in the test data set for the Multiple Linear Regression is presented in Figure 51. The error term represents the differences between actual and predicted values of data. The most error terms were clustered around the mean of zero with asymptotic tails both left and right. The distribution follows a Gaussian-like distribution with calculated skewness of -0.873. This shows the linear regression algorithm made almost adequate inferences. Lower the skewness at the error distribution better the predictive performance of the Multiple Linear Regression model.



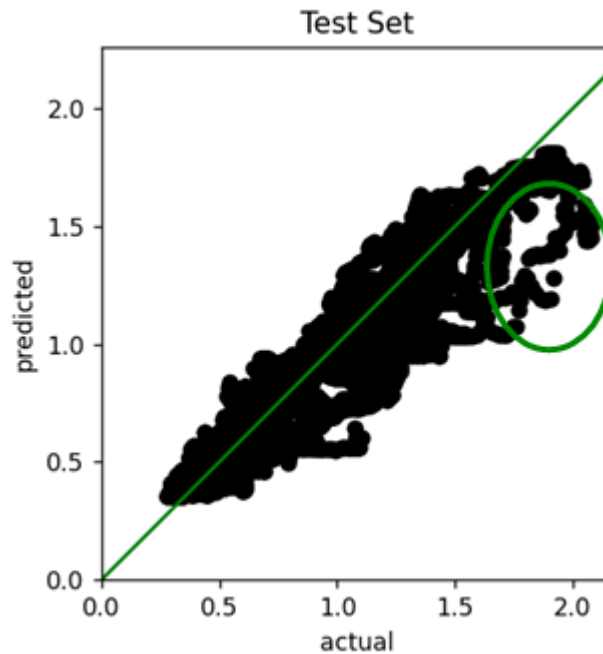
**Figure 51** | *Multiple Linear Regression error term distribution on the test set.*

The water level prediction results from the Random Forest Regression model with 48-hour lead time are presented in Figure 52. The first plot represents the whole time series for the test set, and the others are zoomed-in to the observed peak water levels in the time series. Overall, the Random Forest model was successful in the low water level predictions. For the first and the third peaks, lagged predictions were captured. For the second and fourth peaks, the water level predictions were underestimated. The fluctuations of the predicted water level is decreased compared to the Multiple Linear Regression model due to the decrease in the number of features participated in training of the model. The mean absolute error was recorded as 9.205 cm which is 0.961 cm higher than the Multiple Linear Regression model's result. Root mean squared error calculated as 13.208 cm and coefficient of determination calculated as 0.897.



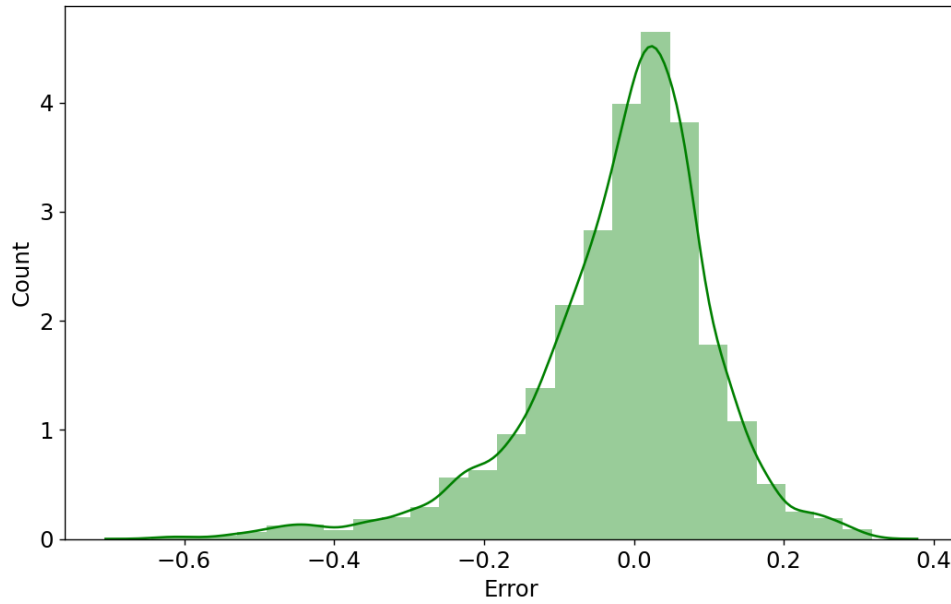
**Figure 52** | *Random Forest Regression water level prediction results on the test set with 48-hour lead time. First plot represents the whole test data set and the others are the zoomed-in versions representing the prediction performance of the machine learning model for peak values.*

In Figure 53, the actual versus predicted values for the Random Forest Regression model is displayed. Based on the deviations from the diagonal presented with the green line, the model showed a narrow deviation band in predicting lower water levels which can be interpreted as the model was more successful for those levels. Underestimation on the peak values can be deduced from the predictions pointed with the green circle at the bottom left side of the diagonal.



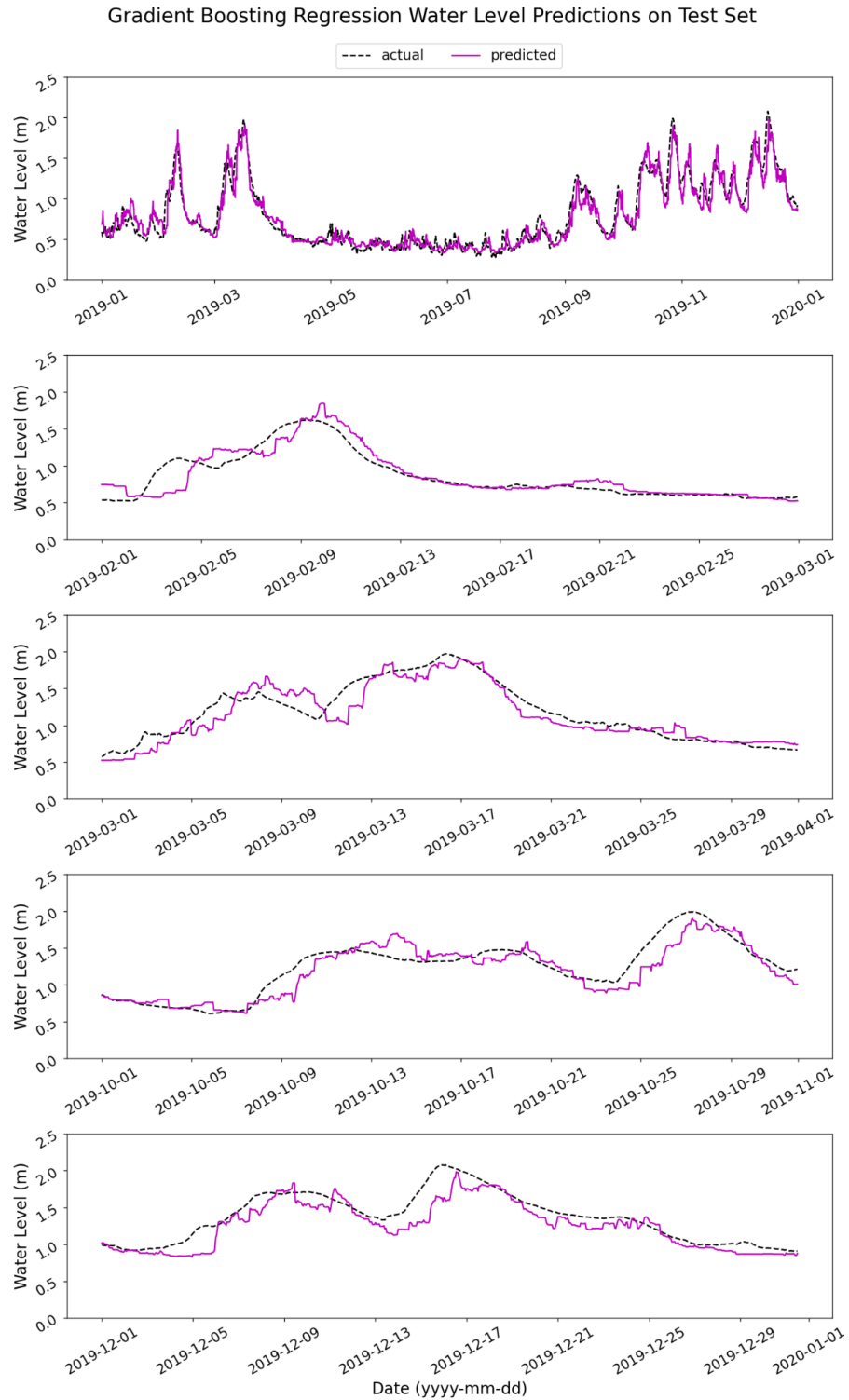
**Figure 53** | *Actual vs. predicted values on the test set for the Random Forest Regression model presented with the black dots. The diagonal is presented with a green line at 45 degrees. The green circle on the right side represents underfitting.*

