M.Sc. Thesis
Master of Science in Engineering

DTU Compute
Department of Applied Mathematics and Computer Science

Automated ARIMA-Type Model Selection

Comparisons of Data-Driven Flow Forecast Models

Ari Jóhannesson (s181320)

Kongens Lyngby 2020

DTU Compute
Department of Applied Mathematics and Computer Science
Technical University of Denmark

Matematiktorvet
Building 303B
2800 Kongens Lyngby, Denmark
Phone +45 4525 3031
compute@compute.dtu.dk
www.compute.dtu.dk

Summary

Hydrological systems benefit both economically and environmentally from high-quality models that can predict and prepare the system for an increase in runoff. Due to data-availability and advancement in machine learning, data-driven models, which are a simplification of the conceptual models, have been gaining attention in recent years due to their speed and flexibility.

ARIMA is a statistical model that can be used for generating forecasts for hydrological systems. The order of the model is traditionally identified by manual inspection of the time series, but this manual inspection can both be tedious and time-consuming.

In this thesis, an automated model selection of ARIMA type models, using meta-optimization on two catchments in Copenhagen, is presented.
For the model selection, the following things are investigated:

- How automated model selection can be carried out efficiently
- Comparison of models with/without precipitation as an input.
- How different optimization algorithms perform in the coefficient estimation (i.e. local/global-search).
- Comparision of models that are fitted using a single-, and multi-step forecasts.

Runoff forecasts for 30, 60, and 90-minute forecasting horizons will be considered. Two different error measures will be used for model selection, Persistence Index skill-score (PI), and accuracy in predicting a simplified version of ATS activation

Preface

This thesis was prepared at the Technical University of Denmark at the Department of Environmental Engineering. The work was done to fulfill the final requirement of obtaining an M.Sc. degree in Mathematical Modelling and Computation.

Kongens Lyngby, August 3, 2020

Ari Jóhannesson (s181320)

Acknowledgements

First and foremost, I would like to thank my supervisor Associate Professor Roland Löwe (DTU Environment) for weekly discussions and strong support throughout this thesis. Also, special thanks to my co-supervisor Associate Professor Luca Vezzaro (DTU Environment) for his guidance, especially regarding the performance measure for ATS activations.

Nomenclature

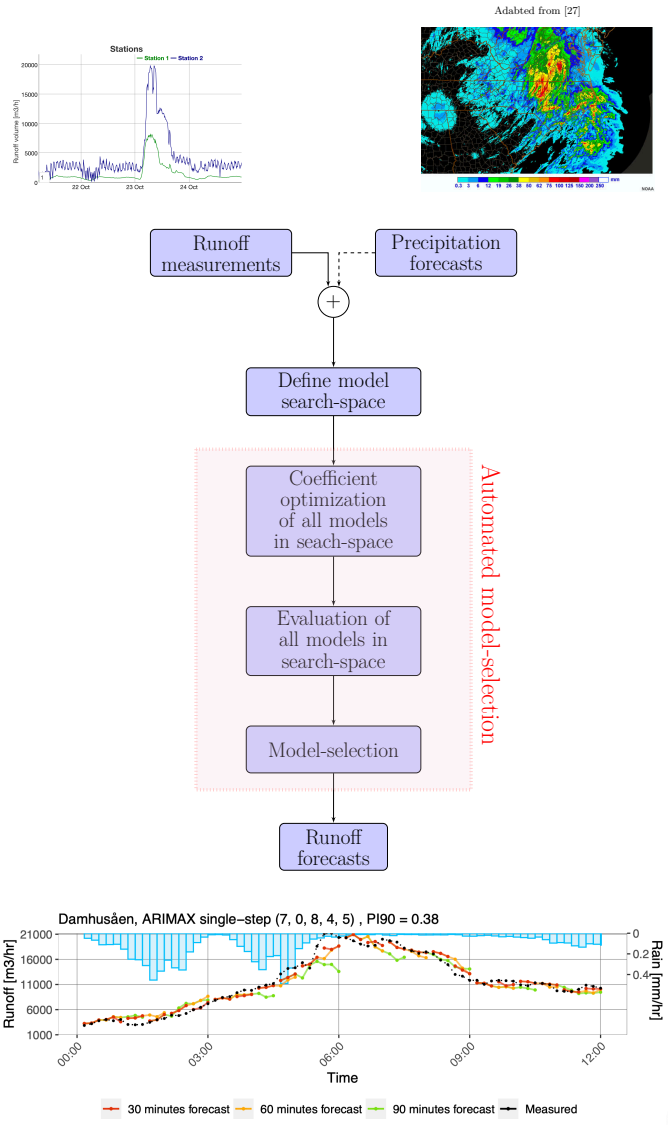
Abbreviations

ANN	Artificial neural network
AR	Auto-regressive
ARIMA	Auto-regressive intergrated moving-average
ARIMAX	Auto-regressive intergrated moving-average with external regressor
ATS	Aeroated tank settling
CCF	Cross-correlation function
CSO	Combined sewage overflow
MA	Moving-average
DDS	Dynamically dimensioned search
PI	Persistence-index
SS	Skill-score
WWTP	Waste water treatment plant

List of Symbols

y_t	observation at time point t
$\hat{y}_{t+k}^{(k)}$	k step-ahead prediction taken from time step t
p	Auto-regressive term
d	Intergration term
q	Moving-average term
r	$\max(p, q)$
θ	Moving-average coefficients
ϕ	Auto-regressive coefficients
ϵ	Residual

Graphical Abstract



Contents

CHAPTER 1

Introduction

1.1 Background

Below the surface of urban infrastructure lies a layer of underground pipes and pumps that play a critical role in sanitation and water supplying. The use of underground pipes for sanitation and water transport dates back thousands of years to civilizations such as the Greeks (Cretes Minoans), the Romans, and the Mayans [3]. Since then, these systems' infrastructures have changed a lot and are becoming yet more sophisticated.

Nowadays, systems have become centralized around wastewater treatment plants (WWTP), where wastewater is treated in various ways to either redistribute it to households or dispose of the treated wastewater into the environment with minimum impact. The wastewater that runs through the WWTP is often a combination of sewage and surface water, referred to as combined sewer (CS). In storms and heavy rain events, CS can lead to wastewater exceeding the pipes' capacity, resulting in combined sewer overflow (CSO), an undesirable discharge of untreated wastewater into the environment. Preventing CSO and optimizing flow in pipes is thus an essential aspect of urban hydrological systems.
One solution is to increase the capacity of pipes and pumps, an enormously expensive and complicated solution. Another much more economical and cleaver solution is to utilize mathematical models to optimize and dynamically control flow in pipes and pumps throughout the system.

The modeling of hydrological systems in urban cities has been gaining attention in recent years, likely due to the ever-increasing data availability, more computation power, and better quality models [33]. Controlling CS dynamically and preventing CSO is often done by using precipitation and flow measurements in the system to forecast forthcoming runoff volume that will drain to the plant. With a quality forecast, WWTP has time to manage itself and prepare for an increase in runoff volume dynamically. There are various ways in which WWTP can prepare itself for an increase in flow rate, for instance, with activation of aeration tank settling. Additionally, [22] have proposed dynamic overflow risk assessment (DORA), a global optimization strategy for minimizing the risk of CSO by utilizing the information on the water volume presently stored in the drainage network along with the forecasts to manage

the system dynamically.

Data-driven black-box models are becoming ever more popular in forecasting hydrological systems. One reason for this is that the classical white-box hydrodynamic models that rely on solving differential equations based on the system's physical structure are slow and tedious in calibration [20] [12]. These data-driven models reduce the model complexity and only rely on the underlying statistical structure and patterns of the data with little (if any) physical interpretation of the system for generating a forecast. Conceptual models such as artificial neural networks (ANNs) and auto-regressive integrated moving-average (ARIMA) type models have been used extensively in hydrology [24] [10] [6] [25]. The tuning of these models can be quite tricky but once calibrated, they can produce computationally efficient and high-quality forecasts. ANNs are machine learning models that give limited suggestions on the system's underlying behavior, often desired information by hydrologists. ARIMA type models are, on the other hand, capable of providing information on the system's characteristics.

In this work, an approach will be taken towards automated model selection using ARIMA type models for two catchments in Copenhagen. The models will be specifically adapted to forecast during extreme events that pose a risk of CSO. The main innovations of this work are development of modules that can be used to fit ARIMA type models to single/multi-step forecasts. Meta-optimization will be used for ARIMA model identification and different objective function criteria (i.e. single/multi-step forecasts) as well as different coefficient optimization algorithms will be compared. Two evaluation metrics are proposed: PI skill-score, and accuracy in predicting a simplified version of ATS activation.

1.2 Research Questions

This work aims at answering the following questions:

R.Q.1 Can competent ARIMA type models be selected in an automated and efficient manner?

R.Q.2 How should the parameter search-space be constrained such that parameter selection can be performed in a computationally efficient manner, while still selecting parameters that adequately capture the complex behavior of the system?

R.Q.3 Do local and global-searches in the coefficient estimation produce significantly different models?

R.Q.4 Do different objective function criteria (i.e., calibrating models to single/multi-step forecasts) generate substantially different models?

R.Q.5 Will proposed error metrics result in analogous models.

R.Q.6 Does precipitation as an external regressor improve forecasting?

R.Q.7 How do ARIMA models compare to the current models in use?

The model framework is shown in Figure 1.1. The figure also shows where research questions are addressed.

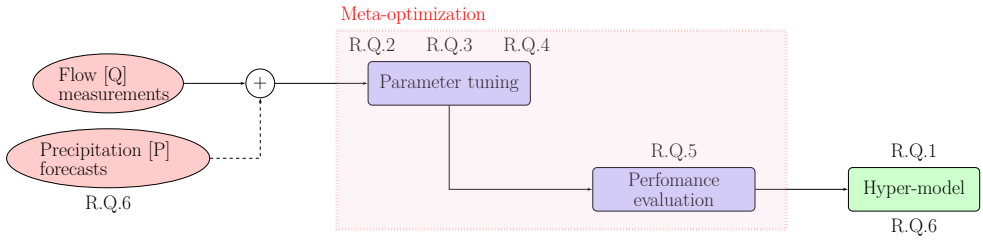


Figure 1.1: Model framework and where research questions are adressed

1.3 Objectives

To investigate already mentioned research questions, flow measurements from two locations in Copenhagen, Denmark, will be used for model fitting and evaluation.

1. Literature review on the calibration process of ARIMA type models and how forecasts are generated.
2. Constructing an index based on a precipitation threshold to denote what data is used in the calibration process. This index helps to fit the model on the rare events that pose CSO risk.
3. Training and validation split.
4. Make functions that generate ARIMA forecasts for given hyper-parameters. These functions support external regressors and are capable of generating single- or multi-step forecasts.
5. Discussion of evaluation metrics, and the consequences of using them.
6. Hyper-parameter search-space defined.
7. Framework desinged to parallelize the model calibration (coefficient estimation) stage in a high-perfomance cluster.
8. All model hyper-parameters in search-space are calibrated with both global- and local-search.

9. Similarly, all model hyper-parameters are fitted to two different objective function criteria i.e., single/multi-step residuals.
10. Evaluation of generated models and their performances.
11. Inference of models and model selection.

CHAPTER 2

Theory

2.1 Current Models in Hydrology

Hydrological models can be applied in either on-line or off-line fashion. The off-line applications focus on longer horizons (years) while the on-line applications focus on a much shorter horizon (minutes or days). This work will focus on on-line hydrological models. The purpose of on-line hydrology modeling is both economical and environmental. The environmental purpose is to prepare the WWTP for an increase in runoff, preventing CSO. The economical aspect of hydrology modeling is that with adequate forecasting models, the hydrological systems cut cost in infrastructural solutions such as enlarging pipes and detention basis [22].

Hydrological models are often classified into white- and black-box models. The white-box models rely on the physical structure of the system. In contrast, the black-box methods are purely data-driven and use statistics without physical interpretation of the system to generate forecasts. In between these models lie the grey-box models, which combine statistical data-driven methods with some physical interpretation.
Hydrodynamic urban drainage models, such as MIKE, are white box-models that use input parameters such as surface water to solve for the pipe flows in the system using differential equations. The differential equations are based on the Saint Venant differential equations and solving for these equations makes the system complex, computationally heavy, and slow. Additionally, the calibration of hydrodynamic models is complex and has to be done manually [20] [12].
Black-box models simplify the system vastly but are capable of producing high-quality forecasts much faster. In hydrology, the black-box models date back to the 1930s, where Sherman introduced the theory of the hydrograph [12], but these models have been shown to have good predictive qualities, even outperforming some of the conceptual white-box models. In [14], simple regression models, stochastic grey-box models, and complex hydrological and full dynamic wave models were compared. The stochastic grey-box model and simple regression proved outperformed the full dynamical wave model.

Nowadays, deep learning and neural networks are among the more popular methods for hydrological forecasting. Still, methods such as K-nearest neighbor (KNN) and support vector machines (SVM) have also been used [37]. One of the advantages of

these methods is that little information on the hydrological system's complex characteristics is necessary to achieve acceptable forecasts. Still, these models rely entirely on input data, statistics, and mathematics forecasting. In [37], different machine learning techniques were used to forecast monthly flow in two different basins in Iran. and concluded that KNN and Radial Basis Neural Network (RBFNN) provided the most accurate forecasts. Other studies such as [6] and [25] have considered ANN for forecasting CSO and for optimizing inner catchment wastewater transfer.

Another type of model that has been used extensively in forecasting is the auto-regressive integrated moving average (ARIMA) model, a statistical model that can produce excellent forecasts. However, these models do not come ready out of the box, and a careful examination of the time series has to be done to estimate the model parameters. In [24], the inflow of the Dez Dam reservoir in Iran was forecasted with ARMA and ARIMA models and compared with static and dynamic neural networks. In the study, 42 years were used for training, and five years for an evaluation. All tested models were successfully used for forecasting reservoir inflow. ARIMA performed better than ARMA models (due to the differencing parameter), and the dynamic neural network performed better than the static one, outperforming the ARMA/ARIMA models as well.

Literature seems to clash regarding the superiority of ANNs and ARIMA type models. Both models can generate good forecasts, but the difference in their predictive capabilities seem to depend on data, specifically the nonlinearity [7]. However, something that the ANNs lack is that they do not give any information on the system's characteristic behavior, something that is often desired by hydrologists, and something that ARIMA type models are well capable of delivering.

2.2 The Box and Jenkins Modeling Approach

The ARIMA models were introduced in the 1950s and then revisited in 1970 by Box and Jenkins. Box and Jenkins published Time Series Analysis: Forecasting and Control [11], where they developed a three-stage iterative analysis for time series identification, estimation, and verification, sometimes referred to as Box and Jenkins approach [15]. The publication had an immense impact on hydrological and wastewater modeling, and other fields involved in time series analysis and forecasting [10]. Since the publication, the methodology has been used to identify model for hydrology [19] [10] as well as in other areas such as tourism [9].

The outline of the Box and Jenkins modeling approach are depicted in Figure 2.1. The procedure starts with inspecting the auto-correlation (ACF) and partial-auto-correlation functions (PACF) visually and checking for stationarity. More explicit confirmation of stationarity can be done with statistical tests such as the Augmented Dickey-Fuller test (ADF). If the time series is not stationarity, differencing it could

make its stationarity. For linear trends, single differencing is generally sufficient to ensure stationarity, but occasionally, further differencing has to be done to ensure stationarity (for instance, quadratic trend). After stationarity has been ensured, a model is selected (often by observing ACF and PACF visually for choosing appropriate lags to include) and its coefficients estimated by minimizing the residuals. As residuals can be seen as single-step forecasts, the Box and Jenkins modeling approach is specifically calibrated to single-step predictions. When the model has been fitted, the residuals are inspected for auto-correlation. If the residuals turn out to look like white noise (no auto-correlation), the model is adequate, but if there exists some auto-correlation within the residuals, a different model is selected. These three steps are the main stages of the Box and Jenkins modeling approach, and if the model passes each state, then it can be used for forecasting. However, to restrict model complexity, the last node in the flowchart is often included.

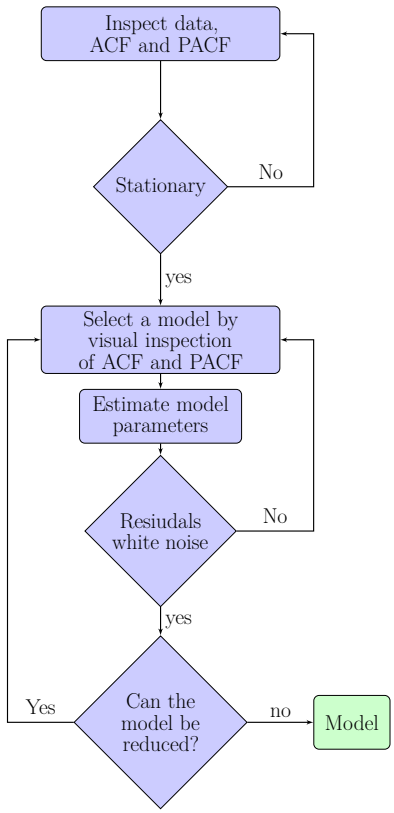


Figure 2.1: The Box and Jenkins modelling approach generally consists of 3 stages, identification, estimation, and verification.

2.3 Auto-Regressive Integrated Moving-Average

In this section, the auto-regressive integrated moving average (ARIMA) model will be discussed and explained how it is estimated, and how forecasts are generated. First, a univariate ARIMA model will be discussed. Once that has been demystified, it will be moved on to discussing the multivariate version of the ARIMA models, often referred to as auto-regressive integrated moving-average with explanatory variable or simply ARIMAX. Throughout this section, author relies on equations and definitions from [30] and [17].

2.3.1 Components of ARIMA Model

The ARIMA models can be broken down into three elements: auto-regressive (AR), integrated (I), and moving-average (MA). AR/MA components are used to structure the model, while the integrated component (I) plays a role in making the time series stationary. Definitions of stationarity differ but in this work, stationarity will be defined as a characteristic of time series where the probabilistic behaviour of a time series at times $y_1, y_2, y_3 \dots y_n$ is the same as for $y_{1+h}, y_{2+h}, y_{3+h} \dots y_{n+h}$.
Following this, time series is stationary if :

- μ is constant and does not depend on t
- σ is constant and does not depend on t

Due to its I component, ARIMA models are especially useful where data shows non-stationarity as it can be eliminated with the integrated component. For a linear trend, differencing once is normally enough to get the time series to a stationary form, but if the time series has a time-varying trend, differencing it more than once is sometimes needed. Nevertheless, it is good to keep in mind that going beyond second-order differencing is rarely done [28].

ARIMA models are denoted ARIMA(p, d, q), where p, d, and q are non-negative integer parameters, used set each component of the model. p and q, for instance, denote the number of lags the auto-regressive and moving-average component take into consideration. The integration component d denotes the degree of integration. By combining these components, a powerful model to forecast future events can be achieved.

Autoregressive (AR)

The AR component consists of a classic linear regression of degree p. In essence, predictions are generated by multiplying past observations by regression coefficients and adding the random white noise error term at time t. The AR model is described by Equation 2.1

$$y_{t+1} = \mu + \phi_1 y_t + \phi_2 y_{t-1} + \cdots + \phi_p y_{t-p-1} + \epsilon_{t+1}$$
$$y_{t+1} = \mu + \sum_{i=1}^{p} \phi_i y_{t-i-1} + \epsilon_{t+1} \tag{2.1}$$

Where μ is mean, ϕ is a vector holding the regression coefficients $\phi_1, \phi_2, \ldots, \phi_p$, and ϵ is the error term. Because ϵ is iid with zero mean and constant variance ($\epsilon_t \sim (0, \sigma_w^2)$), forecasted errors are always 0. A convenient notation to describe time series models (especially when differencing will be introduced) is the notation using backshift operators. The backshift operator is defined as $B^k x_t = x_{t-k}$ where B is the degree of backshift (lag). The auto-regressive operator in Equation 2.2 can be used to simplify the AR model to Equation 2.3.

$$\phi(B) = 1 - \sum_{i=1}^{p} \phi_i B^i \tag{2.2}$$
$$\phi(B) x_t = w_t \tag{2.3}$$

Intergrated (I)

The second component forces the time series to stationarity. This is done by assigning value as the difference between itself and its past value. Sometimes, doing this once is not enough, and a higher degree of differentiation performed. A linear trend can be removed by performing differentiation once, but if there is a quadratic trend, a differentiation of second degree must be done. Equation 2.4 shows first-order differentiation.

$$\nabla y_t = y_t - y_{t-1} \tag{2.4}$$

Figure 2.2 shows how a non-stationary random walk process can be made stationary. In the left figure, a non-stationarity can be observed due to time-varying mean. However, when differentiation of degree one is taken, mean ceases to be time-dependant, and stationarity is achieved.