The error distribution in the test data set for the Random Forest Regression is presented in Figure 54. The error term was calculated similarly to the Multiple Linear Regression model by taking the differences in between actual and the predicted values of data. The distribution resembles a Gaussian-like distribution and exhibits negative skewness with a value of -1.009. The most error terms were accumulated around the mean of zero and the left tail showed a longer distribution.



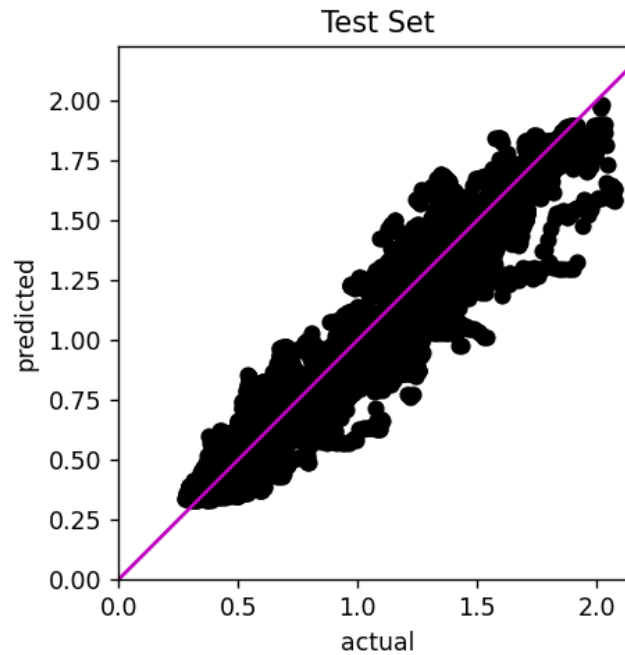
**Figure 54** | *Random Forest Regression error term distribution on the test set.*

The water level prediction results from the Gradient Boosting Regression model with 48-hour lead time are presented in Figure 55 for the whole testing data set and four predefined peaks separately. The model captured the peaks with lag. In the first peak, overestimation was observed. For the rest of the peaks the predictions were slightly underestimated. Fluctuations in the observed water level were decreased compared to the Multiple Linear Regression model due to utilization of less number of features during the training of the Random Forest model. The mean absolute error was recorded as 8.511 cm which is 0.267 cm higher than the Multiple Linear Regression model's result. The root mean square error was 11.602 cm and coefficient of determination was 0.921.



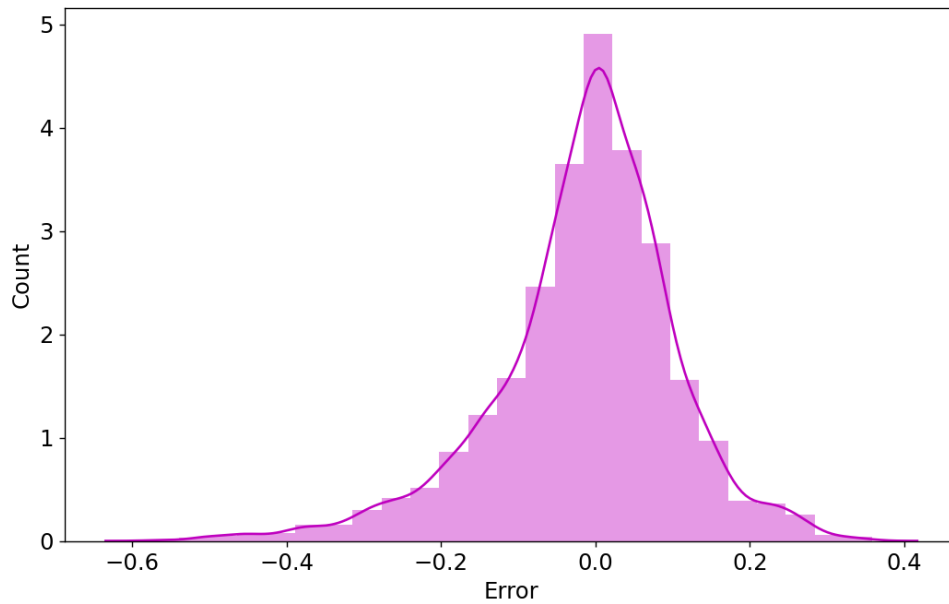
**Figure 55** | *Gradient Boosting Regression water level prediction results on the test set with 48-hour lead time. First plot represents the whole test data set and the others are the zoomed-in versions representing the prediction performance of the machine learning model for peak values.*

The actual versus the predicted plot for the Gradient Boosting Regression model is presented in Figure 56. Compared to the first two models predicted versus actual plots, the Gradient Boosting model showed less deviation from the diagonal presented with the purple line. Although it showed less variation compared to other models, it was still better in predicting lower water level values than the higher water level values which can be deduced from the disorientation of the black dots towards the upper right side of the plot.



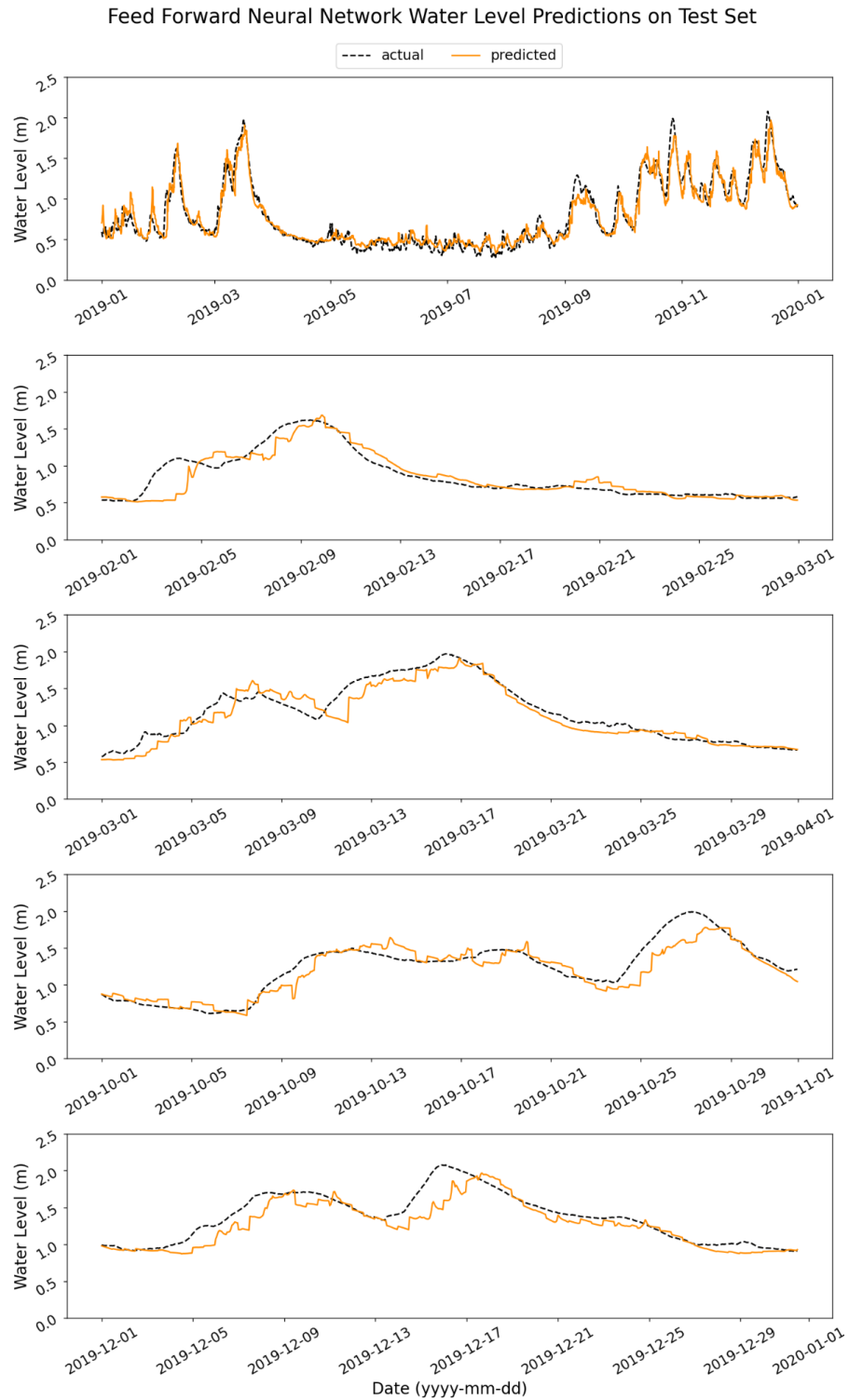
**Figure 56** | *Actual vs. predicted values on the test set for the Gradient Boosting Regression model presented with the black dots. The diagonal is presented with a purple line at 45 degrees.*