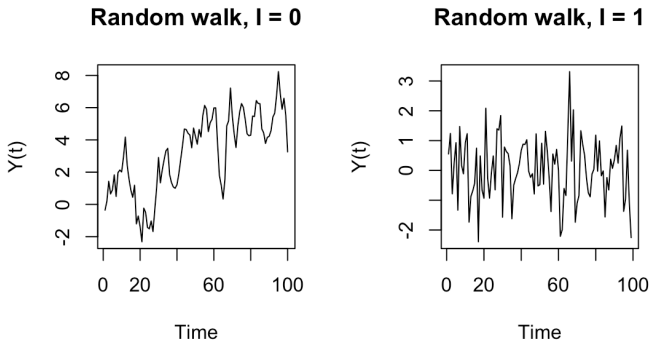


Figure 2.2: Non-stationarity of Random walk process can be removed by taking the difference between current and prior values, in this case y_t and t_{t-1}

.

The backshift operator is especially handy when dealing with differentiation but beacause $By_t = y_{t-1}$, the differentiation of the time series observations $y_t - y_{t-1}$ can be simplified to $\nabla^d y_t$ where $\nabla = (1 - B)$ is the differentiation operator. Thus, to perform a differentiation on a AR model (Equation 2.3) using the backshift operator would be done as in Equation 2.5.

$$\phi(B)\nabla^d y_t = w_t \tag{2.5}$$

Where $\nabla = (1 - B)$, d is the degree of differentiation, $\phi(B)$ the auto-regressive operator, y_t is a time series observation and w_t the error term.

Moving average (MA)

The last component of the ARIMA type models is the moving average component. The MA component uses a linear combination of past forecasting errors multiplied with coefficients for generating forecasts. An MA model of degree q is described in Equation 2.6.

$$\begin{aligned} y_{t+1} &= \theta_1\epsilon_t + \theta_2\epsilon_{t-1} + \cdots + \theta_q\epsilon_{t-q-1} \\ y_{t+1} &= \sum_{i=1}^{q} \theta_i\epsilon_{t-i-1} \end{aligned} \tag{2.6}$$

Where θ is a vector conatining the coefficients $\theta_1, \theta_2, \ldots, \theta_q$ of the model, and ϵ_t forecast errors at time step t. With the backshift operator (Eqations 2.7), an MA(q) model can be represented as in Equation 2.8

$$\theta(B) = 1 + \sum_{i=1}^{q} \theta_i B^i \tag{2.7}$$

$$x_t = \theta(B) w_t \tag{2.8}$$

2.3.2 Assembly of ARIMA Model Components

Having discussed the three components of ARIMA models leads us to a section where these components are assembled. The AR and MA components are simply added together to form an ARMA model or non-differentiated ARIMA model (see Equation 2.9).

$$y_{t+1} = \mu + \sum_{i=1}^{p} \phi_i y_{t-i-1} + \epsilon_{t+1} + \sum_{i=1}^{q} \theta_i \epsilon_{t-i-1} \tag{2.9}$$

With differentiation, the equation for ARIMA models can get messy. The backshift operator was explicitly introduced for that reason, but it simplifies the ARIMA model significantly, especially when differencing is performed. Equation 2.10 shows the equation for the ARIMA model using the backshift operator.

$$\phi(B)\nabla^d y = \alpha + \theta(B)\epsilon \tag{2.10}$$

Where $\phi(B) = 1 - \phi_1 B - \ldots - \phi_p B^p$ is the auto-regressive operator, $\theta(B) = 1 + \theta_1 B + \ldots + \theta_q B^q$ is the moving-average operator, $\nabla = 1 - B$ is the differentiation operator and $\alpha = \mu\,(1 - \phi_1 - \ldots - \phi_p)$.

2.3.3 External Regressors

ARIMA models only work on univariate time series, but with a simple extension, a multivariate model can be attained. This multivariate form of ARIMA is referred to as ARIMAX or ARIMA with an explanatory variable. Adding an external regressor to the ARIMA model is essentially adding a single or more regression coefficients multiplied by lagged time points of an explanatory variable (Equation 2.11). Notice that the exploratory variable is denoted x, and is different from the time series y.
When it is decided to add an external regressor to the ARIMA model, a decision has to be made on how many external regressors are wanted and the lag of these external regressors. For instance, if it is desired to add precipitation as an external

regressor on a hydrological ARIMA type model, the regressors need to be lagged as the precipitation does not cause an instantaneous increase in runoff (the surface water delays in reaching the WWTP).

$$\hat{y_{t+1}} = \underbrace{\mu + \sum_{i=1}^{p} \phi_i y_{t-i-1} + \epsilon_{t+1} + \sum_{i=1}^{q} \theta_i + \epsilon_{t-i-1}}_{\text{ARIMA}} + \underbrace{\sum_{i=1}^{\rho_n} \lambda_i x_{t-i-\rho_{lag}}}_{\text{External regressor}} \tag{2.11}$$

Where ρ_{lag} is the lag of external regressors and ρ_n the number of desired external regressors. With backshift operators, the ARIMAX model can be expressed as in Equation 2.12.

$$\phi(B)\nabla^d y_t = \rho(B)x_t\theta(B)\epsilon_t \tag{2.12}$$

2.3.4 Stationary Process and Unit Roots

Stationarity was briefly discussed in Section 2.3.1 but a time series was said to be stationary if mean and variance are constant and do not depend on time. Now it is turned to the stationary processes of ARIMA type models.
When optimizing the coefficients of a given ARIMA type model, a thing to keep in mind is the possibility of explosive behavior in the AR terms and non-invertibility in the MA terms. For an AR model from Equation 2.1 with order $p = 1$ and zero mean, Equation 2.13 can be derived, where the AR(1) model is expressed as MA(k) model.

$$y_{t+1} = \phi^k x_{t-k} + \sum_{j=0}^{k-1} \phi^j w_{t-j} \tag{2.13}$$

For k iterations backward. If $k \to \infty$, it can be observed that when $\phi_1 < 1$, the process is stationary due to a constant mean and constant variance. However, when $\phi_1 > 1$, the mean will explode, resulting in non-stationarity. If $\phi_1 = 1$, there is a presence of a unit root and the variance gets bigger every time step, resulting in non-stationarity.

Another thing that is normally ensured, is the invertibility of the MA term, a characteristic where a time series can be expressed as a function of past observations. To express ϵ_t with a linear function of past observation consider the transformation of MA(1) model to Equation 2.14. The MA(1) model can be expressed as an AR(∞) model only if $\theta < 1$ where each term will progressively get smaller (infinite geometric sum).

$$y_{y+1} = \epsilon_t - \theta y_t - \theta^2 y_{t-1} - \dots \tag{2.14}$$

Hence, it can be seen that the explosive behavior and the non-invertibility are caused by the presence of a unit roots, which cause the model to behave in a non-stationary process (if no transformation is done). For $p > 1$ and $q > 1$ the same applies but in short, explosive behaviors in AR terms result if inverse-roots of the characteristic equation lie outside the unit circle. In contrast, MA non-invertibility is caused if the inverse roots of the characteristic equation lie inside the unit-circle.

2.3.5 Coefficient Estimation of ARIMA Models

The estimation (sometimes referred to as calibration) of the ARIMA coefficients relies on minimizing/maximizing an objective function with numerical optimization algorithms. First, the objective functions that are normally used will be discussed. Then, it will be moved on to the discussion on how the objective function can be minimized/maximized with numerical algorithms.

2.3.5.1 Maximum likelihood estimation

Maximum-likelihood estimation strives to maximize the likelihood function, a function that is defined as the joint probability density of obtaining time series by varying parameters ϕ and θ. For observations $y_1, y_2, \dots, y_n$ and parameters θ and ϕ, the likelihood function is defined as the joint probability density of the time series based on μ, σ^2, ϕ, and θ (Equation 2.15). The joint probability density can be simplified to Equation 2.16.

$$L(\phi) = f(y_1, y_2, \dots, y_n|\phi) \tag{2.15}$$
$$= f(y_1|\phi) \times f(y_2|\phi) \times \dots \times f(y_n|\phi) \tag{2.16}$$

If the observations $y_1, y_2, \dots, y_n$ are iid$\sim \mathcal{N}\left(\mu, \sigma^2\right)$ the probability density function is Gaussian and the likelihood function becomes a product of of it (Equation 2.18).

$$L(y_1, \dots, y_n|\mu, \sigma) = \prod_i^n \frac{1}{\sigma\sqrt{2\pi}} \exp(-\frac{1}{2}[\frac{y_i - \mu}{\sigma}]^2) \tag{2.17}$$
$$\tag{2.18}$$

Because the natural logarithm is monotonically increasing function (maximum value for the joint probability function occurs at the same point as the log of the probability

function), it can be used to simplify the likelihood function to the log-likelihood function (denoted ℓ) in Equation 2.19.

$$\ell(\mu, \sigma) = -n \log \sigma - \frac{n}{2} \log 2\pi - \frac{1}{2\sigma^2} \sum_{i=1}^{n} (y_i - \mu)^2 \tag{2.19}$$

Maximizing the log-likelihood function exactly can be problematic because the function can not be differentiated concerning the parameters. Thus optimization methods are typically used to find solutions numerically.

2.3.5.2 Least-square estimation

The least-square estimation is a method that estimates the coefficients by minimizing the sum of squares. The least-square estimation is very similar to the maximum-likelihood estimation but rather than relying on likelihood function, the method estimates the coefficients numerically by minimizing Equation 2.20 where y are observations and $\hat{y}$ predictions.

$$S(\phi, \theta) = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \tag{2.20}$$

Least-square estimation aims to minimize the square of the residuals while the maximum-likelihood estimation seeks to maximize the time series's probability.

2.3.5.3 Optimization

The calibration of the ARIMA type models is done with numerical optimization algorithms that vary the coefficients to minimize/maximize an objective function. Objective functions could be, for instance, maximum likelihood (which would be maximized) and sum-of-squares (which would be minimized) discussed in the previous section.

In optimization problems there exist two types of optima [18]:

- Local optima: A point x is a local minimizer if there is a neighborhood $\mathcal{N}$ of x^* such that $f(x^*) \leq f(x)$ for all $x \in \mathcal{N}$
- Global optima: A point x^* is a global minimizer if $f(x^*) \leq f(x)$ for all x

An example of a local optimizer is the Nelder-Mead optimizer. The Nelder-Mead approach was introduced in 1965 by Nelder and Meid. The method uses a simplex with $n + 1$ vertices in $\mathbb{R}^n$ and rejects the vertex with the largest function value and updates the corresponding vertex to a point with better value. A new set of simplexes are then iterated until the minimum point is found. If a better point is not found, the

vertex with the best function value is only kept while the other vertices are shrunk towards that value [18].

The problem with the local-searches is that they may not come close to the global optimum, especially for complex models in hydrology. For problems with multiple local minima, global optimization often achieves better solutions for less computational resources. Dynamically dimensioned search (DDS) is a global optimization algorithm specifically designed for solving computationally expensive optimization problems with many parameters. The algorithm avoids poor local optima and finds a single-solution with a simple stochastic heuristic global search. It aims at finding a globally good solution within the user-specified maximum function evaluations. The algorithm is designed to initially search globally and later on narrow the search down from a global to local search. [4]

2.3.6 Forecasting

2.3.6.1 Single-Step Forecasting

A single-step point prediction can be achieved with Equation 2.9 or 2.11 for all time points $t \in [r+1, n]$ where $r = \max(p, q)$, but since the model depends on previous observations, time points before r cannot be predicted. Single-step point predictions can be visualized in Figure 2.3 where ARIMA(2,0,2) model is used to perform single-step forecast $\hat{y}_t^{(1)}$ for all feasible time points $t \in [3, n]$. Equation 2.21 shows more explicit representation of single-step forecast but forecasted errors ϵ_{t+1} are predicted to be zero as $\epsilon_t \sim (0, \sigma_w^2)$.

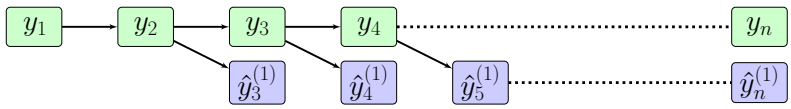


Figure 2.3: One step point predictions use observations (green) to predict point forecast one step ahead.

$$\begin{aligned} \hat{y}_{t+1} &= \mu + \phi_1 y_t + \phi_2 y_{t-1} + \epsilon_{t+1} + \theta_1 \epsilon_t + \theta_2 \epsilon_{t-1} \\ \hat{y}_{t+1} &= \mu + \phi_1 y_t + \phi_2 y_{t-1} + 0 + \theta_1 \epsilon_t + \theta_2 \epsilon_{t-1} \end{aligned} \tag{2.21}$$

2.3.6.2 Multi-step Forecasting

Multi-step forecasts are a series of predictions that are generated for more than one step into the future. Before discussing multi-step forecasts further, it is important to

address the difference between multi-step point-predictions and multi-step forecasts. Consider following defninitions:

- k-step point-prediction:
  A single point-predictions $\hat{y}_{t+k}^{(k)}$, is taken k steps away from obeservation y_t.
- k-step forecast:
  A series of point-predictions $\hat{y}_{t+1}^{(1)}, \hat{y}_{t+2}^{(2)}, \ldots, \hat{y}_{t+k}^{(k)}$, generated at the last observation y_t.

For the single-step forecasting, the single-step point-predictions and single-step forecasts are the same.

The ARIMA(2,0,2) model is revisited in Figure 2.4 where the difference between point-predictions and forecasts can be seen. Single-step point-predictions $\hat{y}_3^{(1)}, \hat{y}_4^{(1)}, \ldots, \hat{y}_n^{(1)}$ are used to form two-step point-predictions $\hat{y}_4^{(2)}, \hat{y}_5^{(2)}, \ldots, \hat{y}_n^{(2)}$. To generate forecasts, it is moved down diagonally from the observations.

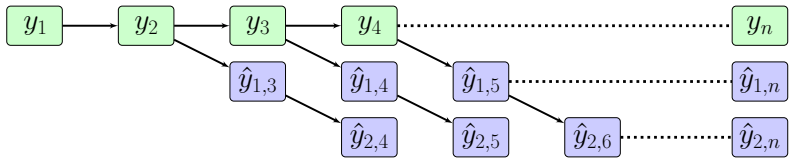


Figure 2.4: Multi-step point-prediction are generated by recursively predicting one-step-ahead. Forecasts are achieved by fusing point predictions (going down diagonally).

To form a two-step point prediction (Equation 2.22), the single-step prediction is treated as a lag-1 observation. Additionally, previously forecasted errors set to zero (ϵ_{t+1}) must remain zero going forward.

$$\hat{y}_{t+2|t+1}^{(2)} = \mu + \phi_1 \hat{y}_{t+1} + \phi_2 y_t + 0 + 0 + \theta_2 \epsilon_t \tag{2.22}$$

Three-step point predictions follow the same procedure. Previously predicted point forecasts are used as observations, and previously predicted errors stay zero (Equation 2.23). Generating multi-step point predictions further into the future is done the same way.

$$\begin{aligned} \hat{y}_{t+3|t+2 \text{ and } t+1}^{(3)} &= \mu + \phi_1 \hat{y}_{2,t+2} + \phi_2 \hat{y}_{1,t+1} + \epsilon_{t+3} + 0 + 0 \\ &= \mu + \phi_1 \hat{y}_{2,t+2} + \phi_2 \hat{y}_{1,t+1} + 0 + 0 + 0 \end{aligned} \tag{2.23}$$

2.3.7 Automated Model Identification

As mentioned, Box-and-Jenkins methodology is one of the major approaches for model identification of ARIMA type models. However, the three-step procedure relies heavily on the expertise, and sometimes, the experts fail to properly identify a model but [36] showed that researchers were only able to correctly identify 28% of computer-generated time series. Another drawback of the Box and Jenkins approach is that the model identification is time-consuming but or real-world applications, where models occasionally need to be re-calibrated or fitted to new catchments, Box-and-Jenkins approach becomes less convenient.

Various methods of automated model identification methods are available. [5] discusses some of those methods and classifies them into three categories, methods that use some penalty function, innovation regression methods, and pattern identification methods. In [35], various of these methods were compared and it was concluded that different methods suit different underlying behavior of the data, but as researchers don't always know the underlying processes of the data, the study only recommends automated model identification methods to be used as a guideline in the model selection process.

The most commonly used approach for automated model identification is to minimize penalized likelihood criteria such as Akaike's Information Criterion (AIC) which compromise between the goodness-of-fit and the model complexity. Using AIC (or other model selection criteria such as BIC) is well suited where data is of low abundance and a hold-out method can not be used. However, when data are abundant, using MSE or other error measures is well suited, as long as the validation is done on the hold-out set [29].

CHAPTER 3

Case study

3.1 Catchments

In this work, the catchment of interest is Damhusåen, one of Copenhagen's larger catchments (reduced area of 1677 ha [2]). The catchment consists of combined sewage which drains to Damhusåen WWTP, located south-east of Copenhagen. Two flow gauges are used for data collection. The first flow gauge is located in Dæmningen, north of Damhusåen WWTP, and serves as wet weather control of Damhusåen WWTP. The flow gauge gives around 30 minutes forewarning if a wet weather control is initiated [2]. The second flow gauge is located in Damhusåen WWTP. Figure 3.1 depicts the Damhusåen catchment in orange color, and flow gauges in Dæmningen and Damhusåen WWTP with green dots.

3.2 Data

This work relies on runoff and precipitation data. This data is collected by Biofos, and is then used by Krüger for the supervision of water treatment plants. The runoff data is collected by the flow gauges of the two previously mentioned locations (Dæmningen and Damhusåen) while the precipitation data is collected by C-band Radar near Stevns. The radar collects nowcast precipitation data by scanning for reflectivities and then uses it to estimate precipitation [34]. The radar data has a temporal resolution of 10 minutes and a spatial resolution of 500m. Further information on the datasets is found in Table 3.1.

Table 3.1: The three data sets used and their range, temporal resolution and units

Data	Range	Temporal resolution	Units
Radar forecast	Jul 2017 - Jan 2020	10 min	mm/h
Dæmningen st.	Aug 2017 - Jan 2020	2 min	m3/h
Damhusåen st.	Apr 2017 - Jan 2020	2 min	m3/h

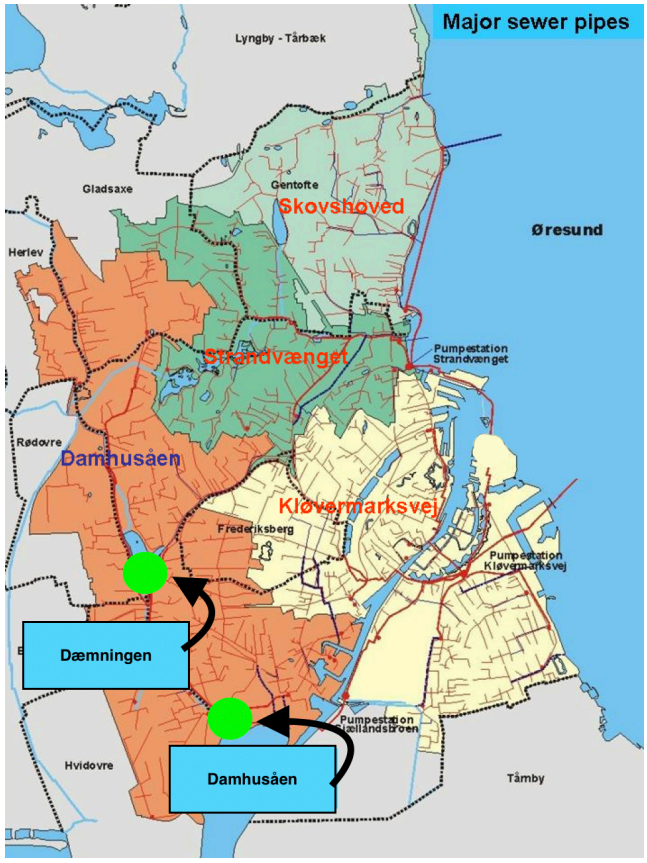


Figure 3.1: Catchment areas of Copenhagen as well as the WWTP. Damhusåen catchment is depicted with red color and is of interest in this thesis. Figure adapted from [2]

3.2.1 Data Splitting

The datasets need to be split up into training and validation sets to get an unbiased validation of the model fit. The ranges of the training- and evaluation-sets are shown in Table 3.2.

Table 3.2: Training and validation set of data

Type	Range
Training	August 16th 2017 - December 31st 2017
Validating	January1st 2018 - December 31st 2018

CHAPTER 4

Exploratory Data Analysis

4.1 Data Visualization

The time series is visualized in Figure 4.1. The green line shows runoff measurements at Dæmningen while the yellow line shows measurements at Damhusåen. Precipitation observations are shown with blue bars sticking from the top of the graph.
Both of the runoffs follow similar patterns, such as increasing and decreasing together, but the runoff data through Damhuåen is much more unstable and of a larger magnitude than its sub-catchment of Dæmningen.
Additionally, some missing data can be seen in the time series. The percentage of missing data for each dataset is shown in Table 4.1.

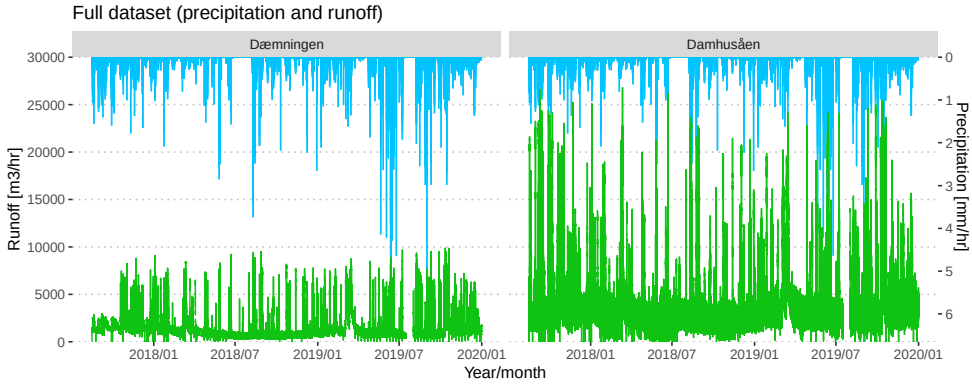


Figure 4.1: Runoff (green line) and precipitation (blue bars) for Dæmningen (left) and Damhuåen (right).

By zooming into the time series patterns can be seen much more clearly. In Figure 4.2, a rain event can be seen occurring a little before an increase in runoff. Damhusåen seems to a lot have more increase in the runoff than Dæmningen, likely due to larger catchment size, and Dæmningen being a sub-catchment of Damnusåen. These inter-

vals of increase in runoff are of interest in this thesis as they possess a risk of CSO. The goal is to forecast these runoff-increases in advance to give the WWTP time to manage and prepare itself.

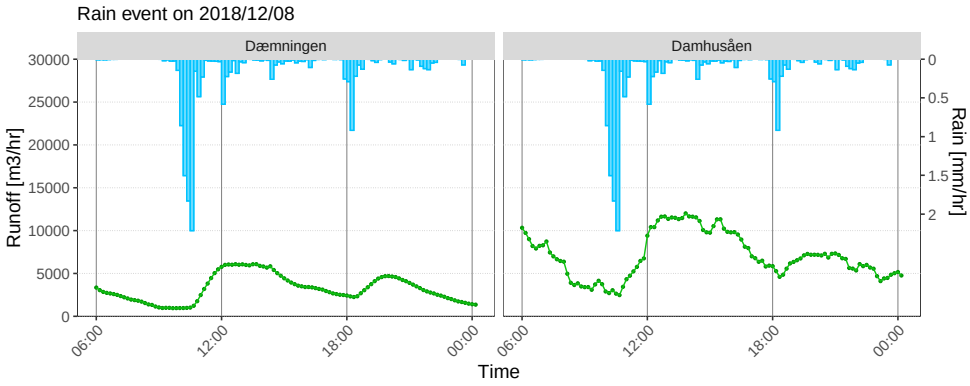


Figure 4.2: An increase in runoff volume through both locations results from increase in precipitation. These are the regions that risk CO and should are investigated in this work.

Table 4.1: Persentage of missing data in datasets

Data	Missing values
Damhusåen	2.9 %
Dæmningen	2.9 %
Precipitation	8.6%

4.2 Cross Correlation

As seen from Figure 4.2, the precipitation (usually) comes just before the increases in runoff volume. For further investigation, the cross-correlation between the datasets is checked. Cross-validation is done by lagging a time series and computing its correlation to another time series.
First, the cross-correlation-factor (CCF) between the stations is computed. As seen in Figure 4.3, the CCF results in distribution with the highest correlation around -3, meaning that an increase in Dæmningen results in an increase in Damhusåen around 30 minutes later (Dæmningen is a sub-catchment upstream of Damhusåen). This is consistent with [2], where it is mentioned that the Dæmningen sub-catchment that gives around 30-minute forewarnings to Damhusåen to initiate wet weather control.

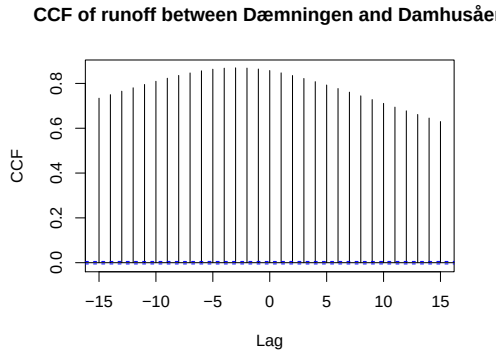


Figure 4.3: Increase in runoff volume through Dæmningen results in increase through Damhusåen around 30 minutes later.

The CCF for Dæmningen is the highest around -12 (corresponding to 120 minutes). For Damhusåen, the CCF is highest around -14 and -13 (corresponding to 130 to 140 minutes). Because Dæmningen is located upstream of Damhusåen, it makes sense that the lag with the highest CCF for Dæmningen comes earlier than for Damhusåen.

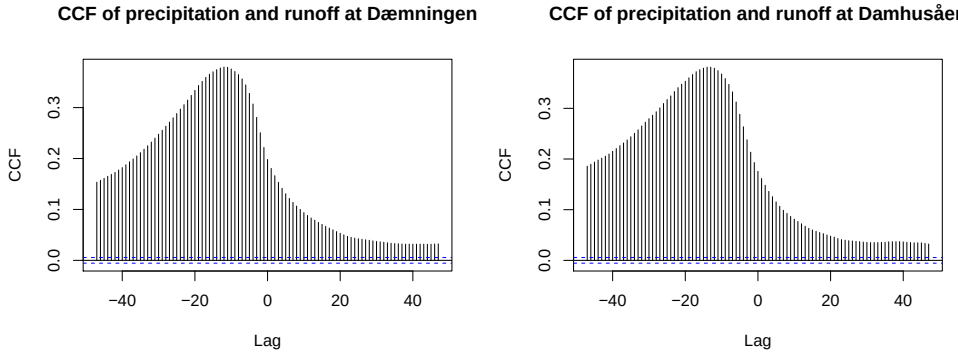


Figure 4.4: CCF between precipitation and the runoff locations.

CHAPTER 5

Methods

This chapter explains how meta-optimization can be used for tuning the model hyper-parameters. The meta-optimization will consist of both models with and without external regressors (ARIMA/ARIMAX) as well as different objective function criteria and coefficient optimization methods. Figure 5.1 depicts the flowchart of the methodology that will be taken as well as in which section they are discussed.
Before further discussion, it is critical that the definitions of hyper-parameters and coefficients are made explicit:

- **Parameters:** Constants that control the model structure i.e. p, d, q, ρ_{lag}, ρ_n. Model hyper-parameters are used to tune the parameters.
- **Coefficients:** Variables that are adjusted to minimize the objective function to generate adequete predictions. These predictions are generated by multiplying the coefficients with observations i.e. ϕ, θ, λ.

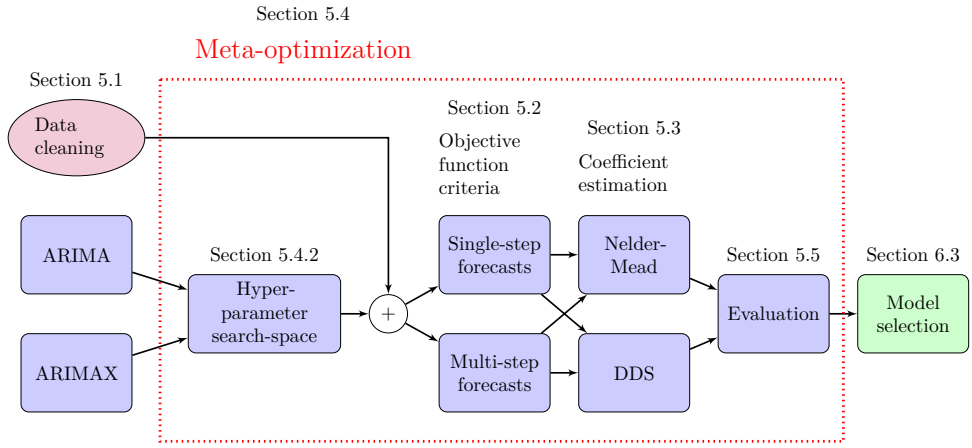


Figure 5.1: Flowchart of methodology

Section 5.1 deals with cleaning and preparing data. To ensure models are trained on events that are considered critical (increase in runoff due to precipitation), a wet weather index is prepared for indicating what data is used in the minimization of the objective function.
Two general model types are considered: basic ARIMA type models using flow measurements, and ARIMAX models using both flow measurements as well as 'perfect' precipitation forecasts. Before considering meta-optimization, different objective function criteria and coefficient estimation are discussed in Sections 5.2 and 5.3 but these different alterations are used to increase the likelihood of capturing a good model. In Section 5.4, the discussion will be turned towards meta-optimization where a search-space of hyper-parameters will be defined, and grid-search used to first estimate hyper-model coefficients and then evaluate the model. Section 5.4.2 discusses the selection of the hyper-parameter search-space and why it is important to limit the search-space. Additionally, section 5.4.3 discusses how the meta-optimization is carried out in a computationally efficient manner. Two different evaluation criteria are presented in Section 5.5. Lastly, the selection of the best performing model for operational purposes will be discussed in Section 6.3

5.1 Data Cleaning and Preparation

In this work, all except one case outliers (one precipitation value is very high and unrealistic), in datasets will are kept.
However, some other data anomalies have to be addressed and fixed before it can be used for training and evaluating models. These anomalies are the following:

- Different ranges of data sets
- Different frequency (temporal resolution) of data sets
- Daylight saving time shifts
- Flatlines in sensor data
- Missing data

All datasets are set to the same range, frequency aggregated to 10 minutes, and data that had been shifted during DST shifted back. All this is discussed in more details in Appendix A.1.

5.1.1 Flatlines and Missing Data

[16] discusses different types of anomalies in hydrological data such as flatlines, missing data, timestamp inconsistency, and how they can be treated. This work deals

with flatlines similarly by applying a rolling window approach of size 5 (corresponding to 50 minutes) through the whole time series to check if all values in the frame are equal. If all values in a frame turn out to be equivalent, a flatline is flagged. By applying this moving window approach, a vector that stores the binary flag for each data point can be constructed. Even though the window size is small, this approach resulted in a nice flatline filter that did not have a lot of jumps where the index is repeatedly turned on/off.

Other undesired common occurrences are missing values. These dubious or erroneous occurrences were analyzed for each dataset. In Figure 5.2, the time series can be seen along with flatline/missing-data flags (red shades). To keep track of where flatlines and missing data are present, a bad-data-index is created which flags these anomalies.

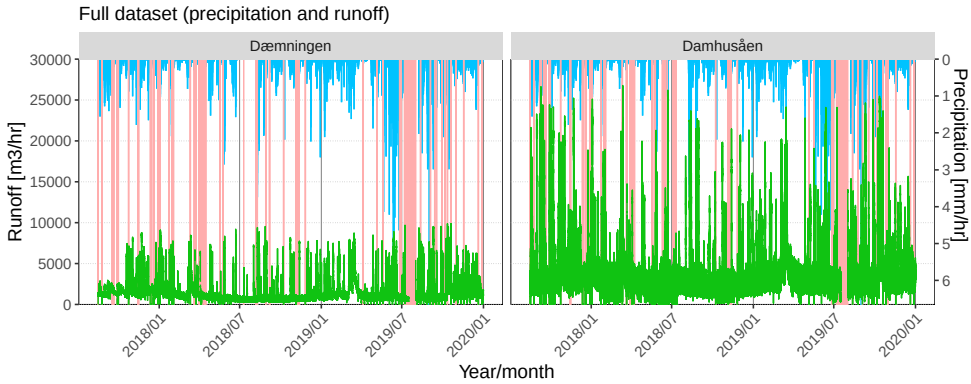


Figure 5.2: Complete time series along with missing data/flatlines flag, represented with red shade

5.1.2 Wet-Weather Index

This project focuses specifically on modeling extreme events, where runoff volume is increased rapidly due to precipitation, risking CSO. Hence, dry periods, which account for a significant amount of the data, are not of interest and must be disregarded during model fitting as they could have a redundant influence on the model fitting. To classify which data periods should be used for fitting and evaluation, a binary flag vector is constructed, which designates which data points are considered 'wet-weather' and consequently used in model fitting and evaluation. A 'wet-weather' threshold was set to 0.665 mm/hr (0.1 after the normalization discussed in Section 5.1.3). All values above the threshold are flagged (rain event) while all bellow it remains unflagged.

The radar data often oscillate around the threshold, creating the illusion of many rain events. To prevent this, and to ensure that rain events are accurately captured, rules from Kruger evaluation document are used [8]. These rules merge temporarily close wet-weather events and are the following:

- Two rain events are merged and considered as one if:
  - The time difference between the start of two rain events is less than 4 hours (Figure 5.3).
  - The time difference between the stop and start of two rain events is less than 2 hours (Figure 5.4).

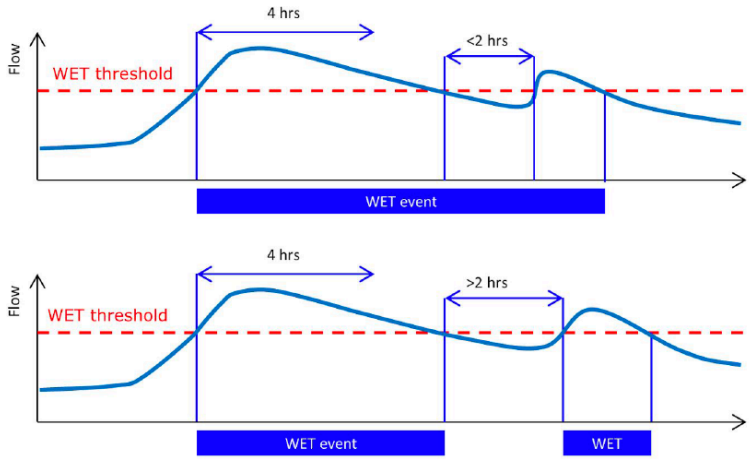


Figure 5.3: If the time difference between the end and the start of two rain events is less than 2 hours, they are considered as one. Figure from [8].

Once rain events have been flagged, and temporarily close rain event merged, an extra 12-hour tail is extended to each rain event. The reason for this is that the system typically needs around 12-20 hours to empty its basins and reach dry weather operation. Figure 5.5 depicts how rain events are defined. Notice that rain events end 12 hours after its precipitation forecast goes bellow threshold.

To ensure that data associated with flatlines or missing data is not included in the training and evaluation process, the bad-data index discussed in Section 5.1.1 is fused with the wet-weather index.

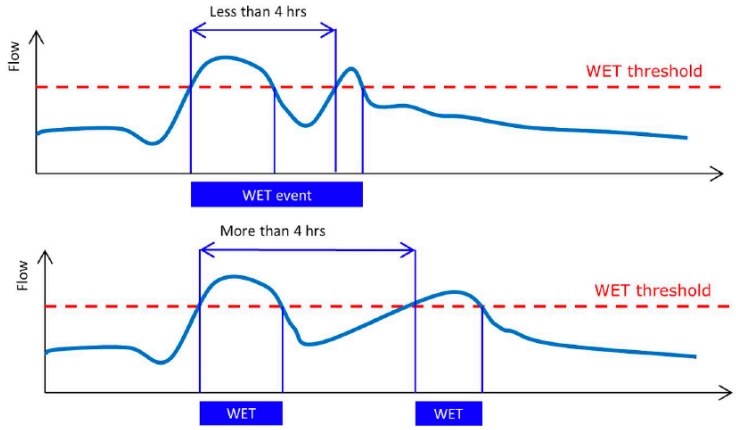


Figure 5.4: If the start time between two rain evetns is less than 4 hours apart, they are considred as one. Figure from [8].

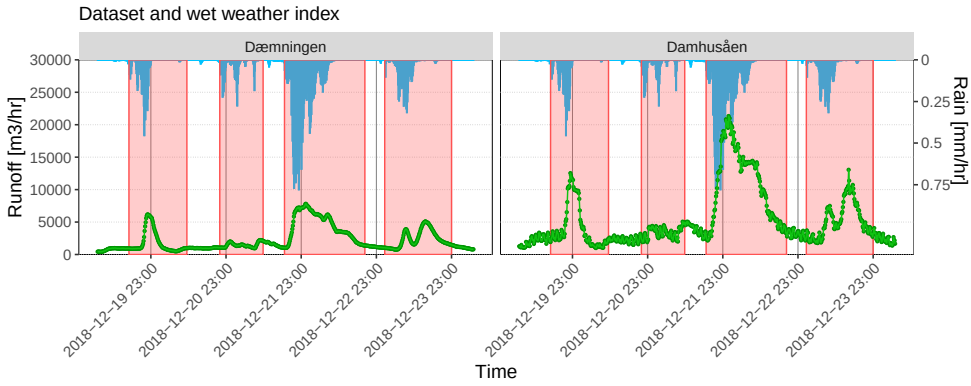


Figure 5.5: Wet weather is flagged accordingly. Rain event starts if precipitation treshold is activated and finishes 12 hours after the precipitation drops bellow treshold. Temporariy close rain events are merged.

5.1.3 Normalization

Normalization is usually done when relations between data on different scales are sought after. In essence, normalization adjusts the data to a common scale and drops units. As the datasets in this work are on a different scale, normalization has to be performed. Different types of normalization exist, but in this thesis, feature scaling is used. Feature scaling is shown in Equation 5.1.

$$x_{\text{new}} = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \tag{5.1}$$

Where x are data points in time series, $x_{\min}$ is the lowest value in time series and $x_{\max}$ is the highest value of time series.

5.2 Objective Function Criteria

The traditional approach of fitting ARIMA type models is to minimize the one-step residuals (see Figure 2.1). However, real-time control of a hydrological system requires forecasts to be generated with various forecast horizons and if model coefficients are tuned for one-step predictions, the model performance may be suboptimal for longer horizons [32].
This section discusses the different objective function criteria and how single-and multi-step predictions can be achieved in a computationally efficient manner.

5.2.1 Single-Step Objective Function Criterion

The single-step objective function criterion involves minimizing the objective function of single-step forecasts/point-predictions which were discussed in Section 2.3.6.1.

The objective function is defined as the sum-of-squares (Equation 2.20) the residuals that result from single-step point-predictions/forecasts for all feasible points $\hat{y}_{r+1}, \hat{y}_{r+2}, \ldots, y_n$ (where $r = \max(p,q)$) over the wet-weather-index (Section 5.1.2).

5.2.2 Multi-Step Objective Function Criterion

Long-term dynamics of hydrological systems cannot always be captured with the single-step objective function criterion. Hence, a multi-step objective function criterion is presented here, where parameters are calibrated to multi-step forecasts.
This approach is seldom used for fitting ARIMA type models and has to be implemented from scratch. Multi-step forecasts are more computationally demanding than single-step forecasts and will be carried out with recursive matrix multiplication.
First, it will be discussed how the multi-step models will be estimated. After that, it is discussed how all this can be achieved in a computationally efficient manner.

5.2.2.1 Performance Estimation

Equation 5.2 shows a scoring criterion for multi-step forecasts which has been considered in [32] as weighted average of residuals over several forecast horizons where

more weight is put on shorter forecasting horizons.

$$SC_i = \frac{1}{\sum_{j=1}^{k}(k-j+1)} \left(\sum_{j=1}^{k}(k-j+1) \cdot SC_{i,j} \right) \tag{5.2}$$

where k is the maximum forecasting horizon and i is the time step. $SC_{i,j}$ are absolute residuals at time step i with forecasting horizon j. In this work, $k = 10$ so multi-step point predictions are performed for $j = 1, 2, \ldots, 10$ and used in the scoring criterium (Equation 5.2). Thereafter, the sum is taken off the scoring criterium, and as before, the objective function will be computed exclusively over wet-weather index.

5.2.2.2 Efficient Implementation

Libraries and function exist for fitting ARIMA type models and generating multi-step forecasts (In R, STATS::ARIMA, STATS::PREDICT). However, these functions are primarily aimed at generating forecasts for a single starting point at the end of the time series. For coefficient estimation using Equation 5.2, N multistep point-predictions need to be generated that start at all N time points in the considered series. To generate these predictions, we could call the standard R functions repeatedly (N times) with an input series of varying length (as shown in Figure 5.6). However, this process is too computationally demanding to be used inside a procedure for coefficient estimation due to the overhead of repeated model setups and data modification for each time step.

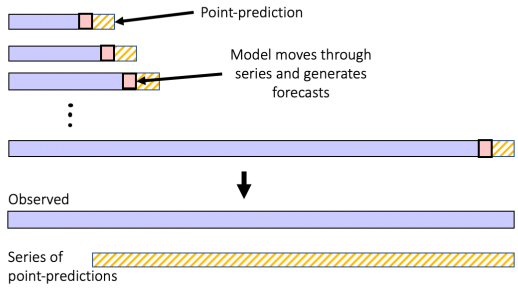


Figure 5.6: Generating multi-step point-predictions with the standard R libraries is too computationaly demanding due to large overlead of repititive model setups.