The error distribution in the test data set for the Gradient Boosting Regression model with 48-hour lead time is presented in Figure 57. The distribution follows a Gaussian-like distribution with a moderate negative skewness of -0.745. Again, the most of the residuals accumulated around the mean value of zero, the longer left tail shows the Gradient Boosting Regression model's predicted values were lower than the actual values, since residuals calculated the predicted value minus the actual value.



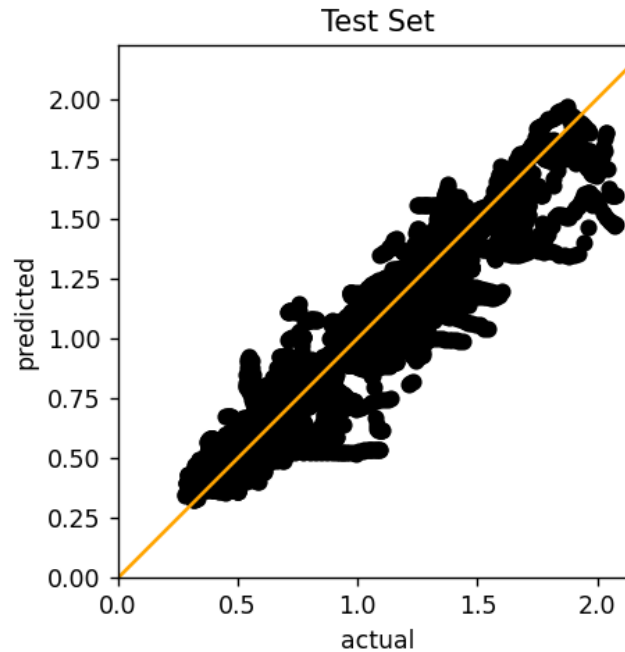
**Figure 57** | *Gradient Boosting Regression error term distribution on the test set.*

The water level prediction results from the Feed Forward Neural Network model with 48-hour lead time are displayed in Figure 58. The predictions for the first two peaks were reasonable, yet predictions for the last two peaks were captured with lag. The water level predictions exhibited variations among peaks, and showed a smooth trend towards tails. The predictions exhibit less variations compared to the Multiple Linear Regression model, although more features were used in training the Feed Forward Neural Network model. This can be explained by the usage of the activation function which can smooth out changes in features during training so that the effect on the target variable gets smaller. The mean absolute error was recorded as 8.242 cm which is 0.003 cm better than the Multiple Linear Regression model's result. The root mean square error was 11.541 cm and coefficient of determination was 0.922.



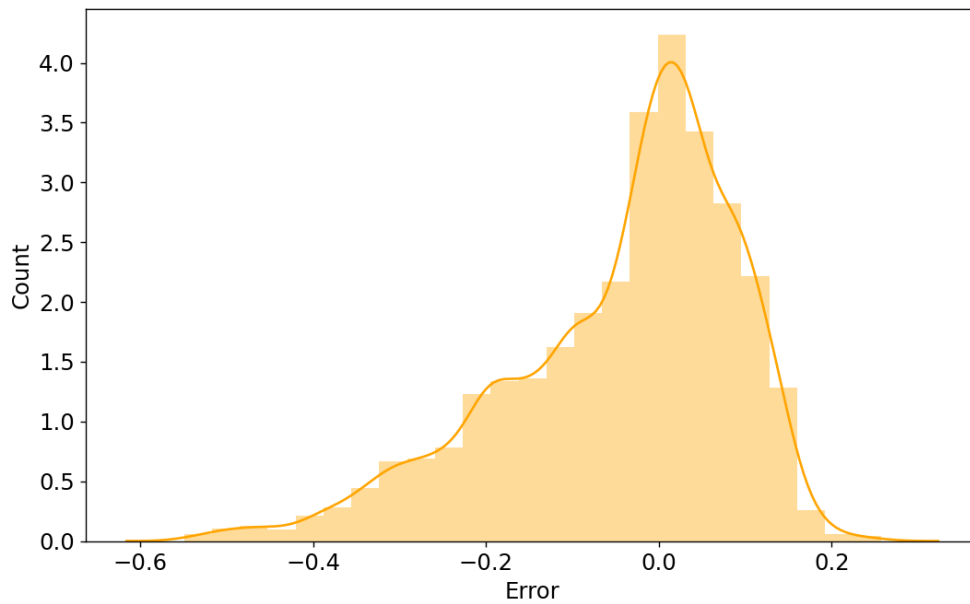
**Figure 58** | *Feed Forward Neural Network water level prediction results on the test set with 48-hour lead time. First plot represents the whole test data set and the others are the zoomed-in versions representing the prediction performance of the machine learning model for peak values.*

The actual vs. predicted plot for the Feed Forward Neural Network is presented in Figure 59. Like in the Gradient Boosting model, the Feed Forward Neural Network showed less deviation from the diagonal presented with the orange line. Like previous three models, this model performed better at lower water level predictions as well. When it comes to higher water levels, some underestimation was observed.



**Figure 59** | *Actual vs. predicted values on the test set for the Feed Forward Neural Network model presented with the black dots. The diagonal is presented with an orange line at 45 degrees.*

The error distribution in the test data set for the Feed Forward Neural Network is presented in Figure 60. The distribution follows the Gaussian-like distribution with a moderate negative skewness of -0.916. Again, the most of the residuals accumulated around the mean value of zero, the longer left tail shows the Feed Forward Neural Network model's predicted values were lower than the actual values, since residuals calculated the predicted value minus the actual value.



**Figure 60** | *Feed Forward Neural Network error term distribution on the test set.*

In summary, the Feed Forward Neural Network model predicted the water level with the least mean absolute error and root mean square error terms and the highest coefficient of correlation. In terms of the mean absolute error and coefficient of determination, the Feed Forward Neural Network model was followed by the Multiple Linear Regression, the Gradient Boosting Regression, and the Random Forest Regression in sequence. In terms of root mean square error, the Gradient Boosting performed slightly better than the Multiple Linear Regression model. Among all, the Random Forest Regression model underperformed. In Table 12, summarization of all four machine learning models is presented.