A more computationally efficient approach is to represent the model in matrix form and to exploit the process properties explained in Section 2.3.6.2. These properties imply that to generate forecasts for horizons $j > 1$, we can replace columns in the input data matrix that are related to AR terms with forecast values generated for shorter forecast horizons, while columns related to MA terms can be replaced by 0 ($\epsilon_{t+k|t} = 0$ for $k > 0$). As the k-step point-forecasts can predict all values the range $t \in [r + k, n]$, where $r = \max(p, q)$, the dimension of the matrix changes with each iteration, loosing its first row. Additionally, as the last point-prediction should be for time step n, the point-predictions generated for a shorter forecast horizon lose their last value $\hat{y}_n^{(k)}$ before being moved in the input data matrix. The forecasted values are denoted $\hat{y}^{(k)}$ for k step forecasts, and $\hat{y}_*^{(k)}$ after the removal of the last value. Same applies for residuals $\epsilon^{(k)}$

To generate multistep predictions for increasing forecast horizons j=1 to k, k iterations are performed where the data matrix is updated in each iteration and then used to generate the forecast for horizon j. As an example, this process is illustrated for forecast horizons of 1, 2 and 3 steps in Equations 5.3, 5.4, and 5.5, where Ψ is the input data matrix, and $\delta = \begin{bmatrix}\mu & \phi & \theta\end{bmatrix}'$ holds the model coefficients.

$$\overbrace{\begin{pmatrix} 1 & y_r & y_{r-1} & \dots & y_{r+1-p} & \epsilon_r & \epsilon_{r-1} & \dots & \epsilon_{r+1-q} \\ 1 & y_{r+1} & y_r & \dots & y_{r+2-p} & \epsilon_{r+1} & \epsilon_r & \dots & \epsilon_{r+2-q} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & y_{n-1} & y_{n-2} & \dots & y_{n-p} & \epsilon_{n-1} & \epsilon_{n-2} & \dots & \epsilon_{n-q} \end{pmatrix}}^{\Psi} \delta = \overbrace{\begin{pmatrix} \hat{y}_{r+1}^{(1)} \\ \hat{y}_{r+2}^{(1)} \\ \vdots \\ \hat{y}_n^{(1)} \end{pmatrix}}^{\hat{y}^{(1)}} \tag{5.3}$$

$$\begin{pmatrix} 1 & \overbrace{\hat{y}_{r+1}^{(1)}}^{\hat{y}_*^{(1)}} & y_r & \dots & y_{r+2-p} & \overbrace{\hat{\epsilon}_{r+1}^{(1)} = 0}^{\hat{\epsilon}_*^{(1)}} & \epsilon_r & \dots & \epsilon_{r+2-q} \\ 1 & \hat{y}_{r+2}^{(1)} & y_{r+1} & \dots & y_{r+3-p} & \hat{\epsilon}_{r+2}^{(1)} = 0 & \epsilon_{r+1} & \dots & \epsilon_{r+3-q} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \hat{y}_{n-1}^{(1)} & y_{n-2} & \dots & y_{n-p} & \hat{\epsilon}_{n-1}^{(1)} = 0 & \epsilon_{n-2} & \dots & \epsilon_{n-q} \end{pmatrix} \delta = \begin{pmatrix} \overbrace{\hat{y}_{r+2}^{(2)}}^{\hat{y}^{(2)}} \\ \hat{y}_{r+3}^{(2)} \\ \vdots \\ \hat{y}_n^{(2)} \end{pmatrix} \tag{5.4}$$

$$\begin{pmatrix} & \overbrace{\hat{y}_*^{(2)}}^{} & \overbrace{\hat{y}_*^{(1)}}^{} & & & \overbrace{\hat{\epsilon}_*^{(2)}}^{} & \overbrace{\hat{\epsilon}_*^{(1)}}^{} & & \\ 1 & \hat{y}_{r+2}^{(2)} & \hat{y}_{r+1}^{(2)} & \dots & y_{r+3-p} & 0 & 0 & \dots & \epsilon_{r+3-q} \\ 1 & \hat{y}_{r+3}^{(2)} & \hat{y}_{r+2}^{(2)} & \dots & y_{r+4-p} & 0 & 0 & \dots & \epsilon_{r+4-q} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \hat{y}_{n-1}^{(2)} & \hat{y}_{n-2}^{(2)} & \dots & y_{n-p} & 0 & 0 & \dots & \epsilon_{n-q} \end{pmatrix} \delta = \begin{pmatrix} \overbrace{\hat{y}_*^{(3)}}^{} \\ \hat{y}_{r+3}^{(2)} \\ \hat{y}_{r+3}^{(2)} \\ \vdots \\ \hat{y}_n^{(2)} \end{pmatrix} \tag{5.5}$$

With each iteration, all point-predictions $\hat{y}^{(k)}$ for forecasting horizon k are stored row-wise in matrix Ω. With Ω, residuals can easily be obtained when computing the scoring criterium (Equation 5.2). Forecasts can also be generated from all observations $y_r, y_{r+1}, \dots, y_{n-1}$. Example in Equation 5.6 shows how the three-step point-predictions are stored (blue line) and how forecast can be obtained from observation y_r.

$$\Omega = \begin{pmatrix} \hat{y}_{r+1}^{(1)} & \hat{y}_{r+2}^{(1)} & \hat{y}_{r+3}^{(1)} & \hat{y}_{r+4}^{(1)} & \hat{y}_{r+5}^{(1)} & \hat{y}_{r+6}^{(1)} & \hat{y}_{r+7}^{(1)} & \dots & \hat{y}_n^{(1)} \\ & \hat{y}_{r+2}^{(2)} & \hat{y}_{r+3}^{(2)} & \hat{y}_{r+4}^{(2)} & \hat{y}_{r+5}^{(2)} & \hat{y}_{r+6}^{(2)} & \hat{y}_{r+7}^{(2)} & \dots & \hat{y}_n^{(2)} \\ & & \hat{y}_{r+3}^{(3)} & \hat{y}_{r+4}^{(3)} & \hat{y}_{r+5}^{(3)} & \hat{y}_{r+6}^{(3)} & \hat{y}_{r+7}^{(3)} & \dots & \hat{y}_n^{(3)} \\ & & & \hat{y}_{r+4}^{(4)} & \hat{y}_{r+5}^{(4)} & \hat{y}_{r+6}^{(4)} & \hat{y}_{r+7}^{(4)} & \dots & \hat{y}_n^{(4)} \\ & & & \ddots & \ddots & \ddots & \ddots & \dots & \hat{y}_n^{(k)} \end{pmatrix} \tag{5.6}$$

5.3 Coefficient Estimation

Different optimization algorithms can generate significantly different results in coefficient estimation. Two general types of optimization will be addressed in this work, Nelder-Mead local-search, and DDS global-search (algorithms discussed in Section 2.3.5.3). Both numerical optimizations will vary the coefficients to minimize the objective function but if the coefficients result in a model of non-stationary process (discussed in Section 2.3.4), the objective function will be returned as infinity. DDS and Nelder-Mead searches are applied in a slightly different manner.

The DDS global-search will perform $2,500$ objective function evaluations and return the coefficients that resulted in the lowest objective function value. No information

on convergence is returned.

The Nelder-Mead local-search will perform $1,000$ iteration and will either reach a convergence (ideal case) or not (maximum iterations reached). For models with many coefficients to optimize for, reaching convergence is often more difficult and because the starting parameters in the optimization algorithm have a significant effect on the local-search, the optimization is performed at most five times with different starting parameters (see Figure 5.7). These starting values are selected randomly from a distribution $\mathcal{N}(0, 1)$, and are always coefficients that undergo unit-root test to ensure stationary process. If the optimization does not reach convergence during these five repetitions, the coefficient set that returned the lowest objective function value (out of the five tries) is returned.

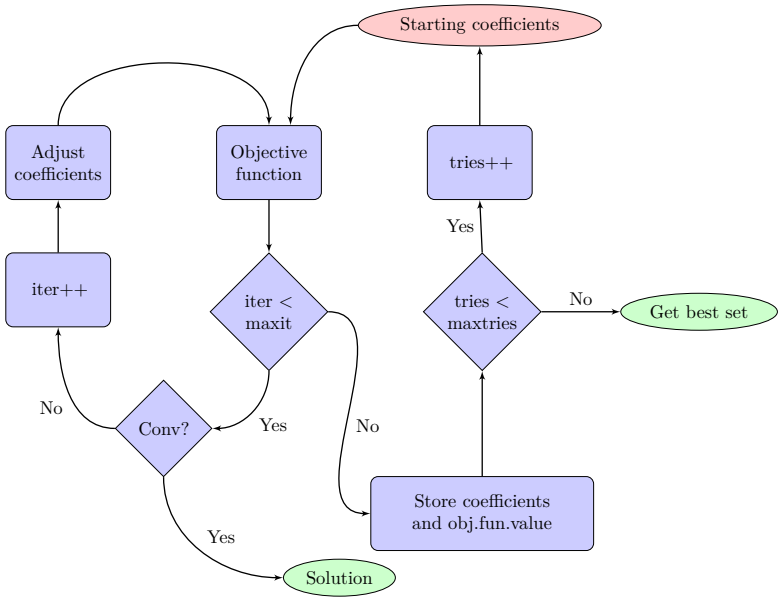


Figure 5.7: Nelder-Mead coefficient optimization tries to get the solution to converge at most 5 times. It the solution fails to converge, the coefficient set which provided the lowest objective function value is selected

5.4 Meta-Optimization

5.4.1 Hyper-Model Definition

For finding the best performing model, meta-optimization will be used to generate multiple hyper-models which will all be evaluated. Hyper-models will consist of both ARIMA and ARIMAX type models with different objective function criteria. Additionally, the model coefficients will be optimized with different optimization algorithms. Table 5.1 shows all different hyper-model-types.
After, calibrating and evaluating all hyper-models, the best performing model for operational purposes can be selected. Two evaluation metrics will be used (Sections 5.5).

Table 5.1: Best performing model for operational puposes will be selected from following hyper-models.

	Type	Objective function criteria	Coefficient optimization
1	ARIMA	Single-step	Nelder-Mead
2	ARIMA	Single-step	DDS
3	ARIMA	Multi-step	Nelder-Mead
4	ARIMA	Multi-step	DDS
5	ARIMAX	Single-step	Nelder-Mead
6	ARIMAX	Single-step	DDS
7	ARIMAX	Multi-step	Nelder-Mead
8	ARIMAX	Multi-step	DDS

5.4.2 Selection of Hyper-Model Parameters

Model identification is done by performing a grid search over a defined hyper-parameter search-space. The search-space suffers from the curse of dimensionality, a phenomenon where more hyperparameters rapidly result in increasing search-space. Hence, limiting search-space is important in terms of efficiency and computing time. This search-space is limited to the model hyper-parameters found in Table 5.2. The definitions and discussion of these ARIMA model parameters are presented in Section 2.3.1.

This exhaustive search produces an abundance of hyperparameter sets, a total of 162 ARIMA models, and $2,430$ for ARIMAX models that need to be calibrated. Figure 5.8 shows the histogram of the number of coefficients. This is important because more coefficients generally require more computing power and are harder to optimize than models with few coefficients. Calibrating all of the models is computationally expensive but what justifies this approach is that this exhaustive search is 'embar-

Table 5.2: Grid-seach will calibrate and evaluate each permutation of following hyperparameters

Parameter	ARIMA	ARIMAX
p	$0, 1, 2, 3, 4, 5, 6, 7, 8$	$0, 1, 2, 3, 4, 5, 6, 7, 8$
d	$0,\ 1$	$0,\ 1$
q	$0, 1, 2, 3, 4, 5, 6, 7, 8$	$0, 1, 2, 3, 4, 5, 6, 7, 8$
ρ	0	$2, 4, 6, 8, 10$
ρ_{lag}	0	$5, 10, 15$
Nr. models	162	2.430

rassingly parallel', meaning that many calibrations can be carried out at the same time. The parallelization is discussed in Section 5.4.3.

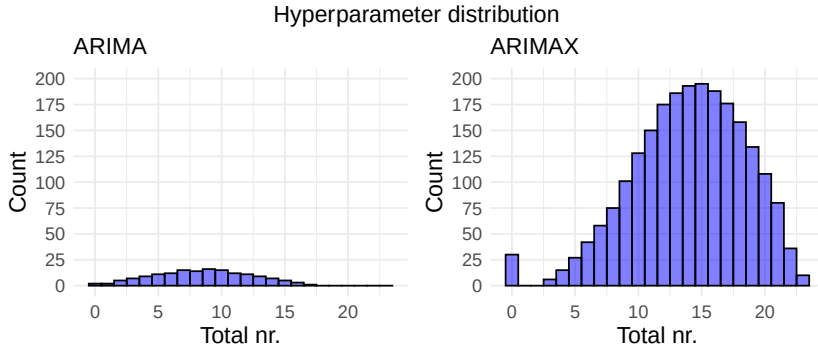


Figure 5.8: Histograms of the number of coefficients. Higher number of coefficients generally cost more in terms of computing power.

5.4.3 Parallel Computing

Doing grid-search over the hyperparameters results in many models that must be fitted, but since grid-search does not operate by any clever heuristic, and there is no dependency in the process, it can be fragmented into many parallel tasks. Taking advantage of parallelization, the search can be carried out on several different CPU (or GPU) cores parallelly.

The computations are carried out in the DTU high-performance cluster. Before initiating the computations, a function that optimizes the coefficients (based on single-or multi-step forecasts) was prepared, as well as a list with all of the hyperparameters. Next, a python script was made to pull each of the hyperparameter sets independently, call the optimization function on it, and to send the job to the HPC. Around

100-120 jobs (calibrations) can be carried out simultaneously. Once each job has finished, the results are saved on .rdata form.

5.5 Evaluation

Hydrological models will be evaluated with two distinct error measures:

- Persistence Index (PI) skill-score.
- Accuracy in predicting a simplified version of ATS activation.

PI measures whether forecasts perform better compared to a reference forecast of predicting the last observation. However, it will not capture the real-world consequences of using the model, but to assess that, accuracy in predicting ATS activations will also be evaluated.

Each of the models will be evaluated for forecasts with 30, 60, and 90-minutes forecasting horizon. It is worth mentioning that this work will not utilize real-world 'imperfect' radar forecasts as input but 'perfect' radar observations. The performance of a model that uses imperfect radar forecasts can thus be expected to fall between the performance of the model using no rainfall forecasts (ARIMA) and models using 'perfect' rainfall forecasts (ARIMAX).

5.5.1 Evaluation of Point Forecast

According to to [1] [31], forecast accuracy can be defined as the average degree of correspondence between a forecast and an observation. Error measures such as MAE and MSE are often used for measuring the accuracy of a specific forecast. Skill score is usually defined as in Equation 5.7, where the forecast accuracy is compared to the accuracy of some standard of reference.

$$\mathrm{SS} = \frac{A_f - A_r}{A_p - A_r} \tag{5.7}$$

Where A_f is the accuracy (MSE) of the forecast, A_r is the reference forecast, and A_p is the accuracy of the perfect forecast. Because the accuracy (in case of MSE) of a perfect forecast is 0, a skill score can be simplified to Equation 5.8.

$$\begin{aligned} \mathrm{SS} &= \frac{A_f - A_r}{0 - A_r} \\ &= 1 - \frac{A_f}{A_r} \end{aligned} \tag{5.8}$$

Skill scores range from negative infinity to 1. Negative values imply that the forecast performs worse than the standard of reference. A value of 0 indicates that the forecast is performing equivalent to the standard-of-reference. A positive value indicates that the forecast performs better than the standard of reference. A skill score of 1 entails a perfect forecast.

The skill score that will be used in this work is the Persistence Index (Equation 5.9), a skill score that uses the most recently observed value as a standard of reference [26].

$$\mathrm{PI} = 1 - \frac{\frac{1}{n}\sum_{i=1}^{n}\left(y_i - \widehat{y}_i\right)^2}{\frac{1}{n}\sum_{i=1}^{n}\left(y_i - y_{i-1}\right)^2} \tag{5.9}$$

The reason for choosing Persistence Index as the skill score is because it measures whether the models are generating forecasts that are more accurate than if the most recently observed value would just be forecasted.

Other skill-scores exist such as the Nashe-Sutcliffe efficiency (NSE), which is often used in hydrology, but since NSE uses the mean of observations as the reference forecast and this work focuses on forecasting extreme events, it does not prove useful skill-score in this case. Error metrics that capture the physical interpretation of the model performance are often used, such as round mean square error (RMSE) and mean absolute error (MAE). In this work, the physical interpretations of the errors are not sought for, and hence, these error metrics will not be used. Additionally, these error measures can be expected to correlate heavily with PI.

5.5.2 Performance of ATS Activation

Skill-score (and classical error metrics) only give some idea on how close the forecast are to the observations, but do not give any explicit information on the real-world consequences of the models. Hence, another error metric is proposed that tries to measure the real-world consequences of the models, based on their performance in forecasting a simplified version of ATS activation.

Aeration tank settling (ATS) activation is a process in which WWTP is prepared for an increase in a runoff by allowing sludge to settle in the aeration tanks. This enables more suspended solids to be stored in the aeration tanks during periods where runoff is increased, increasing the hydraulic capacity of the WWTP. ATS activations must be forecasted some time in advance due to a buffer time in which the system needs to be physically prepared with recirculations of biomass from the clarifiers to the aeration tanks. More detailed discussion of modelling ATS activations can be found in [23] and cite [13].

As the ATS activation depends on many factors other than just inflow, a simplified version of an ATS activation is presented here, denoted WET activation. A WET activation will be defined as a process in which the WWTP goes from dry-weather control to wet-weather control. A WET activation will be turned on if inflow reaches 5000 m3/h. The same approximation has been used by Krüger where the evaluation of the model in forecasting WET activations was following [8] :

- Correct (TP_c): If WET activation was predicted within a 75 min interval (from 60 min before measured flow exceeded the threshold to 15 min after)
- Early (TP_e): If WET activation was predicted earlier than 60 min before the measured flow exceeded the threshold
- False alarm (FP): WET event forecasted, but measured runoff did not exceed the threshold.
- Missed (FN): if WET activation was predicted more than 15 min after the measured flow exceeded the threshold or if no WET was predicted (False Negative)
- Down: no data are available for the specific time interval where the WET event took place (i.e., it is not possible to evaluate the performance of the forecast)

Figure 5.9 depicts how forecasts are classified in terms of correct, early or missed, but if a forecast can also be classified as false alarms.

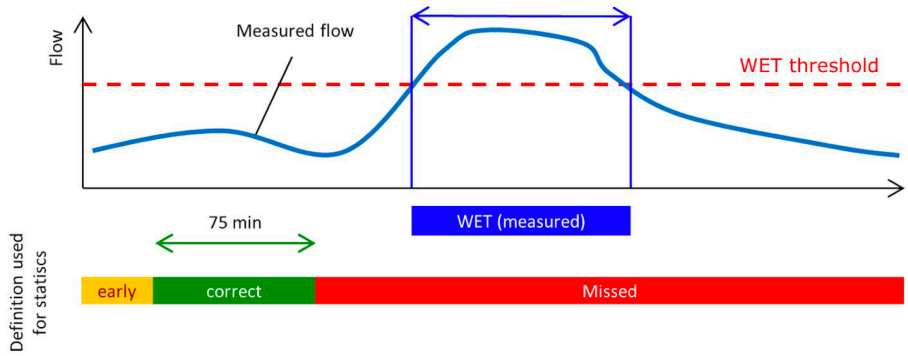


Figure 5.9: Predicting ATS activation can be early, correct, missed or false alarm. Figure from [8]

Predicting WET activation correctly is the ideal scenario where the system prepares itself for an increase in runoff successfully. During WET activation, wastewater is not cleaned as well, releasing biological waste into the environment. Hence, the early

prediction will have a period where the system cleans the wastewater inadequately. False alarms are even worse, as the system cleans wastewater inadequately during the whole WET activation. Finally, missed WET activation is the worst-case scenario as it risks CSO.

Having classified each WET event in terms of correct, missed, early, and false alarms, binary classification metrics can be used to evaluate the models' predictive quality. Accuracy is a statistical measure of the closeness of the measurements to the correct observations by using TP, FP, and FN. Because wet events that are forecasted early cannot be considered false alarms, missed events, nor entirely correct, they will be treated as a particular cause close to being correct. Hence, two accuracies metrics will be defined, the accuracy of correctly predicting WET activation (Equation 5.10), and the accuracy of predicting WET activation either correctly, or early (Equation 5.11).

$$\text{Accuracy}_c = \frac{\text{TP}_c}{\text{TP} + \text{FP} + \text{FN}} \tag{5.10}$$

$$\text{Accuracy}_{c+e} = \frac{\text{TP}_c + \text{TP}_e}{\text{TP} + \text{FP} + \text{FN}} \tag{5.11}$$

5.6 Tools

5.6.1 R

R is a powerful free software used for statistical computations and graphics. R offers a wide variety of powerful libraries for data analysis and plotting. Rstudio is an IDE for the R language. A large proportion of the project was carried out in Rstudio 1.2.1335 using R 3.6.2. In the high-performance cluster, R versions 3.5.1 was used. The following R libraries were used throughout the project.

- **ggplot2**: A powerful package used for producing high quality graphics
- **xts**: Extendable time series packate that makes easy handling of time-based data classes
- **dygraphs**: Package the uses xts time series for plotting highly interactive plots for highlighting, zooming/panning and various graph overlays
- **scales**: Scaling time based variables when plotting
- **ggpubr**: Arranging multiple plots produced by ggplot2.

5.6.2 High Performance Cluster

Ssh connection was used to connect to the HPC. Bash is the language used in UNIX machines, and working in the HPC requires knowledge in Bash to manage files and send jobs.

5.6.3 Python3

Python3 is used to run a Python program that allows parallel coefficient estimation. The Python code that takes multiple sets of model hyper-parameters from the search-space, and passes them to the HPC as independent jobs. Each job passes the model hyper-parameters into an R function where the model coefficients are estimated.

CHAPTER 6

Results

As this work considers various types of hyper-models, the results will be broken down into a few subsections, where each subsection narrows down the model selection. The first thing considered is whether global (DDS) or local (Nelder-Mead) optimization generates better performing models. Next, it is investigated how different objective function criteria affect the performance of the hyper model (i.e. single-step forecast or multi-step forecasts) and whether external regressors improve performance. Finally, the inference and model selection will be made.

6.1 Comparisons of Optimization Methods in Coefficient Estimation

6.1.1 Computing Time

The most notable difference between Nelder-Mead and the DDS optimizations is their computing time. Figure 6.1 compares the average computing time for multi-step models on Damhusåen between DDS and Nelder-Mead optimization. The choice of using multi-step models here is arbitrary but similar results are obtained for single-step calibrated models on both catchments (Appendix B.1.1).
When optimizing a few numbers of coefficients, Nelder-Mead optimization has a lower computing time, but for more coefficients (around ≥ 6), DDS optimization takes much less time. This is likely because, for few coefficients, Nelder-Mead reaches convergence relatively fast and stops as soon as convergence is reached. However, DDS optimization always carries out $2,500$ iterations..

Another thing to note from Figure 6.1 is that for DDS, the computing time goes down when the number of coefficients is increased, something that is quite counterintuitive. The reason for the decrease in computing time lies in the calibration procedure. To ensure stationarity, models are checked for unit-roots (non-stationary process) for each iteration, but if the coefficients are of the non-stationary process, the objective function is not calculated and simply returned as infinity.
Both optimization algorithms start with coefficients that ensure stationary process but because Nelder-Mead is a local-search algorithm, the number of times coefficient iteration results in non-stationarity are less common. However, for DDS global-search,

the majority of iterations result in non-stationarity, where the objective function is not calculated. Figure 6.2 shows the computing time and the number of times infinity is returned for 100 iterations. More coefficients generally have a higher likelihood of resulting in non-stationarity (up to a certain point) and hence, with more coefficients the objective function is more frequently not calculated, explaining why computing time decreases with more coefficients.

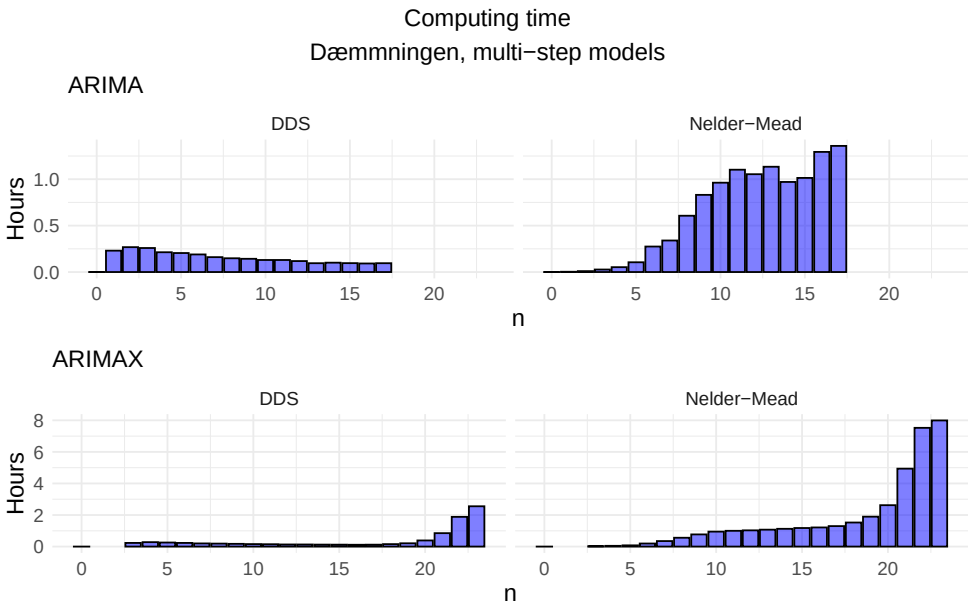


Figure 6.1: Nelder-Mead local-search is on average much more computationally expendsive than DDS global-search.

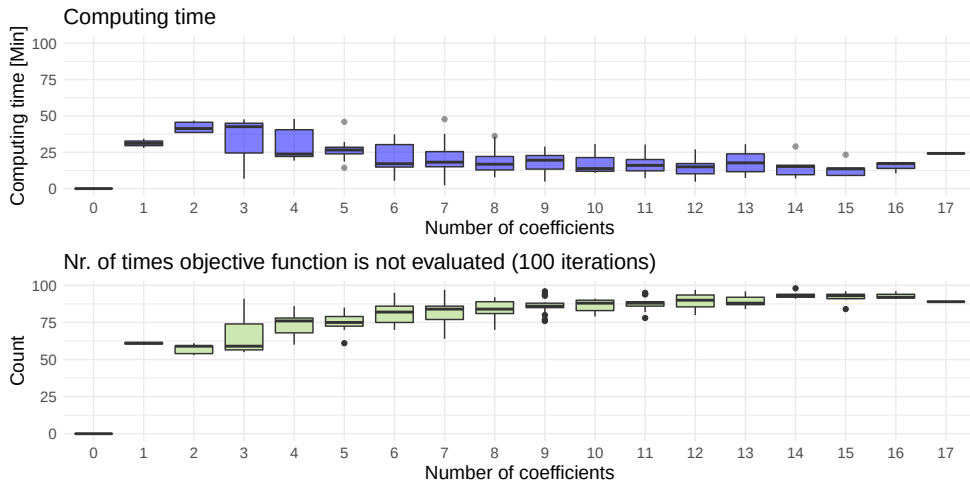


Figure 6.2: If coefficicients result in non-stationarity, the objective function is not calculated and just returned as infinity, explaining why computing time decreases (up to some point) for more coefficients in DDS optimization

6.1.2 Minimized Objective Function

Often, different optimization algorithms differ in their ability to minimize the objective function. Figure 6.3 shows violin plots of the minimized objective function value against the number of coefficients that are used for fitting multi-step models on Dæmningen. As seen in Figure 6.3, Nelder-Mead can reach a lower objective function value than DDS can. The mean (blue dot) and median (red dot) can also be seen to generally be lower for Nelder-Mead. Even though DDS can obtain fairly low objective function values, its distribution is much more spread out than the distribution on Nelder-Mead, which can be seen quite successful in minimizing most models. The difference between the coefficient optimization is more obvious for multi-step models than for single-step models but distributions of Damhusåen and single-step models can be found in Appendix B.1.2

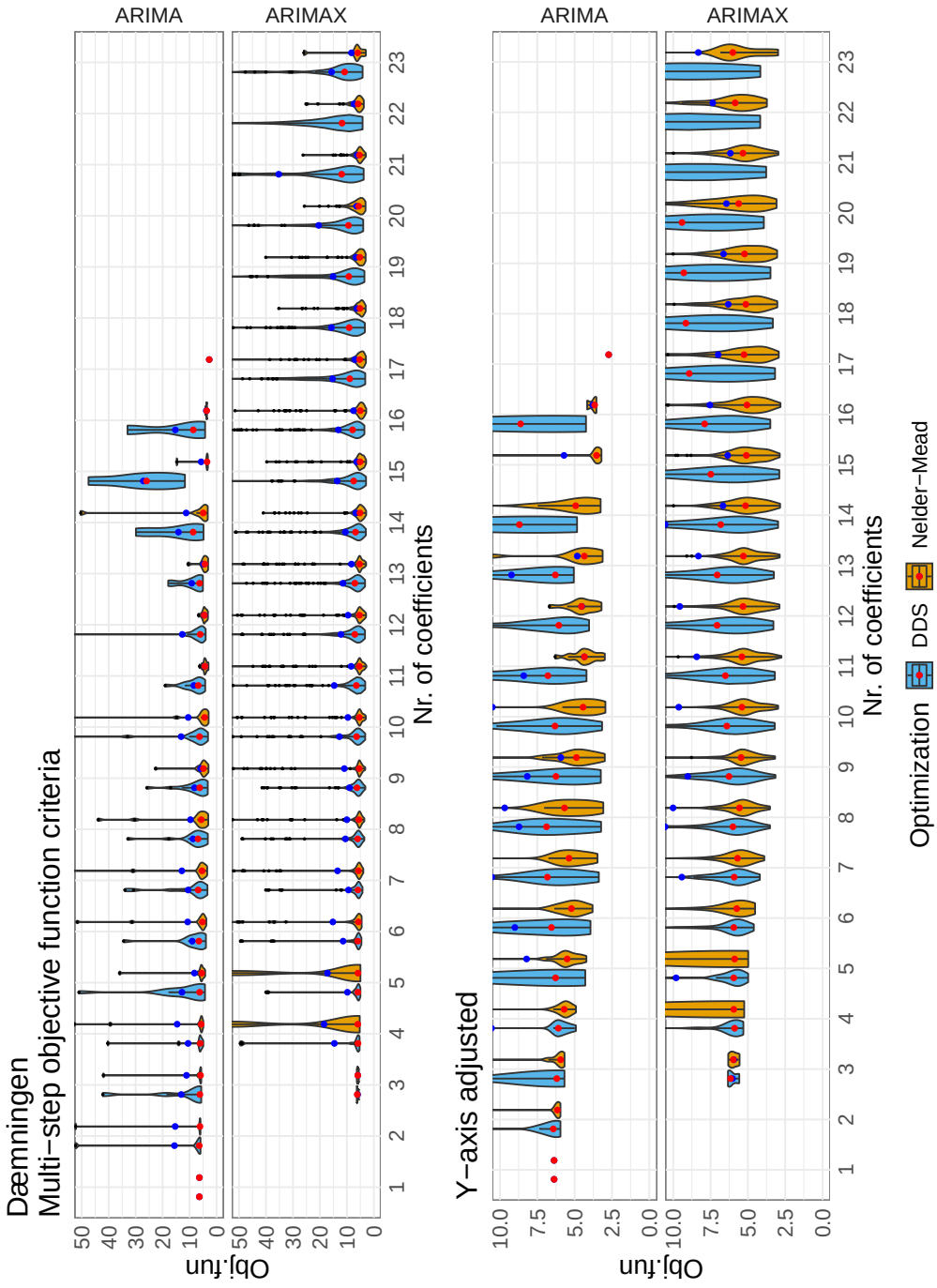


Figure 6.3: Nelder-mead is more sucessfull in achiveing low objective function value. Scoring criterium for multi-step objective function criteria is found in Equation 5.2. Blue dot is mean, red dot is median.

Looking at the difference in the ability to minimize objective function is not enough to assess which optimization algorithm is better but hence, the model is also evaluated on the validation data. Figure 6.4 shows the violin plot of PI value for all hyper-models. For Dæmningen, the distribution of DDS optimized models is much more spread out than the distribution of Nelder-Mead optimized models. Additionally, by inspecting the mean (blue dot) or adjusting the y-axis, it can be seen that Nelder-Mead can generate values with higher PI skill-score. For Damhusåen, the distributions are quite similar but the average (blue dot), PI skill-score achieved with Nelder-Mead coefficient optimization is a lot higher than of models optimized with DDS. When the y-axis is adjusted, it is observed that and DDS can generate models with higher PI skill-score based on 30 and 60-minute forecasting horizon.

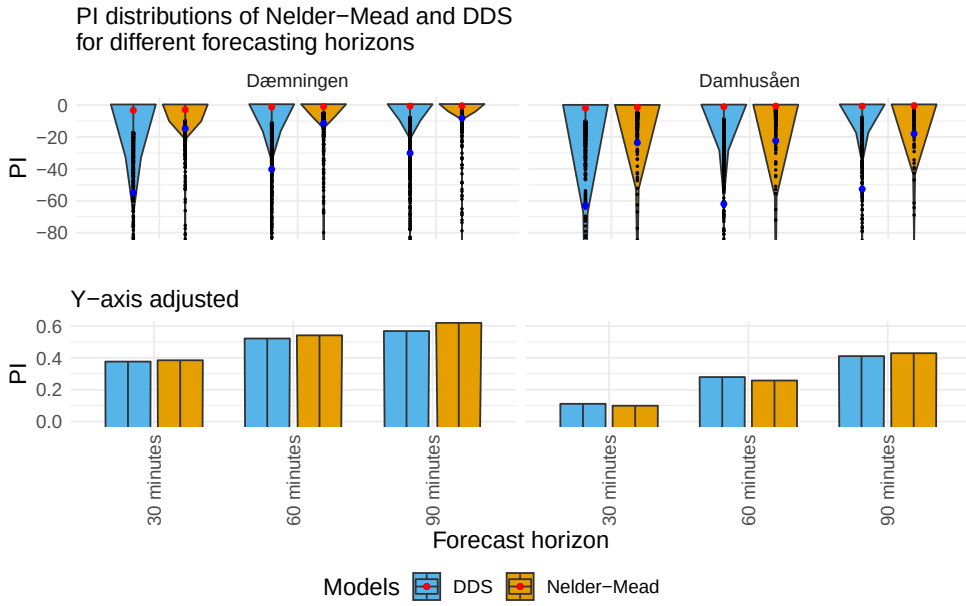


Figure 6.4: Nelder-Mead coefficient optimization can generate models with higher PI skill-score for Dæmningen. Damhusåen generally has higher PI skill-score for DDS optimization. Blue dot is mean, red dot is median.

Similar information is seen in Figure 6.5 where the y-axis represents the accuracy of correct alarms. The DDS optimization generally produces a model with higher accuracy.

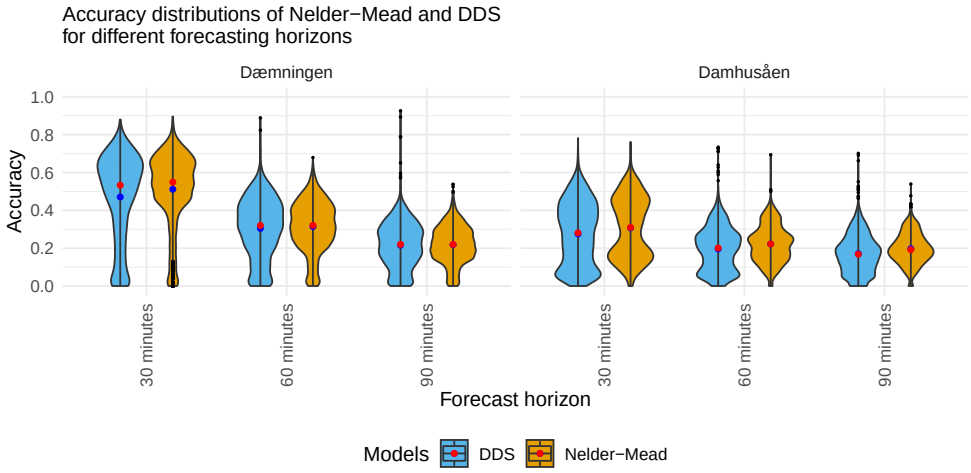


Figure 6.5: The models with the highest accuracy are generally DDS optimized models. Blue dot is mean, red dot is median.

6.2 Comparisons of Nelder-Mead optimized Hyper-Models

This section investigates whether there is a significant difference between the hyper-models (i.e., single/multi-step objective function criteria and ARIMA/ARIMAX models). To keep the discussion concise, the investigation will be limited to hyper-models that have optimized coefficients with Nelder-Mead optimization. However, equivalent figures for DDS optimized models can be found in Appendix B.2

6.2.1 Influence of Regressors on Objective Function

A comparison of the minimization of the objective function between ARIMA and ARIMAX models can be seen in Figure 6.6. The model with external regressors can, on average, get lower objective function values than models without external regressors. Dæmningen achieves its lowest objective function for lag 5 and 4 regressors (corresponding to regressors on lag 6, 7, 8, and 9) while Damhusåen for lag 5 and 6 regressors (corresponding to regressors on lag 6 - 11). The similar figure for DDS search can be seen in Figure B.7 but since this is the average objective function value, the methods differ on the regressors with the lowest value.

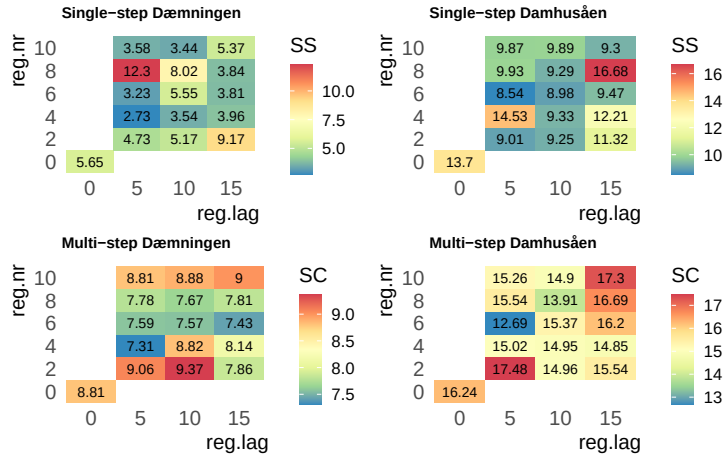


Figure 6.6: Average minimization of the objective function with Nelder-Mead search associated with lags and number of external regressor. Objective function (f) is sum-of-squares for single-step models and SC (Equation 5.2) for multi-step models.

6.2.2 Objective Function Criteria

Violin plots of evaluation metrics of the hyper-models are shown in Figure 6.7. The multi-step hyper-models generally have higher PI skill score whereas the single-step hyper-models generally have higher accuracy. When hyper-models with the same objective function criteria are compared with and without external regressors, it can be observed that models with external regressors can achieve higher PI skill-score and accuracy than models without external regressor, especially for long forecasting horizon. However, higher average PI/accuracy (blue dots) or median (red dots) are not observed. For models with coefficients optimized with DDS, similar things are observed (Figure B.8)

To investigate whether a selection of a model based on an evaluation-metric for certain forecasting horizon results in a good performance for other forecasting horizons, consider Figure 6.8. Best models are selected based on PI (upper figure) and accuracy (lower figure) for 30, 60, and 90-minute forecasting horizon. Different colors and shapes represent different objective function criteria and the performance of the selected model for other forecasting horizons. When models are selected on the PI for longer forecasting horizon, the very best models are overfitted and give a bad performance on the shorter horizons, especially for multi-step hyper-models. The very best models based on accuracy have very high accuracy for short-horizon but lower for longer horizons. The similar figure for DDS optimized models can be seen in Figure B.9

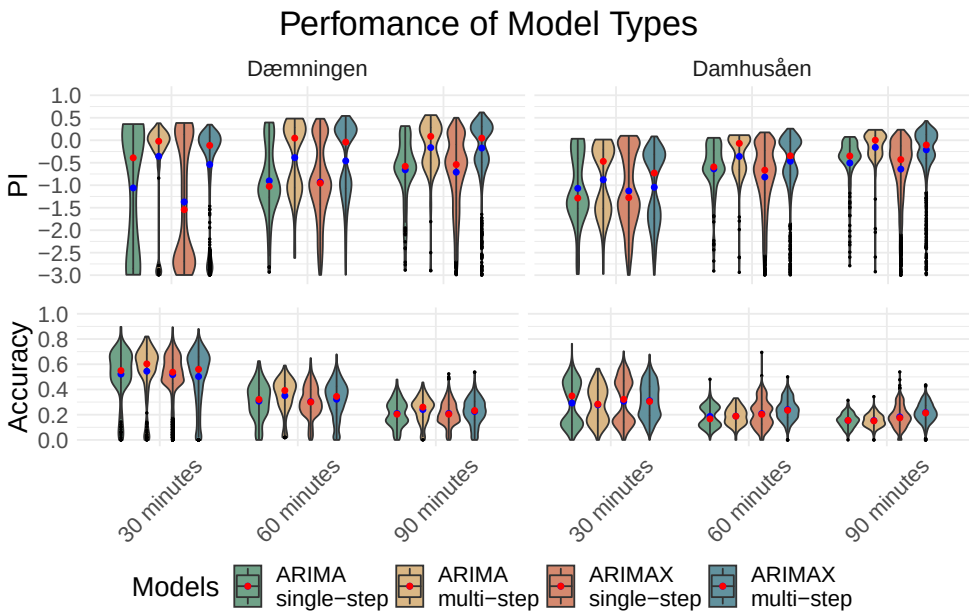


Figure 6.7: Multi-step models generally have higher PI skill-score while accuracy seems to be higher for single-step models.