**Table 12** | *Prediction results of the all machine learning models on training and test dataset*

	Highest				Lowest			
Accuracy								
Evaluation Criteria	Train				Test			
	MLR	RF	GB	FFNN	MLR	RF	GB	FFNN
MAE	0.060	0.057	0.046	0.052	0.082	0.092	0.085	0.082
RMSE	0.086	0.084	0.065	0.079	0.116	0.132	0.116	0.115
R <sup>2</sup>	0.940	0.944	0.966	0.950	0.921	0.897	0.921	0.922

According to Table 12, all the machine learning methods' performances were lower on the test set than the training set which can be an indicator of overfitting. In fact it is very common to see better performance on the training set. During this research some techniques were applied such as hyperparameter tuning through randomized search, introducing k-fold cross validation, utilizing the recursive feature elimination technique to decrease the overfitting on the training data. After all the models are overfitting way less yet there is still room for improvement.

## Chapter 7. Conclusion and Recommendations

*This final chapter summarizes all the findings of this research, draws conclusions based on the findings, and answers the research questions proposed in the first chapter. Moreover, the limitations encountered during the research work are discussed with possible solutions. Finally, the recommendations for future research are presented.*

### 7.1 Main Conclusions

The objective of this research was to study river stage forecasting using different machine learning models and to achieve a 48-hour lead time for the Storå River, Denmark. Creating a proper forecasting model for this area can contribute to impact assessment of potential riverine flood, flood risk management, dissemination of information and warning messages, and enhancing early warning systems as explained in Chapter 1.

In Chapter 2, a review of related literature was presented in the context of the study objectives. The next chapter was devoted to acquiring general information about the site and the data of the case study area. In Chapter 4, the research methodology and corresponding theory behind each procedure were discussed. In the scope of this thesis work, the total number of machine learning algorithms applied was selected as four and the thinking process explained in Chapter 4.3. It started from a simple model and complexity added up both for machine learning algorithms and the data have been utilized throughout the research project.

Sound data analysis was considered as the pillar for this research. Well prepared data is the reason why machine learning algorithms can learn and develop themselves. Therefore, Chapter 5 was dedicated to data analysis. The data was acquired from several sources as described in Table 1. After the visualization of data, some missing values, erroneous information, and outliers were detected. Hence, in the preprocessing step several fast missing value imputation techniques were considered. Although the selected approach was not the best fit for observed precipitation data, no further missing value imputation techniques were performed considering the objectives of the research. The erroneous information was replaced and the outliers were removed. Afterwards, the feature selection was performed through the filter method. Data splitting was then performed considering the physical and statistical properties of the time series. Thereafter, several data transformation and scaling techniques were applied to the dataset. The aim of this chapter was bringing the data to ready-to-use format by the machine learning algorithms.

The research questions were answered in Chapter 6 together with a detailed discussion of this research work. The impact on the machine learning model's prediction performance considering scaling and transformation was elaborated in 6.1 in order to answer the first research question. Considering the importance of standard normal distribution in statistics and in machine learning, the features were transformed to follow normal or normal-like distributions. However, according to the results the transformed data slightly underperformed. During this analysis, the results of scaled data using normalization and standardization techniques were compared as well. Tree based algorithms are scale-invariant and the results showed that this to be true. The slight changes observed in the mean absolute error stemmed from the randomized search hyperparameter tuning. As a result, the normalization technique applied on the raw data gave the lowest mean absolute error for all three models and decided to utilize moving forward.

The detailed analysis of machine learning models' prediction performances using different feature sets was presented in Chapter 6.2. Feature sets were created from three features, expanding to nineteenth features. It was observed that with the addition of features, the Multiple Linear Regression, the Random Forest Regression and the Gradient Boosting Regression models were inclined to perform better. For sure some feature sets influenced the performance of machine learning models more than others. Addition of forecasted precipitation variables in feature set 3 and water level measurements coming from Skærum Bro Station in feature set 4 made the highest influence. The reason why the first two feature sets were included in the analysis was to observe the ability of machine learning models to perform with limited information. Water level data has low variation, thus, even without a forecasted precipitation it can be still expected to have good predictions to some degree especially during dry weather conditions. Keeping the first two feature sets helped to quantify how much the addition of the forecasted precipitation data improves the prediction performance of the machine learning models.

The results of correlation analysis and the mutual information revealed that precipitation has the least importance among all features on the future water level predictions. The correlation analysis concluded water level measures at Skærum Bro Station had the highest score. On the other hand, mutual information analysis concluded simulated discharge at Skærum Bro Station had the highest score as described in Chapter 5. Additionally, recursive feature elimination method was explored during feature selection and it is concluded even after dropping approximately 74% of the features ergo information, Random Forest and Gradient Boosting models continued to perform. In fact, the predictive performance for hereinabove models performed slightly better as elaborated in Chapter 6.3.2. Thus, combining filter and wrapper methods improved the machine learning prediction performance as answering the third research question.

For the last research question, comparing the predictive performance of machine learning models, it can be concluded that each model had some advantages and some flaws. In terms of mean absolute error, the Feed Forward Neural Network performed slightly better. However, the difference between the Multiple Linear Regression model was only 0.003 cm. Previous research showed that Feed Forward Neural Networks were favorable in prediction and forecasting in general. Although this study confirmed that, it

argues when the time, resources, or expert knowledge is limited, the Multiple Linear Regression model can be thought as an alternative instead of the Feed Forward Neural Network model considering the almost insignificant mean absolute error term in between. The Random Forest Regression model underperformed among all other models.

Another interference from this research is that although the Random Forest Regression model can deal with nonlinearity in the data, work well with the huge dataset, generate good predictions, and handle missing data, it poses a serious difficulty when encountering unseen data. It sticks to the range of training data and cannot extrapolate outside of that range. In other words, predicting a target variable based on feature values that are outside of the range of the original training dataset, the Random Forest will assume the target variable will be around the largest number in the training set because, in the Random Forest the trees are created based on the training data. The reason why validation errors were considerably high for the Random Forest as presented in chapter 6.2, data split for the training and validation sets have different ranges of data for some features. This is applicable for the Gradient Boosting Regression as well, since the algorithm is tree based. Contrarily, the linear regression and neural networks can extrapolate. It can also be deduced from this research that the Random Forest Regression model is highly inclined to overfit on the training dataset unless hyperparameter tuning is performed. Although both are tree based machine learning algorithms, the Gradient Boosting did not suffer from the same problem as displayed in Figure 46.

The approach through predicting water level can be utilized in other sites. It was a good practice to start with the Multiple Linear model due to its simplicity and time efficiency. Tuning hyperparameters and feature selection through recursive feature elimination method proved their worth in improving the predictive performance of machine learning models.

In this research machine learning models were preferred in predicting water level at the Storå River over numerical models. Although the numerical model is robust and aids in managing the basin but may not be very accurate in specific locations. This research focuses on building locally accurate machine learning models that may complement the numerical model in managing the basin by providing locally accurate water level forecasting.

## 7.2 Limitations and Recommendations

### 7.2.1 Limitations

There were some limitations encountered during this research. The first limitation was the gap in the TIGGE data for the forecasted precipitation. This led to interruptions in the dataset and some problems in data splitting. As discussed in the conclusion, having different data ranges for training, validation and test data directs to the unfavorable predictions by the tree based algorithms. Solution for this limitation using forecasted precipitation data from local sources can be considered.

Missing value imputation for the historical precipitation by replacing them with zero led to an underestimation for the yearly accumulated precipitation in years 2014 and 2016. Even though these variables were eliminated after the filter methods and were not used in the machine learning models it is important to bring it as a limitation. In fact, the reason why those precipitation features were not included in the model might be underestimated missing value imputation. Solution for this limitation can be using machine learning algorithms in missing value imputation or coming up with new historical precipitation sources that can be used to merge data.

Another limitation about missing value imputation, the imputation with linear interpolation technique would not be possible if the forecasting is on real time since the next non-missing value is not observed yet. The solution for this can utilize other techniques in imputing missing values that does not require the usage of upcoming data to fill the missing value.