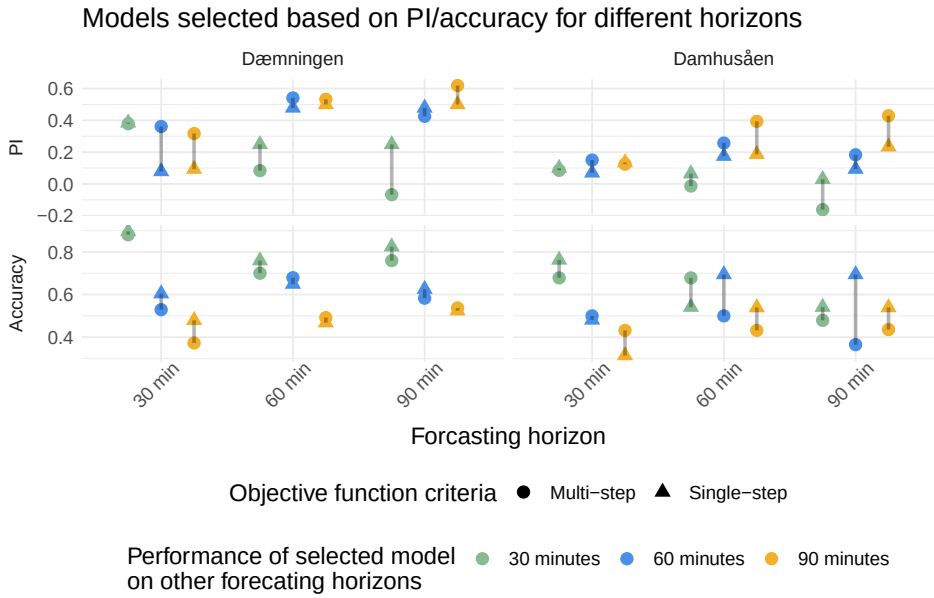


Figure 6.8: The best performing models for differenct forcasting horizon and objective function criteria. Multi-step models are more overfitted than the single-step models, especially for longer forecasting horizons.

6.3 Selection of Best Performing Model for Operational Purpuses

6.3.1 On Evaluation Measures

To examine whether selecting on one of the evaluation metrics results in a good performance in the other evaluation metric, models were selected on both evaluation metrics for 90-min forecasting horizon on Dæmningen (Table 6.1) and Damhusåen (Table 6.2)

From the table, it can be observed that:

- All of the top-10 performing models based on the PI with a 90-min forecasting horizon are overfitted multi-step models that have poor performance for other forecasting horizons. Selecting other forecasting horizons gave similar results. Hence, PI skill-score might not be ideal criteria for selecting models in the meta-optimization.
- When the models are selected on the accuracy, we get good accuracy, but terrible skill-score values. Hence, accuracy should not be used for selecting models in the meta-optimization as high accuracy models give poor PI-skill scores.

Table 6.1: Best perfoming models of Dæmningen when selected on PI/Accuracy for 90-min forecasting horizon. Selecting based on PI gives overfitted models that perfom badly for other forecasting horizon (especially 30 min), but decent accuracy. Selecting on accuracy gives terrible PI skill-score.

Dæmningen													
Model					Persistence Index			Accuracy					
	ofc	order	ρ_{nr}	ρ_{lag}	30 min	60 min	90 min	correct 30 min	+early 30 min	correct 60 min	+early 60 min	correct 90 min	+early 90 min
Skill score of best performing models based on PI for 90 minute forecasting horizon													
1	multi-step	0, 1, 8	8	5	-0.07	0.43	0.62	0.84	0.84	0.55	0.55	0.47	0.47
2	multi-step	1, 1, 8	4	5	-0.01	0.38	0.59	0.65	0.65	0.46	0.46	0.29	0.29
3	multi-step	0, 1, 8	6	5	0.04	0.53	0.58	0.76	0.76	0.56	0.56	0.39	0.39
4	multi-step	1, 1, 8	8	5	-0.05	0.39	0.58	0.62	0.62	0.38	0.38	0.33	0.33
5	multi-step	1, 1, 8	6	5	0.03	0.51	0.57	0.63	0.63	0.47	0.47	0.30	0.30
6	multi-step	4, 1, 7	10	5	-0.05	0.39	0.57	0.63	0.63	0.39	0.39	0.33	0.33
7	multi-step	3, 1, 8	4	10	-0.07	0.38	0.57	0.71	0.71	0.47	0.47	0.35	0.35
8	multi-step	3, 1, 7	4	10	-0.03	0.46	0.57	0.67	0.67	0.41	0.41	0.37	0.37
9	multi-step	2, 1, 7	4	5	0.06	0.47	0.56	0.66	0.66	0.39	0.39	0.32	0.32
10	multi-step	1, 1, 7	4	10	0.03	0.45	0.56	0.60	0.60	0.36	0.36	0.33	0.33
Skill score of best performing models based on accuracy for 90 minute forecasting horizon													
1	multi-step	1, 0, 6	10	5	-3.94	-1.13	-0.22	0.76	0.77	0.58	0.60	0.54	0.55
2	multi-step	7, 0, 7	4	5	-3.68	-1.07	-0.33	0.62	0.63	0.59	0.60	0.54	0.55
3	single-step	3, 0, 6	8	10	-172.94	-51.49	-25.62	0.82	0.88	0.62	0.70	0.52	0.60
4	single-step	4, 1, 6	10	5	-3.67	-2.06	-1.94	0.89	0.90	0.62	0.65	0.50	0.53
5	multi-step	5, 0, 5	8	5	-4.06	-0.90	-0.31	0.70	0.70	0.68	0.69	0.49	0.50
6	single-step	6, 0, 6	6	15	-2.57	-0.57	-0.28	0.75	0.75	0.65	0.66	0.49	0.49
7	single-step	2, 0, 2	4	5	-3.12	-1.14	-0.31	0.74	0.75	0.53	0.55	0.48	0.50
8	single-step	6, 1, 8	6	5	-0.44	-0.03	0.22	0.79	0.79	0.62	0.62	0.48	0.48
9	multi-step	6, 0, 8	2	5	-3.53	-0.89	-0.25	0.63	0.64	0.54	0.54	0.48	0.49
10	single-step	4, 1, 7			-2.02	-0.70	-0.47	0.90	0.90	0.60	0.63	0.48	0.50

Table 6.2: Best perfoming models of Damhusåen when selected on PI/Accuracy for 90-min forecasting horizon. Same clues are observed as were in Table 6.1.

Damhusåen													
Model					Persistence Index			Accuracy					
	ofc	order	ρ_{nr}	ρ_{lag}	30 min	60 min	90 min	correct 30 min	+early 30 min	correct 60 min	+early 60 min	correct 90 min	+early 90 min
Skill score of best performing models based on PI for 90 minute forecasting horizon													
1	multi-step	1, 1, 8	8	5	-0.16	0.18	0.43	0.48	0.50	0.36	0.39	0.29	0.32
2	multi-step	2, 1, 6	2	5	-0.21	0.21	0.43	0.45	0.47	0.32	0.34	0.26	0.29
3	multi-step	4, 1, 6	2	5	-0.05	0.21	0.42	0.45	0.48	0.34	0.37	0.27	0.30
4	multi-step	3, 1, 6	2	5	-0.13	0.21	0.42	0.45	0.47	0.31	0.34	0.28	0.31
5	multi-step	3, 1, 7	2	5	-0.11	0.19	0.41	0.44	0.46	0.34	0.35	0.30	0.32
6	multi-step	1, 1, 7	2	5	-0.23	0.22	0.41	0.46	0.48	0.33	0.36	0.29	0.31
7	multi-step	4, 1, 7	2	5	-0.09	0.23	0.41	0.50	0.51	0.37	0.38	0.33	0.35
8	multi-step	1, 1, 5	2	5	-0.15	0.24	0.40	0.42	0.44	0.32	0.33	0.26	0.27
9	multi-step	2, 1, 5	2	5	-0.14	0.23	0.40	0.51	0.52	0.38	0.39	0.29	0.31
10	multi-step	7, 1, 5	2	5	-0.01	0.26	0.39	0.39	0.40	0.31	0.32	0.22	0.23
Skill score of best performing models based on accuracy for 90 minute forecasting horizon													
1	single-step	6, 0, 5	8	15	-5616.00	-4166.20	-2262.60	0.54	0.86	0.69	0.81	0.54	0.67
2	single-step	3, 0, 2	2	15	-611.02	-532.26	-436.38	0.66	0.82	0.51	0.66	0.48	0.57
3	multi-step	6, 1, 8	4	5	-0.95	-0.11	0.14	0.48	0.48	0.36	0.39	0.44	0.46
4	single-step	2, 1, 4	4	5	-1.90	-2.18	-1.67	0.60	0.65	0.46	0.53	0.43	0.52
5	multi-step	6, 1, 0	4	5	-0.49	0.01	0.01	0.68	0.70	0.50	0.54	0.43	0.48
6	multi-step	0, 0, 8	10	5	-4.92	-2.84	-2.39	0.26	0.36	0.42	0.52	0.42	0.53
7	multi-step	8, 1, 8	4	5	-0.68	-0.16	0.16	0.50	0.53	0.37	0.43	0.42	0.46
8	multi-step	8, 1, 8	2	5	-0.38	0.08	0.29	0.55	0.58	0.45	0.50	0.42	0.46
9	single-step	3, 1, 8	2	5	-3.20	-4.34	-2.53	0.54	0.56	0.37	0.42	0.41	0.47
10	single-step	1, 1, 8	4	5	-0.14	-0.15	-0.13	0.55	0.57	0.38	0.44	0.41	0.45

6.3.2 Model selection

In Section 6.3.1, it was inferred that the model should not be selected based on accuracy. Additionally, selecting a model based on PI skill-score can be complicated due to overfitting, especially when selecting a model based on the PI with longer forecasting horizons, where the model had high PI value for the selected forecasting horizon, but poor performance on other forecasting horizons. The reason for overfitting is that the PI evaluates model performance based on the benchmark, which is the last observation. With longer forecasting horizon, the benchmark is further away from the predicted values.

To overcome this, model selection can be done on the highest average skill-score (i.e. the average of PI skill-scores for 30, 60, and 90 min forecasting horizon). In Table 6.3, 10 models have been selected based on their average skills-score on both catchments, where coefficients were optimized with Nelder-Mead optimization. Table 6.4 shows similar table but for models that use DDS for coefficient estimation.

Table 6.3: Top-10 perfoming Nelder-Mead optimized models based on average PI skill-score for both catchments.

Skill score of best performing Nelder-Mead optimized models based on average-PI													
Model					Persistence Index			Accuracy					
	ofc	order	ρ_{nr}	ρ_{lag}	30 min	60 min	90 min	correct 30 min	+early 30 min	correct 60 min	+early 60 min	correct 90 min	+early 90 min
Dæmningen													
1	single-step	1, 1, 7	4	10	0.34	0.47	0.45	0.75	0.75	0.38	0.38	0.33	0.33
2	single-step	1, 1, 7	8	15	0.25	0.48	0.50	0.75	0.75	0.58	0.58	0.43	0.43
3	single-step	1, 1, 8	2	10	0.35	0.47	0.40	0.66	0.66	0.40	0.40	0.31	0.31
4	single-step	0, 1, 8	6	15	0.35	0.42	0.42	0.70	0.70	0.45	0.45	0.37	0.37
5	single-step	1, 1, 7	6	5	0.34	0.42	0.42	0.73	0.73	0.43	0.43	0.27	0.27
6	single-step	0, 1, 8	8	5	0.35	0.42	0.40	0.69	0.69	0.33	0.33	0.28	0.28
7	single-step	2, 1, 6	8	5	0.35	0.46	0.36	0.58	0.58	0.39	0.39	0.23	0.23
8	single-step	1, 1, 8	4	10	0.36	0.40	0.40	0.62	0.62	0.31	0.31	0.29	0.29
9	single-step	1, 1, 6	2	5	0.38	0.41	0.37	0.63	0.63	0.39	0.39	0.23	0.23
10	multi-step	0, 1, 6	2	5	0.15	0.52	0.49	0.73	0.73	0.56	0.57	0.36	0.37
Damhusåen													
1	multi-step	7, 1, 5	2	5	-0.01	0.26	0.39	0.39	0.40	0.31	0.32	0.22	0.23
2	multi-step	4, 1, 6	2	5	-0.05	0.21	0.42	0.45	0.48	0.34	0.37	0.27	0.30
3	multi-step	4, 1, 3	4	5	0.06	0.18	0.33	0.64	0.66	0.47	0.48	0.35	0.39
4	multi-step	4, 1, 7	2	5	-0.09	0.23	0.41	0.50	0.51	0.37	0.38	0.33	0.35
5	multi-step	5, 1, 3	4	5	0.06	0.21	0.27	0.51	0.52	0.38	0.39	0.28	0.32
6	multi-step	1, 1, 8	2	5	-0.08	0.24	0.39	0.49	0.51	0.34	0.36	0.29	0.32
7	multi-step	4, 1, 3	2	5	0.06	0.19	0.28	0.53	0.56	0.39	0.40	0.29	0.33
8	multi-step	4, 1, 2	6	5	0.03	0.18	0.31	0.51	0.52	0.39	0.41	0.34	0.36
9	multi-step	3, 1, 3	4	5	0.05	0.17	0.28	0.57	0.57	0.41	0.42	0.32	0.35
10	multi-step	6, 1, 2	2	5	0.06	0.17	0.28	0.57	0.59	0.39	0.40	0.33	0.34

Table 6.4: Top-10 perfoming DDS optimized models based on average PI skill-score for both catchments.

Skill score of best performing DDS optimized models based on average-PI													
Model					Persistence Index			Accuracy					
	ofc	order	ρ_{nr}	ρ_{lag}	30 min	60 min	90 min	correct 30 min	+early 30 min	correct 60 min	+early 60 min	correct 90 min	+early 90 min
Dæmningen													
1	single-step	0, 1, 7	8	5	0.34	0.48	0.40	0.67	0.67	0.42	0.43	0.31	0.31
2	single-step	1, 1, 8	2	15	0.34	0.45	0.37	0.62	0.62	0.43	0.43	0.22	0.22
3	single-step	0, 1, 8	4	5	0.30	0.42	0.44	0.77	0.77	0.50	0.50	0.32	0.32
4	single-step	1, 1, 6	10	5	0.32	0.40	0.41	0.69	0.69	0.35	0.35	0.35	0.35
5	multi-step	0, 1, 8	2	10	0.08	0.49	0.53	0.71	0.71	0.49	0.49	0.38	0.38
6	single-step	0, 1, 7	4	15	0.33	0.43	0.35	0.60	0.60	0.35	0.35	0.20	0.20
7	single-step	1, 1, 6	2	5	0.36	0.39	0.35	0.63	0.63	0.38	0.38	0.24	0.24
8	single-step	1, 1, 7	2	5	0.37	0.41	0.32	0.58	0.58	0.37	0.37	0.20	0.20
9	multi-step	1, 1, 8	6	5	0.03	0.52	0.54	0.67	0.67	0.49	0.49	0.32	0.32
10	single-step	0, 1, 5	8	5	0.37	0.38	0.35	0.71	0.71	0.40	0.40	0.26	0.26
Damhusåen													
1	multi-step	3, 0, 5	4	5	0.01	0.27	0.37	0.52	0.57	0.42	0.45	0.33	0.38
2	single-step	2, 1, 8	2	5	0.08	0.17	0.32	0.53	0.54	0.35	0.38	0.32	0.34
3	multi-step	1, 1, 5	2	5	-0.10	0.22	0.40	0.45	0.47	0.33	0.34	0.27	0.29
4	multi-step	4, 1, 4	2	5	-0.01	0.18	0.30	0.51	0.53	0.32	0.34	0.25	0.28
5	multi-step	3, 1, 6	2	5	-0.15	0.19	0.41	0.41	0.43	0.32	0.33	0.28	0.31
6	multi-step	0, 1, 6	2	5	-0.09	0.18	0.35	0.43	0.47	0.28	0.31	0.27	0.30
7	single-step	1, 1, 8	2	5	0.11	0.13	0.20	0.56	0.58	0.40	0.42	0.33	0.37
8	single-step	7, 0, 8	4	5	-0.17	0.22	0.38	0.50	0.52	0.43	0.44	0.38	0.40
9	multi-step	4, 1, 0	2	5	0.05	0.15	0.23	0.51	0.52	0.37	0.37	0.32	0.34
10	multi-step	2, 1, 5	2	5	-0.17	0.20	0.40	0.47	0.49	0.36	0.37	0.26	0.28

From Tables 6.3 and 6.4, it can be observed that both optimization methods can generate decent models. However, with Nelder-Mead coefficient estimation, the models are a wee bit better than the DDS optimized models. This is consistent with what was observed when the optimization methods were compared in their ability to minimize the objective function and their performance in generating a good PI skill-score.
Based on the top-10 performing Nelder-Mead models in Table 6.3, the results will conclude with model selection in Table 6.5.

6.4 Real-World-Forecast of selected model

The real-world forecasts of the selected model are shown in Figure 6.9. Dæmningen is has a much slower increase than Damhusåen, and based on its model preference, it has regressors going much further back. Forecast generated on Dæmningen produce these long arms where predicted values are not far off last observed values. However, the selected model for Damhusåen has this oscillatory behavior.
Real-world forecasts of all model shown in Table 6.3 can be found in Appendix B.3

Table 6.5: Each catchment has noticable 'preference' in model selection.

Dæmningen	**Damhusaåen**
Best performing models are models with single-step ojbective function criteria	Best performing models are models with multi-step ojbective function criteria
Models are differentiated (d = 1), have short AR, and long MA terms	Models are differentiated (d = 1), and have medium AR/MA terms
External regressors on lags further back	External regressors on lags which are closer
PI values decent for all forecasting horizons	PI values get smaller for shorter forecasting horizon
Accuracy decent, especially for model 2	Accuracy decent, especially for model 3
Models selected	
Model 2 from Table 6.3 $(1, 1, 7, 8, 15)$	Model 3 from Table 6.3 $(4, 1, 3, 4, 5)$

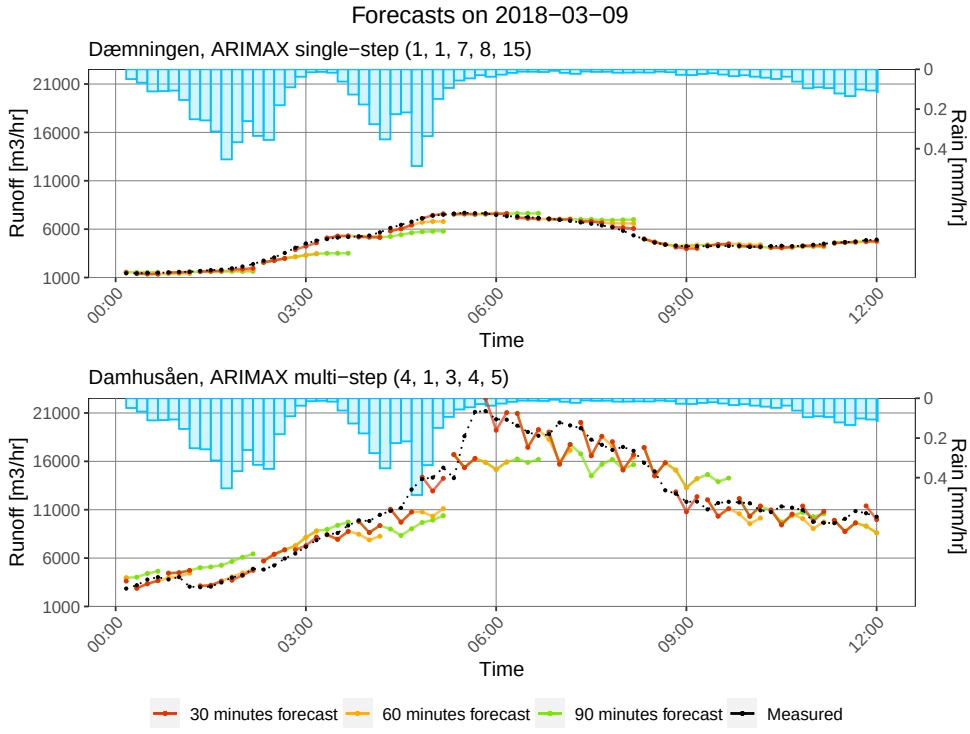


Figure 6.9: Real-word forecasts with the selected models: single-step $(1, 1, 7, 8, 15)$ for Dæmningen, and $(4, 1, 3, 4, 5)$ for Damhusåen.