The time was a big constraint for this research. During randomized search only 5-fold cross validation and 10 iterations were employed due to time limitation. Although this helped to decrease the overfitting of the machine learning models at a certain limit, it can be improved further by increasing the number of folds in cross validation and the iterations. It is important to note that the improvement does not follow a linear pattern. In fact, for the number of folds in cross validation the optimum number is required to be found, otherwise increasing the folds leads the machine learning models to work with smaller training sets in each iteration which lowers the bias and increases run time and variance of the estimate. Increasing the number of iterations forces the machine learning model to try more combinations from the grid of hyperparameter values which would be a potential time burden. Although increasing these numbers has potential to improve the prediction accuracy of the machine learning models, it should be done carefully in order not to end up with worse estimations and high model execution time.

Usage of the correlation analysis as a filter method might pose a limitation. If a feature has a non-linear relationship with the target variable but not a strong linear one, the feature selection through the correlation analysis considers eliminating that feature due to low correlation score, although it could be useful for the non-linear methods.

### 7.2.2 Recommendations

There are some aspects that can be explored for future works and have potential to improve this research work further. Addition of new features can be counted as one of the aspects that can be explored further. Addition of temperature, evapotranspiration, irrigation supply, runoff, topography, land use data can be counted as some of them. Besides, historical precipitation data from a different source would help to increase the accuracy as well. Local observation centers can be used for this purpose. Moreover, as a new popular technique, estimating the precipitation level through radar which provides high spatial and temporal resolution can be employed for this purpose (Kreklow, 2020).

As another recommendation, it was demonstrated in the literature that the machine learning model's performance could be improved through hybridization with other machine learning algorithms, soft computing techniques, numerical simulations, and/or physical models (Mosavi et al., 2018). There have been several studies done creating hybrid models by combining machine learning models and physical models (Farfán et al., 2020; Hosseiny et al. 2020). These implementations supply higher robustness and efficiency to the model in predicting complex hydrologic behavior which can be considered as future work for this research.

Improving the missing value imputation techniques would help with the performance of the machine learning models. In this research only fast imputation techniques were harnessed. Several researchers have demonstrated the utilization of machine learning models in missing value imputation resulted with trustable accuracy (Gill et al., 2007; Petty and Dhingra, 2017). Employing machine learning algorithms to get better estimation for missing values of water level and precipitation parameters can be investigated further. Furthermore, changing the temporal resolution from hourly to daily would smoothen the variations in the data which generates an advantage on missing value imputation especially for precipitation data. As Hema and Kant mentioned, hourly precipitation observations exhibit very high variation (Hema and Kant, 2017) and this creates a challenge for the imputation of the missing values. Converting it to daily makes the missing value computation easier which improves the prediction performance of the machine model as well.

The station selected for water level was not highly influenced by the tidal effect of the Nisum Fjord. The tidal influence in the selected water level station is very small during very few days in the year. For the future works it would be interesting to use one station that is further down closer to the fjord that has a very clear tidal effect in order to investigate the tidal effect on the river. The tidal influence is also an important phenomena to consider in case the findings of this research work would be utilized in other sites that are influenced by the tides.

Creating a stacked regression model would help to achieve more accurate predictions than a single model can perform. The stacking regression constitutes a linear combination of different machine learning models. In basic terms, the predictions coming from one model are used for another model's input (Breiman, 1996; Pavlyshenko 2018). After going through all the improvements presented, a stacked regression model can be created by using the presented models and several others to achieve better predictions.

## References

- Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., ... Zheng, X. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.
- Abrahart, R. J., Heppenstall, A. J., & See, L. M. (2007). Timing error correction procedure applied to neural network rainfall—runoff modelling. *Hydrological Sciences Journal*, 52(3), 414–431. doi:10.1623/hysj.52.3.414
- Alvisi, S., Mascellani, G., Franchini, M., & Bárdossy, A. (2006). Water level forecasting through fuzzy logic and artificial neural network approaches. *Hydrology and Earth System Sciences*, 10(1), 1-17. doi:10.5194/hess-10-1-2006
- Alloghani M., Al-Jumeily D., Mustafina J., Hussain A., Aljaaf A.J. (2020). A Systematic Review on Supervised and Unsupervised Machine Learning Algorithms for Data Science. In: Berry M., Mohamed A., Yap B. (eds) Supervised and Unsupervised Learning for Data Science. Unsupervised and Semi-Supervised Learning. Springer, Cham. [https://doi.org/10.1007/978-3-030-22475-2\\_1](https://doi.org/10.1007/978-3-030-22475-2_1)
- Aqil, M., Kita, I., Yano, A., & Nishiyama, S. (2007). A comparative study of artificial neural networks and neuro-fuzzy in continuous modeling of the daily and hourly behaviour of runoff. *Journal of Hydrology*, 337(1-2), 22–34. doi:10.1016/j.jhydrol.2007.01.013
- Asrol M., Papilo P., Gunawan F.E. (2021). Support Vector Machine with K-fold Validation to Improve the Industry's Sustainability Performance Classification. *Procedia Computer Science*, 179. (pp. 854-862). ISSN 1877-0509. <https://doi.org/10.1016/j.procs.2021.01.074>.
- Atiquzzaman, M., & Kandasamy, J. (2015). Prediction of hydrological time-series using extreme learning machine. *Journal of Hydroinformatics*, 18(2), 345–353. doi:10.2166/hydro.2015.020
- Bae, D.-H., Jeong, D. M., & Kim, G. (2007). Monthly dam inflow forecasts using weather forecasting information and neuro-fuzzy technique. *Hydrological Sciences Journal*, 52(1), 99–113. doi:10.1623/hysj.52.1.99
- Ben Brahim, A., & Limam, M. (2016). A hybrid feature selection method based on instance learning and cooperative subset search. *Pattern Recognition Letters*, 69, 28–34. doi:10.1016/j.patrec.2015.10.0
- Berrar, D. (2018). Cross-Validation. Reference Module in Life Sciences. doi:10.1016/b978-0-12-809633-8.20349-x

- Bi, Y., Xue, B., & Zhang, M. (2021). Genetic programming for image classification: An automated approach to feature learning. <https://doi.org/10.1007/978-3-030-65927-1>.
- Bock, S., Goppold, J., & Weiß, Martin. (2018). An improvement of the convergence proof of the ADAM-Optimizer. arXiv:1804.10587
- Bonaccorso, G. (2017). Machine Learning Algorithms: Popular algorithms for data science and machine learning. Packt.
- Borra, S., & Di Ciaccio, A. (2010). Measuring the prediction error. A comparison of cross-validation, bootstrap and covariance penalty methods. *Computational Statistics & Data Analysis*, 54(12), 2976–2989. doi:10.1016/j.csda.2010.03.00
- Breiman, L. (1996). Stacked regressions. *Mach Learn* 24, 49–64. <https://doi.org/10.1007/BF00117832>
- Breiman, L. (2001). *Machine Learning*, 45(1), 5–32. doi:10.1023/a:1010933404324
- Bronstert, A., Creutzfeldt, B., Graeff, T., Hajnsek, I., Heistermann, M., Itzerott, S., ... Zehe, E. (2011). Potentials and constraints of different types of soil moisture observations for flood simulations in headwater catchments. *Natural Hazards*, 60(3), 879–914. doi:10.1007/s11069-011-9874-9
- Casper, M., Gemmar, P., Gronz, O., Johst, M., & Stüber, M. (2007). Fuzzy logic-based rainfall—runoff modelling using soil moisture measurements to represent system state. *Hydrological Sciences Journal*, 52(3), 478–490. doi:10.1623/hysj.52.3.478
- Chang, F.-J., Chang, L.-C., & Wang, Y.-S. (2007). Enforced self-organizing map neural networks for river flood forecasting. *Hydrological Processes*, 21(6), 741–749. doi:10.1002/hyp.6262
- Chang, F.-J., & Chen, Y.-C. (2003). Estuary water-stage forecasting by using radial basis function neural network. *Journal of Hydrology*, 270(1-2), 158–166. doi:10.1016/s0022-1694(02)00289-5
- Chau, K. W. (2007). A split-step particle swarm optimization algorithm in river stage forecasting. *Journal of Hydrology*, 346(3-4), 131–135. doi:10.1016/j.jhydrol.2007.09.004
- Chen, K.-Y., & Wang, C.-H. (2007). A hybrid SARIMA and support vector machines in forecasting the production values of the machinery industry in Taiwan. *Expert Systems with Applications*, 32(1), 254–264. doi:10.1016/j.eswa.2005.11.027
- Cheng, G.-J., Cai, L., & Pan, H.-X. (2009). Comparison of Extreme Learning Machine with Support Vector Regression for Reservoir Permeability Prediction. 2009 International Conference on Computational Intelligence and Security. doi:10.1109/cis.2009.124
- Cheng, K.-S., Lien, Y.-T., Wu, Y.-C., & Su, Y.-F. (2016). On the criteria of model performance evaluation for real-time flood forecasting. *Stochastic Environmental Research and Risk Assessment*, 31(5), 1123–1146. doi:10.1007/s00477-016-1322-7

Chen, W., Li, Y., Xue, W., Shahabi, H., Li, S., Hong, H., ... Bin Ahmad, B. (2019). Modeling flood susceptibility using data-driven approaches of naïve Bayes tree, alternating decision tree, and random forest methods. *Science of The Total Environment*, 134979. doi:10.1016/j.scitotenv.2019.134979

Chollet, F. (2015) keras, GitHub. <https://github.com/fchollet/keras>

Conrads, P.A., and Roehl, E.A., Jr., 2007, Hydrologic record extension of water-level data in the Everglades Depth Estimation Network (EDEN) using artificial neural network models, 2000–2006: U.S. Geological Survey Open-File Report 2007-1350, 56 p. (only online at <http://pubs.water.usgs.gov/ofr2007-1350>)

Corzo, G., & Solomatine, D. (2007). Baseflow separation techniques for modular artificial neural network modelling in flow forecasting. *Hydrological Sciences Journal*, 52(3), 491–507. doi:10.1623/hysj.52.3.491

Cunningham, S. C., Griffioen, P., White, M. D., & Nally, R. M. (2017). Assessment of ecosystems: A system for rigorous and rapid mapping of floodplain forest condition for Australia's most important river. *Land Degradation & Development*, 29(1), 127–137. doi:10.1002/ldr.2845

De Paiva, R. C. D., Buarque, D. C., Collischonn, W., Bonnet, M.-P., Frappart, F., Calmant, S., & Bulhões Mendes, C. A. (2013). Large-scale hydrologic and hydrodynamic modeling of the Amazon River basin. *Water Resources Research*, 49(3), 1226–1243. doi:10.1002/wrcr.20067

DHI Water and Environment. 1999. MIKE 11 Reference Manual.

Droutsas, K. G., Balaras, C. A., Lykoudis, S., Kontoyiannidis, S., Dascalaki, E. G., & Argiriou, A. A. (2020). Baselines for Energy Use and Carbon Emission Intensities in Hellenic Nonresidential Buildings. *Energies*, 13(8), 2100. doi:10.3390/en13082100

Efron, B., & Tibshirani, R. J. (1993). *An Introduction to the Bootstrap*. New York, NY: Chapman & Hall.

Elsafi, S. H. (2014). Artificial Neural Networks (ANNs) for flood forecasting at Dongola Station in the River Nile, Sudan. *Alexandria Engineering Journal*, 53(3), 655–662. doi:10.1016/j.aej.2014.06.010

European Commission. (2019). Commission staff working document First flood risk management plans - Member state: Denmark. Brussel, 26.2.2019. <https://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=SWD:2019:0060:FIN:EN:PDF>

European Environment Agency. (2010). Mapping the Impacts of Natural Hazards and Technological Accidents in Europe: An Overview of the Last Decade; Technical Report No 13/2010; EEA: Copenhagen, Denmark, ISSN 1725-2237. <https://www.eea.europa.eu/publications/mapping-the-impacts-of-natural>

European Environment Agency. (2017). Framework Service Contract EEA/IDM/15/026/LOT 2 for Services supporting the European Environment Agency's (EEA) implementation of cross-cutting activities for coordination of the in situ component of the Copernicus Programme Services. Doc. ID: EG-RPT-EEA-SC1-0020. <https://insitu.copernicus.eu/library/reports/CUF201716DigitalElevationModelReport.pdf>

European Union. (2007). Directive 2007/60/EC of the European Parliament and of the council of 23 October 2007 on the assessment and management of flood risks. Official Journal of the European Union, L 288/27 (6/11/2007). [https://www.bmu.de/fileadmin/Daten\\_BMU/Download\\_PDF/Binnengewasser/richtlinie\\_management\\_hochwasserrisiken\\_en.pdf](https://www.bmu.de/fileadmin/Daten_BMU/Download_PDF/Binnengewasser/richtlinie_management_hochwasserrisiken_en.pdf)

Farfán, J. F., Palacios, K., Ulloa, J., & Avilés, A. (2020). A hybrid neural network-based technique to improve the flow forecasting of physical and data-driven models: Methodology and case studies in Andean watersheds. *Journal of Hydrology: Regional Studies*, 27, 100652. doi:10.1016/j.ejrh.2019.100652

Feng, J. & Lu, S. (2019). Performance Analysis of Various Activation Functions in Artificial Neural Networks. *Journal of Physics: Conference Series*. 1237. 022030. 10.1088/1742-6596/1237/2/022030.

Figure 5. Graph of Recorded Water Level of Store River at Storebro Station, Holstebro. Holstebro Kommune. Retrieved from: [https://www.holstebro.dk/Files/Images/1-borger/natur-miljo/Aa\\_so\\_fjord/Oversvoemelse/Stor%C3%A5\\_%20graf%20af%20historiske%20oversv%C3%B8mmelser%20ved%20Storebro%20i%20Holstebro%20per%202014-12-2015.JPG](https://www.holstebro.dk/Files/Images/1-borger/natur-miljo/Aa_so_fjord/Oversvoemelse/Stor%C3%A5_%20graf%20af%20historiske%20oversv%C3%B8mmelser%20ved%20Storebro%20i%20Holstebro%20per%202014-12-2015.JPG)

Friedman, J. H. (1999). Greedy Function Approximation: A Gradient Boosting Machine. *The Annals of Statistics* 29(5), 1189–1232. (last modification: April, 19 2001) doi:10.1214/aos/1013203451

Gill, M. K., Asefa, T., Kaheil, Y., & McKee, M. (2007). Effect of missing data on performance of learning algorithms for hydrologic predictions: Implications to an imputation technique. *Water Resources Research*, 43(7). doi:10.1029/2006wr005298

Gudivada, Venkat & Apon, Amy & Ding, Junhua. (2017). Data Quality Considerations for Big Data and Machine Learning: Going Beyond Data Cleaning and Transformations. *International Journal on Advances in Software*. 10. 1-20.

Guyon, I., & Elisseeff, A. (2003). An Introduction of Variable and Feature Selection. *Journal of Machine Learning Research*. 3. 1157 - 1182  
doi:10.1162/153244303322753616.

Harmel, R. D., Smith, P. K., Migliaccio, K. W., Chaubey, I., Douglas-Mankin, K. R., Benham, B., ... Robson, B. J. (2014). Evaluating, interpreting, and communicating performance of hydrologic/water quality models considering intended use: A review and

recommendations. *Environmental Modelling & Software*, 57, 40–51.  
doi:10.1016/j.envsoft.2014.02.013

Hastie, T. J., Tibshirani, R. J., & Friedman, J. H. (2001). *The Elements of Statistical Learning*. New York, NY: Springer

Hema, N., & Kant, K. (2017). Reconstructing missing hourly real-time precipitation data using a novel intermittent sliding window period technique for automatic weather station data. *Journal of Meteorological Research*, 31(4), 774–790.  
doi:10.1007/s13351-017-6084-8

Holstebro Kommune, Teknik og Miljø. (2011). *Storå, oversvømmelser i Holstebro i 1970, 2007 og 2011* by Flemming Kofoed.