CHAPTER 7

Discussion

This chapter will return to the research questions asked in Section 1.2 and focuses on answering and explaining each of the questions.

R.Q.1 **Can competent ARIMA type models be selected in an automated and efficient manner?**
Yes. With the implementation introduced in this thesis, ARIMA type models can be selected in an automated fashion and have reasonably good PI skill-score and accuracy in forecasting a simplified version of ATS activation. For efficiency, calibrations must be done in parallel.

R.Q.2 **How should the parameter search-space be constrained such that parameter selection can be performed in a computationally efficient manner, while still selecting parameters that adequately capture the complex behavior of the system?**
Due to the curse of dimensionality, is critical that the hyper-parameter search-space is kept limited. However, the limitation depends heavily on the coefficient optimization method. In this work, the hyper-parameter search-space was limited due to the slow speed of Nelder-Mead optimization. If DDS coefficient optimization (which is much faster) should only be used, search-space could be increased quite a bit.

R.Q.3 **Do local and global-searches in the coefficient estimation produce significantly different models?**
In general, local-search produces many good models for high computation cost while global-search produces some good models for low computation cost. When best models are compared, Nelder-Mead coefficient optimization achieves wee bit better models than DDS optimization although the difference is very little. However, the DDS search gives solutions with more 'bang for the buck', and increasing its iterations (which is entirely possible due to its speed) could generate better models than Nelder-Mead does.

R.Q.4 **Do different objective function criteria (i.e., calibrating models to single/multi-step forecasts) generate substantially different models?**
Yes, single-step models are pretty robust while multi-step models are quite

prone to overfit. Based on average PI, the two catchments differ in their preference of objective function criteria. Dæmningen selects single-step models while Damhusåen selects multi-step models.

R.Q.5 **Will proposed error metrics result in analogous models.**
No. If models are selected based on the simplified version of ATS activation, the models have poor PI skill-score. However, if models are selected based on PI skill-score, models result in fairly good accuracy. Although PI is quite robust, it is not perfect as it makes models overfit on longer horizons. Because of this overfitting on a longer forecasting horizon, this work proposes to select the model based on average PI skill-score over all forecasting horizons.

R.Q.6 **Does precipitation as an external regressor improve forecasting?**
Yes. All of the top-performing models use precipitation as an external regressor. Because of the correlation between precipitation and runoff, adding external regressors to the models improves their forecasting capability.

R.Q.7 **How do ARIMA models compare to the current models in use?**
The currently used models for the Damhusåen catchment are stochastic grey-box models. These models take uncertainty into account and use actual radar rainfall forecasts for generating runoff forecasts. The models perform well but are quite slow and complex. The ARIMA type models presented in this work are a bit different. The models are calibrated and evaluated on 'perfect' rainfall data, overestimating the real-world performance of the model. However, the models are fast and quite simple, and perform well on the used data.
The accuracy in predicting the simplified ATS activation has been evaluated on the grey-box models for the Damhusåen catchment. The results can be found in the Krüger evaluation report [8]. Comparisons (which should not be taken seriously) of the grey-box models and the ARIMA type models is shown in following table:

Forecasting horizon	Accuracy	Grey-box	ARIMA
30 min	Correct	0.14,	0.64
30 min	Correct + early	0.39	0.66
60 min	Correct	0.09	0.47,
60 min	Correct + early	0.39	0.48

The ARIMA type models presented in this work can be seen to perform well. However, a comparison with actual radar forecast should be done to get a more fair comparison.

CHAPTER 8

Conclusions

The main objectives of this work were automated model selection of ARIMA type models by using meta-optimization. Runoff measurements of two locations in Copenhagen were used: Dæmningen and Damhusåen, and the precipitation data were composed of 'perfect' radar observations. Only data associated with wet-weather was used in the calculation of the objective function.

By defining a search-space of model hyper-parameters and using meta-optimization for the model selection, ARIMA type models can be generated with good performance. Regarding, the meta-optimization, a few things were observed:

(a) Model hyper-parameter search-space must be kept limited depending on the method used for coefficient optimization.

(b) In coefficient optimization, DDS global-search is much faster than Nelder-Mead local-search and can generate models of similar quality

(c) Models that are calibrated over multi-step forecasts are more prone to overfit than models calibrated over single-step forecasts.

(d) Model selection should not be done on accuracy in predicting WET activation. Additionally, model selection based on PI should be done carefully as longer forecasting horizons overfit and poor performance on shorter horizons.

(e) The model selection of single/multi-step models is catchment-based i.e. single-step models are selected for Dæmningen while multi-step models are selected for Damhusåen.

(f) Precipitation as an external regressor improves model performance.

Proposed models are a considerable simplification to the conceptual models that are normally used for hydrological forecasting. Despite that, the models are simple and fast, and are capable of generating quality forecasts.

CHAPTER 9

Outlook

The main focus of this work was to investigate factors that affect the automated model selection of ARIMA type models. As always, there is some room for improvement. First of all, increasing the iterations of DDS coefficient estimation and the model hyper-parameter search-space could produce models of better quality. Additionally, using the accuracy of predicting the simplified version of ATS activation in the objective function could be investigated.
Occasionally, the models need to be recalibrated due to changing factors such as temperature, hydration of soil, etc. Hence, a relevant thing to this work would be to analyze how often the models need to be re-calibrated and how much training data is needed.

It was observed that using 'perfect' rainfall data, ARIMA type models achieved good performance. However, to assess the real-world performance of the ARIMA type models, real 'non-perfect' rainfall radar forecasts should be used. This allows a more fair comparison to the currently used grey-box models at Damhusåen. Additionally, future work could compare models generated in this work to other data-driven models, such as artificial neural networks (ANNs).

Finally, due to the simplification of the conceptual hydrological model, some uncertainty can arise, such as input uncertainty, measurement uncertainty, model structure uncertainty, etc. Future work could focus on accounting for forecast uncertainty, something which has been shown to have a positive effect [21].
Additionally, automated data cleaning and correction of measurements could be something considered moving forward.

APPENDIX A

Data Treatment

A.1 Data Treatment

The precipitation data is achieved with radar measurements every 10 minutes. The original data file has 4 columns. The first two columns correspond to date and time. The second two columns correspond to measurements every 2 minutes and every 10 minutes. At the start of the data file, measurements are taken every 10 minutes and the 2-minute measurement column stays empty. Later on, measurements switch to being obtained every 2 minutes. When the 2-minute measurements are analyzed it can be seen that the values are just data measurements every 10 minutes but repeated five times. In later chapters, it will be described how data will be modified to be consistently acquired.

The runoff measurements in files s1 and s2 have a frequency of 2 minutes. However, data is not always acquired every 2-minutes. There are times in which measurements are not observed and thus, timestamp skipped. To overcome these missing value problems (and to get the same number of data points for precipitation and runoff) these measurements will be mean-aggregated later on such that measurements are consistently gathered every 10 minutes.

A.1.0.1 Common Range

The original range of the data files is shown in Table A.1. It was decided to cut the data files to their common range.

Data file	From	To
Precipitation	2017/07/04 12:30	2020/01/01 00:00
Station 1	2017/08/16 09:28	2019/12/31 23:58
Station 2	2017/04/05 14:58	2019/12/31 23:58
Common range	2017/08/16 10.00	2019/12/31 23:50

Table A.1: The range of the raw data differs so that getting the data on common range is done.

A.1.0.2 Getting Data on 10 Minutes

As we are modeling the data points in a regressional manner, and the two data files have a different frequency, it makes sense to aggregate the data to the same frequency. Thus, the runoff measurements in s1 and s2 are aggregated by taking the mean of the prior 5 measurements to get the flow for each point. I.e. to get the flow at time Q_t, mean of values $Q_{t-1}, Q_{t-2}, Q_{t-3}, Q_{t-4}, Q_{t-5}$ are taken.
This synchronizes the frequency of the data files such that time series modeling with regression terms can be done on it.

A.1.0.3 Daylight Saving

Depending on location, European countries can vary significantly in daylight throughout the year. In Copenhagen Denmark, daylight during summer can stretch up to 18 hours while in the wintertime, daylight drops to around 8 hours. To cope with having less daylight in the wintertime many countries located on the northern hemisphere shift the clock 1 hour ahead during the less bright months and shift it back when daylight is more abundant. Generally, these shifts are usually done last Sunday in March and October at 01:00 UTC. However, the local time of these shifts varies slightly between European countries as they span different time zones.

As we shift the clock one hour ahead in October and then shift if one hour back in the March, values might be omitted or duplicated. For instance, in the precipitation dataset, winter shift values are duplicated and summer shift values omitted. As Figure A.1 shows, this results in the duplication of timestamps from 02:00 to 02:58, and the corresponding timestamps having different values. Forecast data from 02:00 - 02:58 is omitted at the summer shift.
As for the runoff data, no duplicate values were observed during the winter shift. This is likely due to one of the duplicated pair to be omitted (see Figure A.1). To be able to shift the index one hour ahead and filling in the missing summer shift hour, the first shifted hour (03:00-03:58) is simply a duplicate of the preceding hour.
As the data ranges from August 2017 to the year-end of 2019, there are several daylight saving shifts necessary to take account for (see Figure A.2). If any value with a timestamp falling between a summer shift event and the previous year's winter shift, it should be shifted one hour ahead.

Year	Wintershift	Summershift
2017	2017/10/29 02:00	NA
2018	2018/10/28 02:00	2018/03/25 01:50
2019	2019/10/27 02:00	2019/03/31 01:50
2020	2020/10/25 02:00	2020/03/29 01:50

Table A.2: Daylight saving shifts

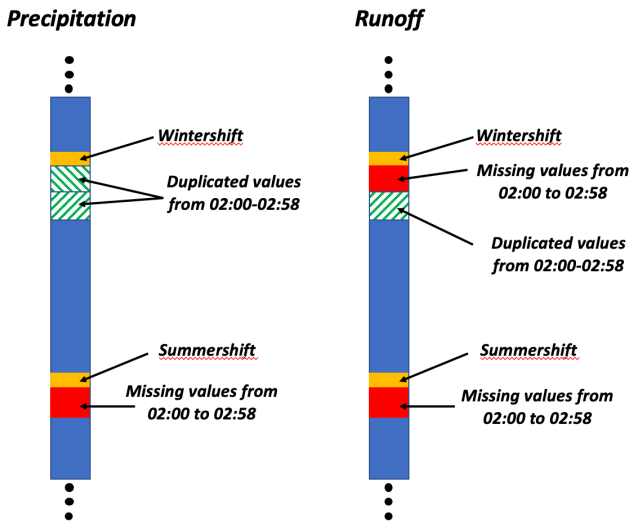


Figure A.1: Data structure before fixing daylight saving shifts.

APPENDIX B
Additional Figures

B.1 Optimization Methods

B.1.1 Computing Time

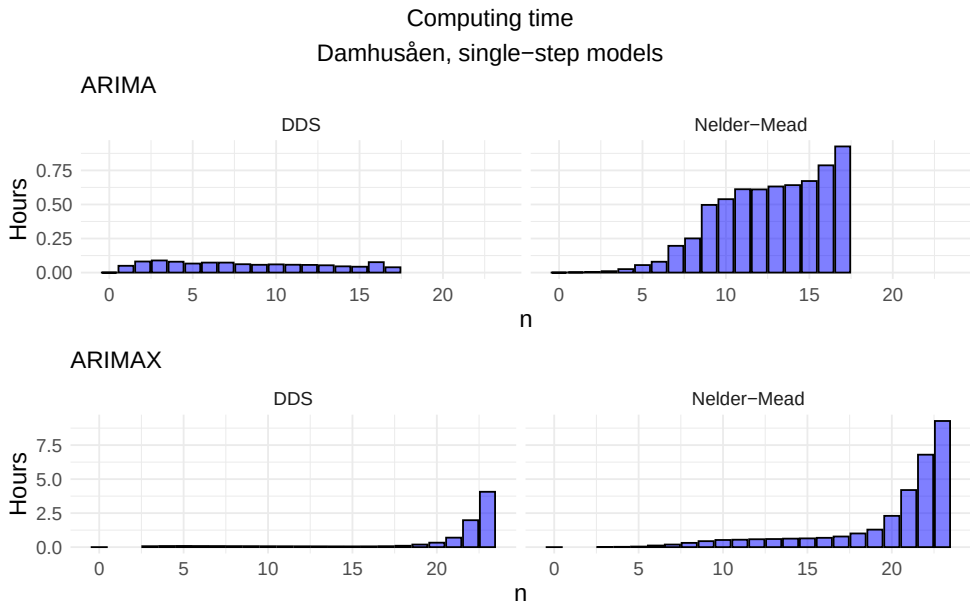


Figure B.1: Computing cost for single-step models on Damhusåen.

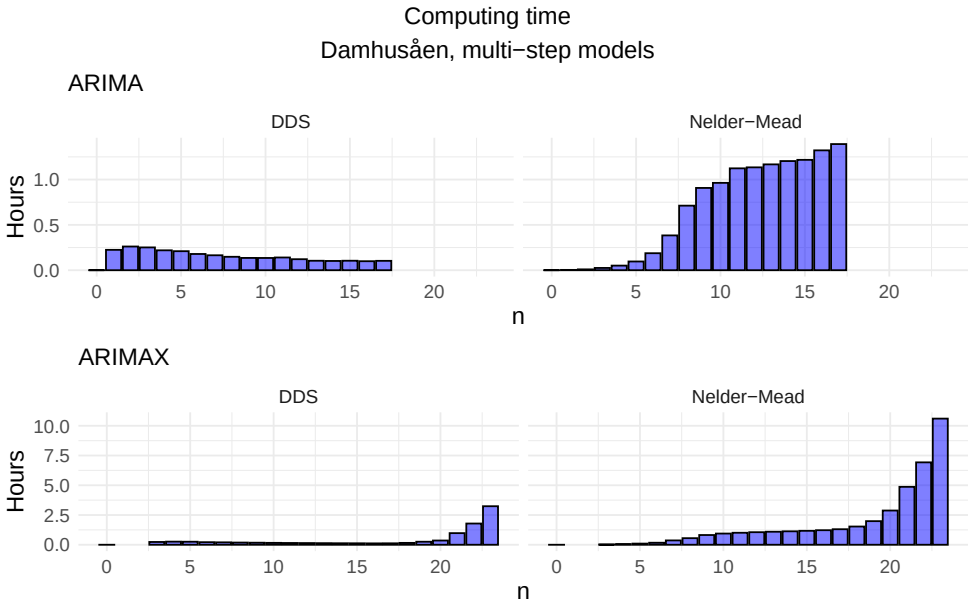


Figure B.2: Computing cost for multi-step models on Damhusåen.

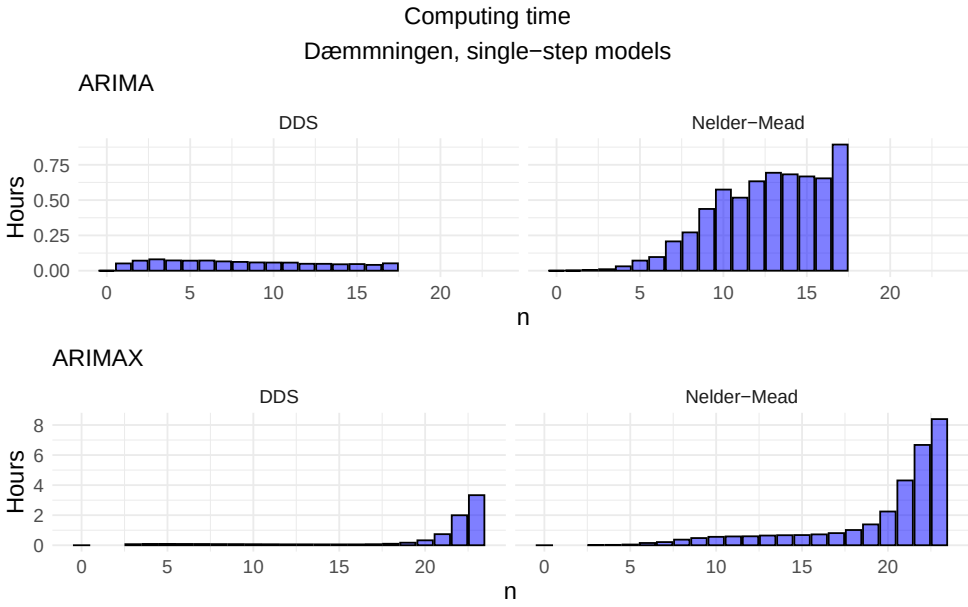


Figure B.3: Computing cost for single-steo models on Dæmningen.

B.1.2 Minimizing Objective Function

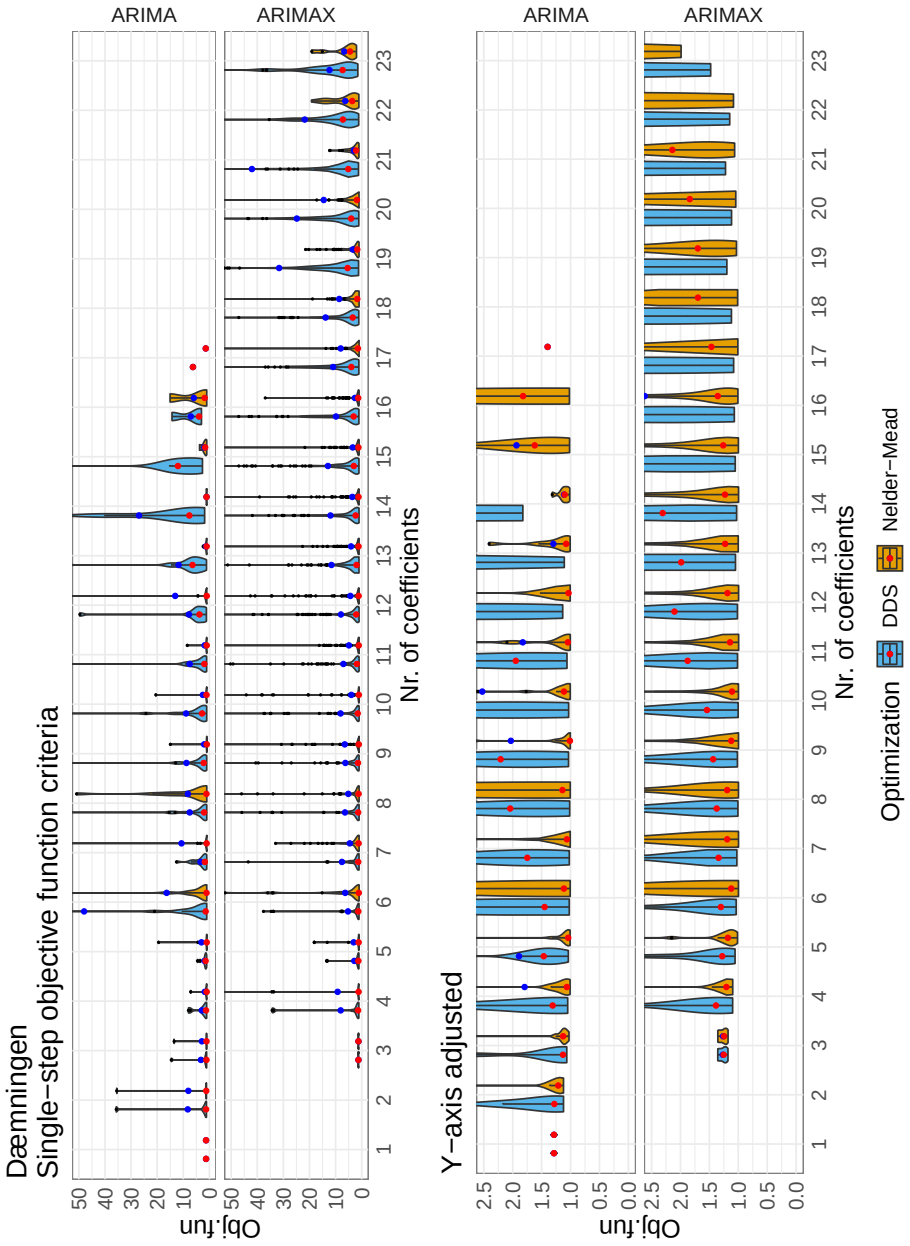


Figure B.4: Nelder-mead is more sucessfull in achiveing low objective function value. Blue dot is mean, red dot is median.