Holstebro Kommune, Teknik og Miljø. (2015). *Oversvømmelse i Holstebro fra Storå den 7. december 2015*.

Hoque, N., Bhattacharyya, D. K., & Kalita, J. K. (2014). MIFS-ND: A mutual information-based feature selection method. *Expert Systems with Applications*, 41(14), 6371–6385. doi:10.1016/j.eswa.2014.04.019

Hosseiny, H., Nazari, F., Smith, V., & Nataraj, C. (2020). A Framework for Modeling Flood Depth Using a Hybrid of Hydraulics and Machine Learning. *Scientific Reports*, 10(1). doi:10.1038/s41598-020-65232-5

Hughes, B., Bothe, S., Farooq, H., & Imran, A. (2019). Generative Adversarial Learning for Machine Learning empowered Self Organizing 5G Networks. 2019 International Conference on Computing, Networking and Communications (ICNC). doi:10.1109/iccnc.2019.8685527

Jebens, M., Sorensen, C., & Piontkowitz, T. (2016). Danish risk management plans of the EU Floods Directive. *E3S Web of Conferences*, 7, 23005.  
doi:10.1051/e3sconf/20160723005

Jia, Y., & Culver, T. B. (2006). Bootstrapped artificial neural networks for synthetic flow generation with a small data sample. *Journal of Hydrology*, 331(3-4), 580–590.  
doi:10.1016/j.jhydrol.2006.06.005

Jierula, A., Wang, S., OH, T.-M., & Wang, P. (2021). Study on Accuracy Metrics for Evaluating the Predictions of Damage Locations in Deep Piles Using Artificial Neural Networks with Acoustic Emission Data. *Appl. Sci.*, 11, 2314.  
<https://doi.org/10.3390/app11052314>

Jovic, A., Brkic, K., & Bogunovic, N. (2015). A review of feature selection methods with applications. 2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO).  
doi:10.1109/mipro.2015.7160458

- Kabir, S., Patidar, S., Xia, X., Liang, Q., Neal, J., & Pender, G. (2020). A deep convolutional neural network model for rapid prediction of fluvial flood inundation. *Journal of Hydrology*, 125481. doi:10.1016/j.jhydrol.2020.125481
- Kajornrit, J., K. W. Wong, and C. C. Fung, 2012: Estimation of missing precipitation records using modular artificial neural networks. *Neural Information Processing: Lecture Notes in Computer Science*. Huang, T. W., Z. G. Zeng, C. D. Li, et al., Eds., Springer, Berlin Heidelberg, 7666, 52–59, doi:10.1007/978-3-642-34478-7\_7
- Kearns M. (1988). Thoughts on Hypothesis Boosting, Unpublished manuscript (Machine Learning class project.) <https://www.cis.upenn.edu/~mkearns/papers/boostnote.pdf>
- Khashei, M., & Bijari, M. (2010). An artificial neural network (p,d,q) model for timeseries forecasting. *Expert Systems with Applications*, 37(1), 479–489. doi:10.1016/j.eswa.2009.05.044
- Kingma, D.P., Ba, J. (2015). Adam: A Method for Stochastic Optimization
- Kisi, O., & Kerem Cigizoglu, H. (2007). Comparison of different ANN techniques in river flow prediction. *Civil Engineering and Environmental Systems*, 24(3), 211–231. doi:10.1080/10286600600888565
- Kraskov, A., Stögbauer, H., & Grassberger, P. (2004). Estimating mutual information. *Physical Review E*, 69(6). doi:10.1103/physreve.69.066138
- Kreklow, Jennifer. (2020). Improving Usability of Weather Radar Data in Environmental Sciences: Potentials, Challenges, Uncertainties and Applications. doi:10.15488/10144.
- Kumar, S., & Chong, I. (2018). Correlation Analysis to Identify the Effective Data in Machine Learning: Prediction of Depressive Disorder and Emotion States. *International Journal of Environmental Research and Public Health*, 15(12), 2907. doi:10.3390/ijerph15122907
- Lal, T. N., Chapelle, O., Weston, J., & Elisseeff, A. (2006). Embedded Methods. *Studies in Fuzziness and Soft Computing*, 137–165. doi:10.1007/978-3-540-35488-8\_6
- Lazzeri, F. (2020). In *Machine learning for time series forecasting with python*. (42–43). Wiley.
- Leahy, P., Kiely, G., & Corcoran, G. (2008). Structural optimisation and input selection of an artificial neural network for river level prediction. *Journal of Hydrology*, 355(1-4), 192–201. doi:10.1016/j.jhydrol.2008.03.017
- Lee, H., and K. Kang, 2015: Interpolation of missing precipitation data using Kernel estimations for hydrologic modeling. *Adv.Meteor.*, 2015, 935868, doi:10.1155/2015/935868.

- Legates, D. R., & McCabe, G. J. (1999). Evaluating the use of “goodness-of-fit” Measures in hydrologic and hydroclimatic model validation. *Water Resources Research*, 35(1), 233–241. doi:10.1029/1998wr900018
- Li, B., & Cheng, C. (2014). Monthly discharge forecasting using wavelet neural networks with extreme learning machine. *Science China Technological Sciences*, 57(12), 2441–2452. doi:10.1007/s11431-014-5712-0
- Lima, A. R., Cannon, A. J., & Hsieh, W. W. (2016). Forecasting daily streamflow using online sequential extreme learning machines. *Journal of Hydrology*, 537, 431–443. doi:10.1016/j.jhydrol.2016.03.017
- Lin, J.-Y. Cheng, C.-T., & Chau, K.-W. (2006). Using support vector machines for long-term discharge prediction. *Hydrological Sciences Journal*, 51(4), 599–612. doi:10.1623/hysj.51.4.599
- Liong, S.-Y., Lim, W.-H., & Paudyal, G. N. (2000). River Stage Forecasting in Bangladesh: Neural Network Approach. *Journal of Computing in Civil Engineering*, 14(1), 1–8. doi:10.1061/(asce)0887-3801(2000)14:1(1)
- Ly, S., C. Charles, and A. Degré, 2011: Geostatistical interpolation of daily rainfall at catchment scale: The use of several variogram models in the Ourthe and Ambleve catchments, Belgium. *Hydrol. Earth. Syst. Sci.*, 15, 2259–2274, doi:10.5194/hess-15-2259-2011.
- Mayr, A, Binder, H., Gefeller, O., & Schmid, M. (2014). The Evolution of Boosting Algorithms . *Methods of Information in Medicine*, 53(06), 419–427. doi:10.3414/me13-01-0122
- Mosavi, A., Ozturk, P., & Chau, K. (2018). Flood Prediction Using Machine Learning Models: Literature Review. *Water*, 10(11), 1536. doi:10.3390/w10111536
- Paulescu, M., Paulescu E., Badescu V. (2021). Chapter 9 - Nowcasting solar irradiance for effective solar power plants operation and smart grid management. *Predictive Modelling for Energy Management and Power Systems Engineering*. Elsevier. ISBN 9780128177723. <https://doi.org/10.1016/B978-0-12-817772-3.00009-4>.
- Papacharalampous, G., Tyralis, H., Langousis, A., Jayawardena, A.W., Sivakumar, B., Mamassis, N., Montanari, A., Koutsoyiannis, D. Probabilistic Hydrological
- Pavlyshenko, B. (2018). Using Stacking Approaches for Machine Learning Models. *IEEE Second International Conference on Data Stream Mining & Processing (DSMP)*. (pp. 255-258). doi: 10.1109/DSMP.2018.8478522.
- Post-Processing at Scale: Why and How to Apply Machine-Learning Quantile Regression Algorithms. *Water* 2019, 11, 2126. <https://doi.org/10.3390/w11102126>

- Petty, T. R., & Dhingra, P. (2017). Streamflow Hydrology Estimate Using Machine Learning (SHEM). *JAWRA Journal of the American Water Resources Association*, 54(1), 55–68. doi:10.1111/1752-1688.12555
- Phan, T.-T.-H., & Nguyen, X. H. (2020). Combining Statistical Machine Learning Models with ARIMA for Water Level Forecasting: The Case of the Red River. *Advances in Water Resources*, 103656. doi:10.1016/j.advwatres.2020.103656
- Prasad, A. M., Iverson, L. R., & Liaw, A. (2006). Newer Classification and Regression Tree Techniques: Bagging and Random Forests for Ecological Prediction. *Ecosystems*, 9(2), 181–199. doi:10.1007/s10021-005-0054-1
- Qi, J., Du, J., Siniscalchi, S. M., Ma, X., & Lee, C.-H. (2020). On Mean Absolute Error for Deep Neural Network Based Vector-to-Vector Regression. *IEEE Signal Processing Letters*, 1–1. doi:10.1109/lsp.2020.3016837
- Ratan, S. K., Anand, T., & Ratan, J. (2019). Formulation of Research Question - Stepwise Approach. *Journal of Indian Association of Pediatric Surgeons*, 24(1), 15–20. [https://doi.org/10.4103/jiaps.JIAPS\\_76\\_18](https://doi.org/10.4103/jiaps.JIAPS_76_18)
- Raymaekers, J., & Rousseeuw, P.J. Transforming variables to central normality. *Mach Learn* (2021). <https://doi.org/10.1007/s10994-021-05960-5>
- Richter, K., Hank, T. B., Atzberger, C., & Mauser, W. (2011). Goodness-of-fit measures: what do they tell about vegetation variable retrieval performance from Earth observation data. *Remote Sensing for Agriculture, Ecosystems, and Hydrology XIII*. doi:10.1117/12.897980
- Ruder, S., (2016). An overview of gradient descent optimization algorithms. *ArXiv*. 1609.04747v.
- Sarker, I.H. (2021). Machine Learning: Algorithms, Real-World Applications and Research Directions. *SN COMPUT. SCI.* 2, 160. <https://doi.org/10.1007/s42979-021-00592-x>
- Sazli, M. (2006). A brief review of feed-forward neural networks. *Communications, Faculty Of Science, University of Ankara*. 50. 11-17. doi:10.1501/0003168.
- Singh, P., & Borah, B. (2013). Indian summer monsoon rainfall prediction using artificial neural network. *Stochastic Environmental Research and Risk Assessment*, 27(7), 1585–1599. doi:10.1007/s00477-013-0695-0
- SKlearn. (n.d.). sklearn.model\_selection.RandomizedSearchCV. [https://scikit-learn.org/stable/modules/generated/sklearn.model\\_selection.RandomizedSearchCV.html](https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.RandomizedSearchCV.html)

Sun, W., & Trevor, B. (2017). Combining k-nearest-neighbor models for annual peak breakup flow forecasting. *Cold Regions Science and Technology*, 143, 59–69. doi:10.1016/j.coldregions.2017.08.009

Sung, J., Lee, J., Chung, I.-M., & Heo, J.-H. (2017). Hourly Water Level Forecasting at Tributary Affected by Main River Condition. *Water*, 9(9), 644. doi:10.3390/w9090644

Sutton, C. D. (2005). Classification and Regression Trees, Bagging, and Boosting. *Data Mining and Data Visualization*, 303–329. doi:10.1016/s0169-7161(04)24011-1

Tang, Q. H., A. W. Wood, and D. P. Lettenmaier, 2009: Real-time precipitation estimation based on index station percentiles. *J.Hydrometeor.*, 10, 266–277, doi: 10.1175/2008JHM1017.1.

Thirumalaiah, K., & Deo, M. C. (1998). River Stage Forecasting Using Artificial Neural Networks. *Journal of Hydrologic Engineering*, 3(1), 26–32. doi:10.1061/(asce)1084-0699(1998)3:1(26)

Toth, E., Brath, A., & Montanari, A. (2000). Comparison of short-term rainfall prediction models for real-time flood forecasting. *Journal of Hydrology*, 239(1-4), 132–147. doi:10.1016/s0022-1694(00)00344-9

United Nations Office for Disaster Risk Reduction. (2015). The human cost of weather related disasters (1995-2005).

[https://www.unisdr.org/2015/docs/climatechange/COP21\\_WeatherDisastersReport\\_2015\\_FINAL.pdf](https://www.unisdr.org/2015/docs/climatechange/COP21_WeatherDisastersReport_2015_FINAL.pdf)

Wagenaar, D., Curran, A., Balbi, M., Bhardwaj, A., Soden, R., Hartato, E., ... Lallemand, D. (2020). Invited perspectives: How machine learning will change flood risk and impact assessment. *Natural Hazards and Earth System Sciences*, 20(4), 1149–1161. doi:10.5194/nhess-20-1149-2020

Wang, W., Gelder, P. H. A. J. M. V., Vrijling, J. K., & Ma, J. (2006). Forecasting daily streamflow using hybrid ANN models. *Journal of Hydrology*, 324(1-4), 383–399. doi:10.1016/j.jhydrol.2005.09.032

Wang, Z., Lai, C., Chen, X., Yang, B., Zhao, S., & Bai, X. (2015). Flood hazard risk assessment model based on random forest. *Journal of Hydrology*, 527, 1130–1141. doi:10.1016/j.jhydrol.2015.06.008

Williams, M., Gomez Grajales, C. & Kurkiewicz, D. (2013). Assumptions of Multiple Regression: Correcting Two Misconceptions. *Practical Assessment, Research & Evaluation*. 18(11). ISSN 1531-7714.

World Health Organization. (2013). Floods in the WHO European Region: Health effects and their prevention / edited by Bettina Menne and Virginia Murray. Copenhagen : WHO Regional Office for Europe.

[https://www.euro.who.int/\\_data/assets/pdf\\_file/0020/189020/e96853.pdf](https://www.euro.who.int/_data/assets/pdf_file/0020/189020/e96853.pdf)

World Meteorological Organization, 2007: Technical Regulations. Vol. III: Hydrology. (WMO-No. 49), Geneva. [https://library.wmo.int/doc\\_num.php?explnum\\_id=4564](https://library.wmo.int/doc_num.php?explnum_id=4564)

Wu, C. L., Chau, K. W., & Li, Y. S. (2008). River stage prediction based on a distributed support vector regression. *Journal of Hydrology*, 358(1-2), 96–111. doi:10.1016/j.jhydrol.2008.05.028

Wu, C. L., Chau, K. W., & Li, Y. S. (2009). Predicting monthly streamflow using data-driven models coupled with data-preprocessing techniques. *Water Resources Research*, 45(8). doi:10.1029/2007wr006737

Wu, J., Chen, X.-Y., Zhang, H., Xiong, L.-D., Lei, H., Deng, S.-H. (2019). Hyperparameter Optimization for Machine Learning Models Based on Bayesian Optimization. *Journal of Electronic Science and Technology*, 17(1). ISSN 1674-862X. <https://doi.org/10.11989/JEST.1674-862X.80904120>.

Xie, Y., & Lou, Y. (2019). Hydrological Time Series Prediction by ARIMA-SVR Combined Model based on Wavelet Transform. *Proceedings of the 2019 3rd International Conference on Innovation in Artificial Intelligence - ICIAI 2019*. doi:10.1145/3319921.3319959

Yu, L., & Liu, H. (2003). Feature selection for high-dimensional data: A fast correlation-based filter solution. In *Proceedings of the Twentieth International Conference on Machine Learning (ICML-03)*.

Zeng, Z., Zhang, H., Zhang, R., & Zhang, Y. (2014). A Hybrid Feature Selection Method Based on Rough Conditional Mutual Information and Naive Bayesian Classifier. *ISRN Applied Mathematics*, 2014, 1–11. doi:10.1155/2014/382738

Zhang, G., & Lu, Y. (2012). Bias-corrected random forests in regression. *Journal of Applied Statistics*, 39(1), 151–160. doi:10.1080/02664763.2011.578621

Zhou, T., Wang, F., & Yang, Z. (2017). Comparative Analysis of ANN and SVM Models Combined with Wavelet Preprocess for Groundwater Depth Prediction. *Water*, 9(10), 781. doi:10.3390/w9100781