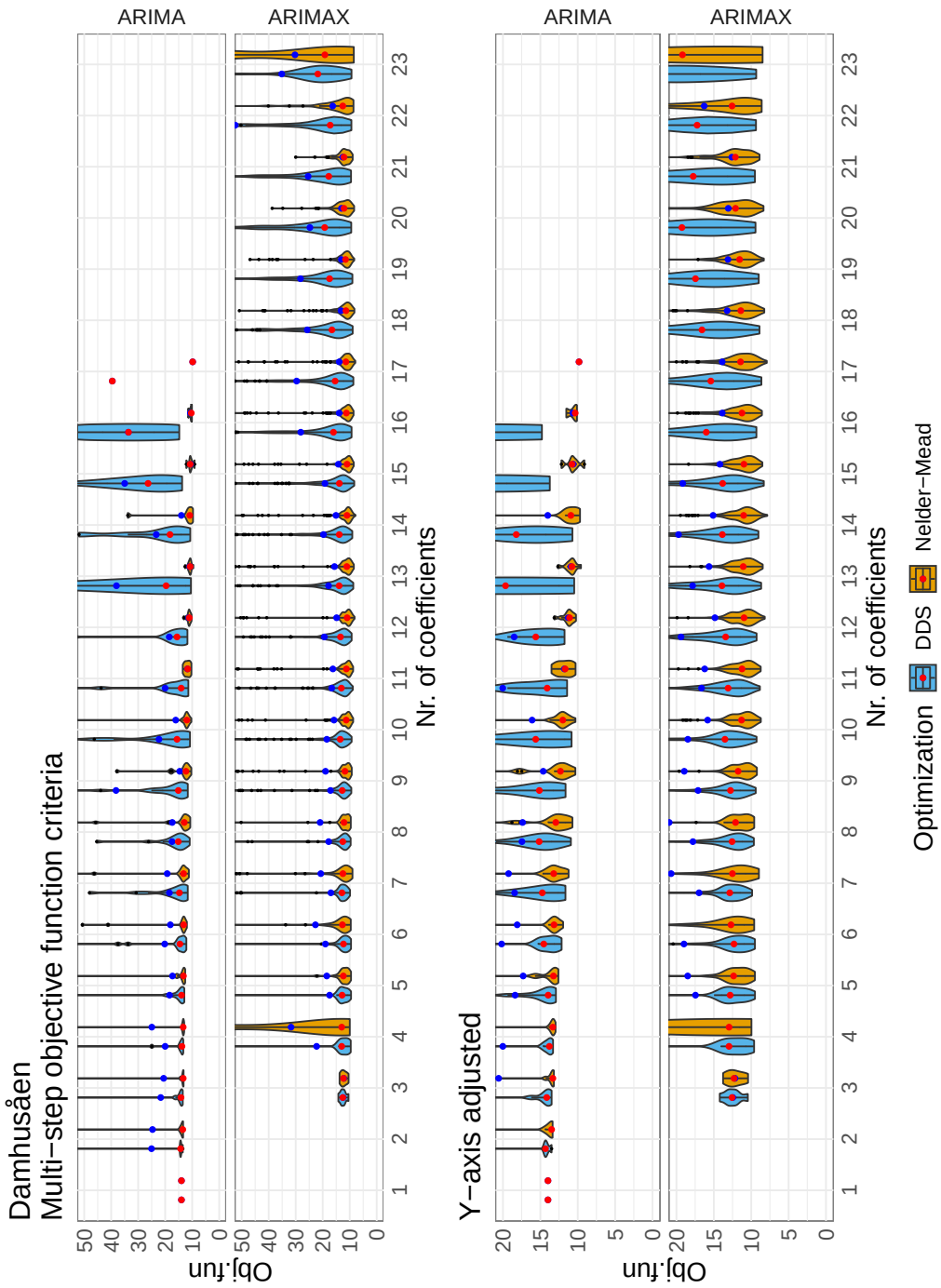


Figure B.5: Nelder-mead is more sucessfull in achiveing low objective function value. Scoring criterium for multi-step objective function criteria is found in Equation 5.2. Blue dot is mean, red dot is median.

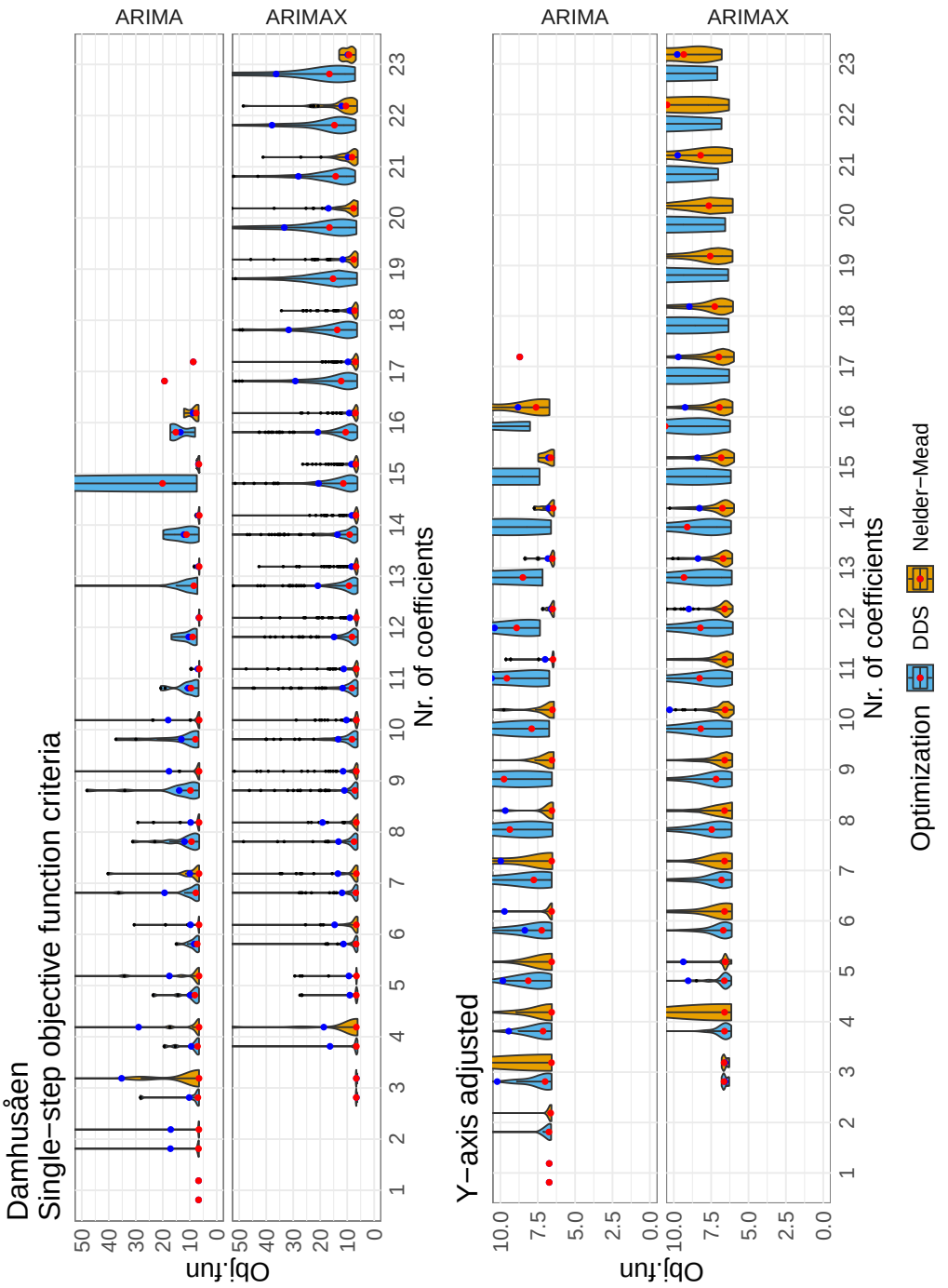


Figure B.6: Nelder-mead is more sucessfull in achiveing low objective function value. Scoring criterium for multi-step objective function criteria is found in Equation 5.2. Blue dot is mean, red dot is median.

B.2 Comparison of DDS optimized hyper-models

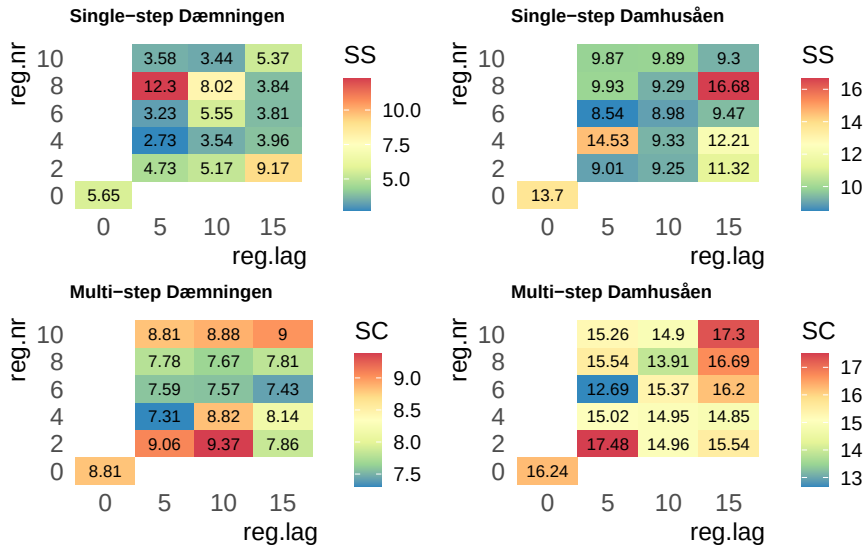


Figure B.7: Average minimization of the objective function with DDS search associated with lags and number of external regressor. Objective function is sum-of-squares for single-step models and SC (Equation 5.2) for multi-step models.

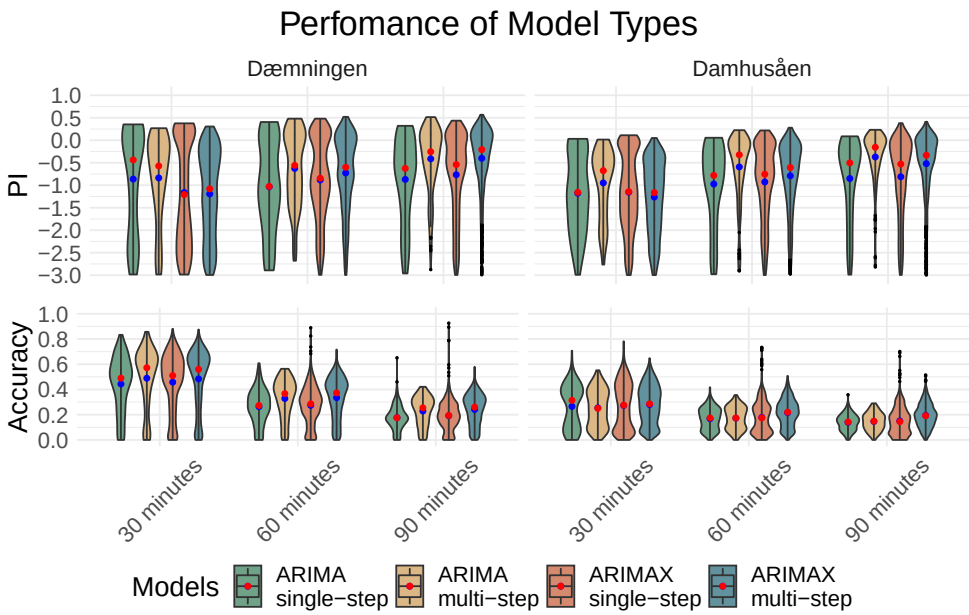


Figure B.8: Multi-step models generally have higher PI skill-score while accuracy seems to be higher for single-step models.

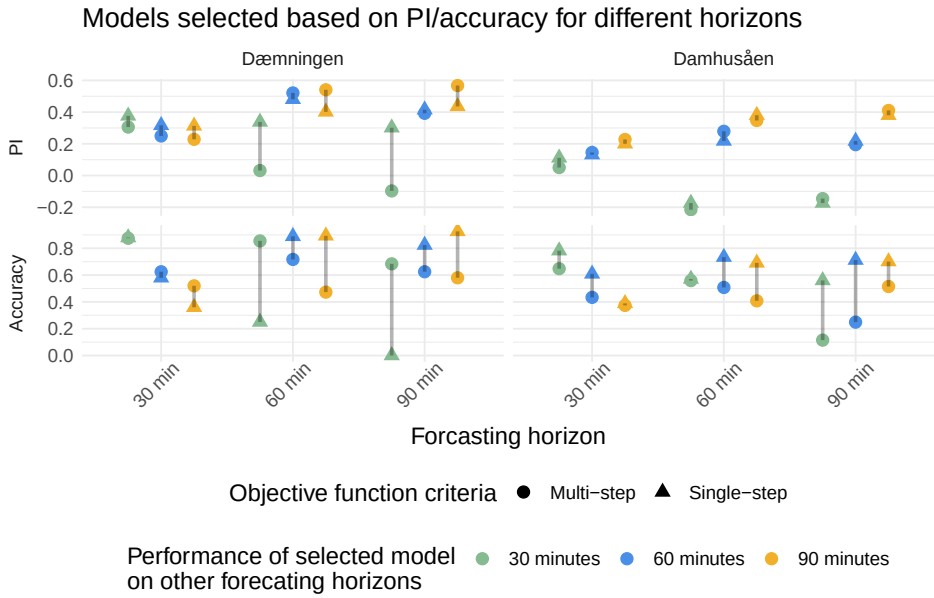


Figure B.9: The best performing models for differenct forcasting horizon and objective function criteria. Multi-step models are more overfitted than the single-step models, especially for longer forecasting horizons.

B.3 Real-World Forecasting Models

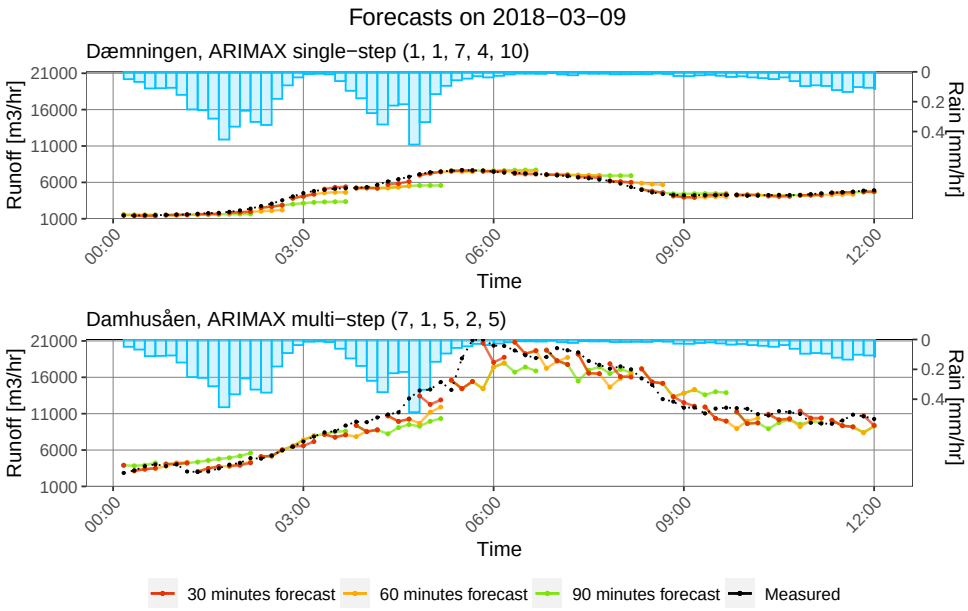


Figure B.10: Real-world forecasts of model number 1 for both catchments from Table 6.3

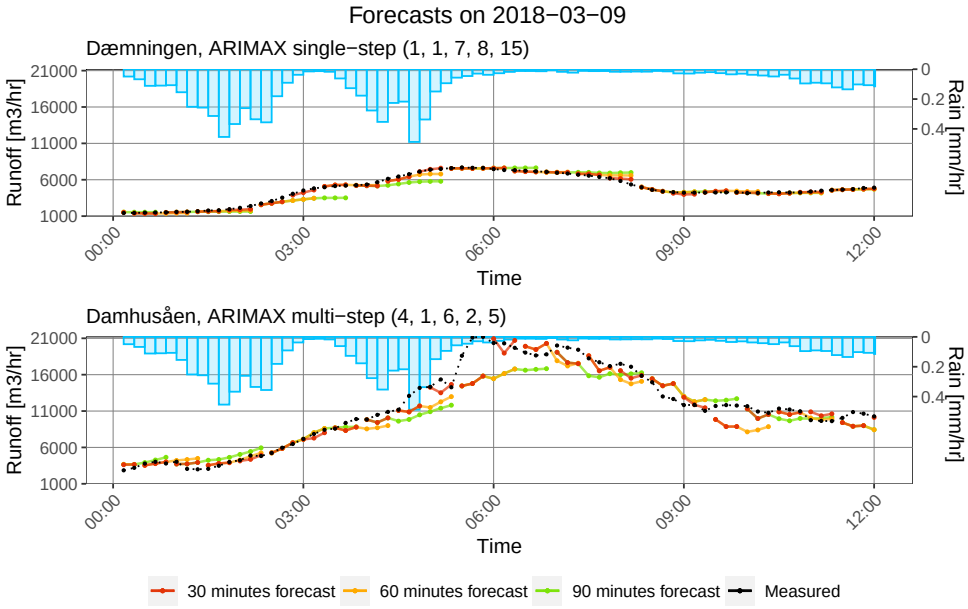


Figure B.11: Real-world forecasts of model number 2 for both catchments from Table 6.3

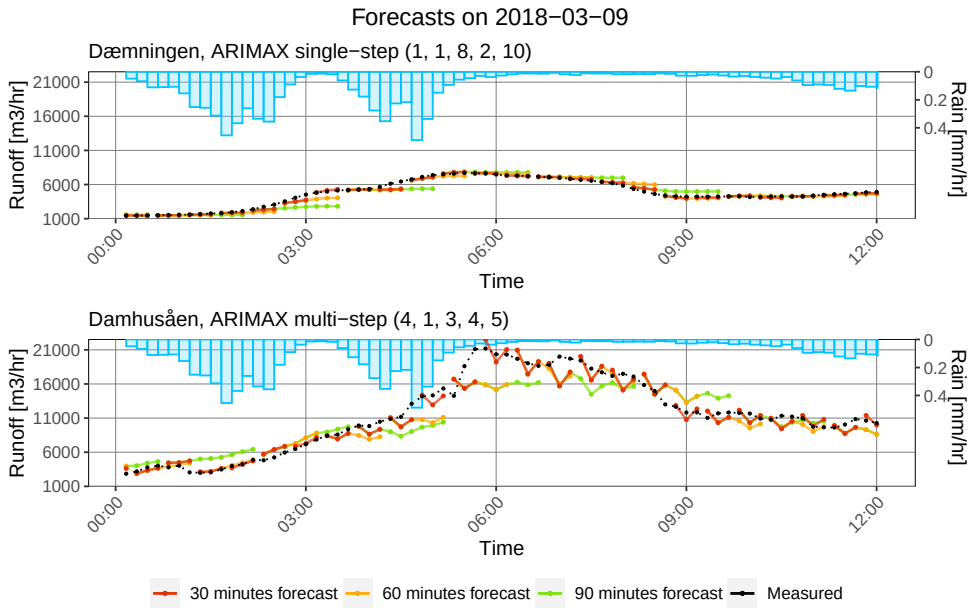


Figure B.12: Real-world forecasts of model number 3 for both catchments from Table 6.3

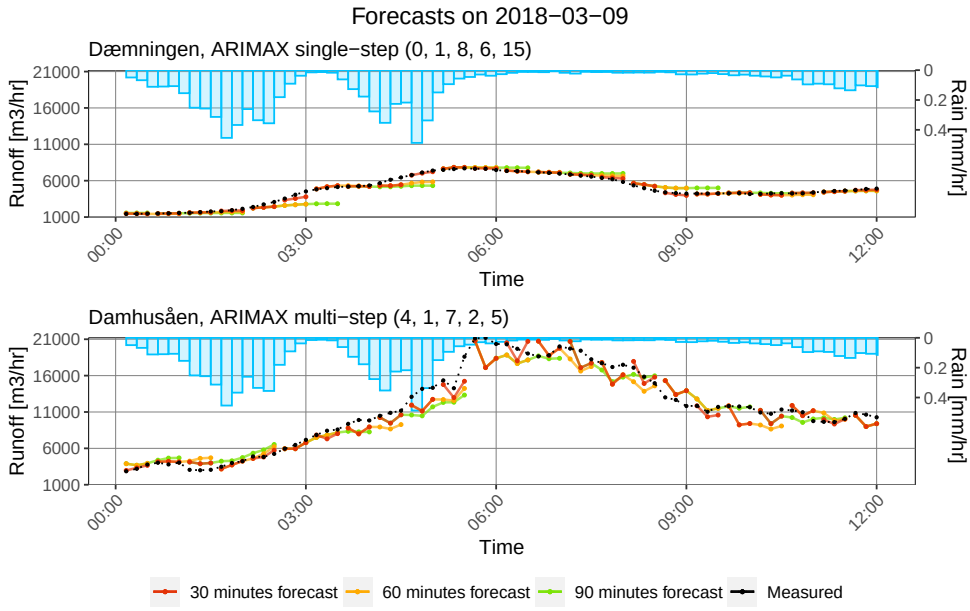


Figure B.13: Real-world forecasts of model number 4 for both catchments from Table 6.3

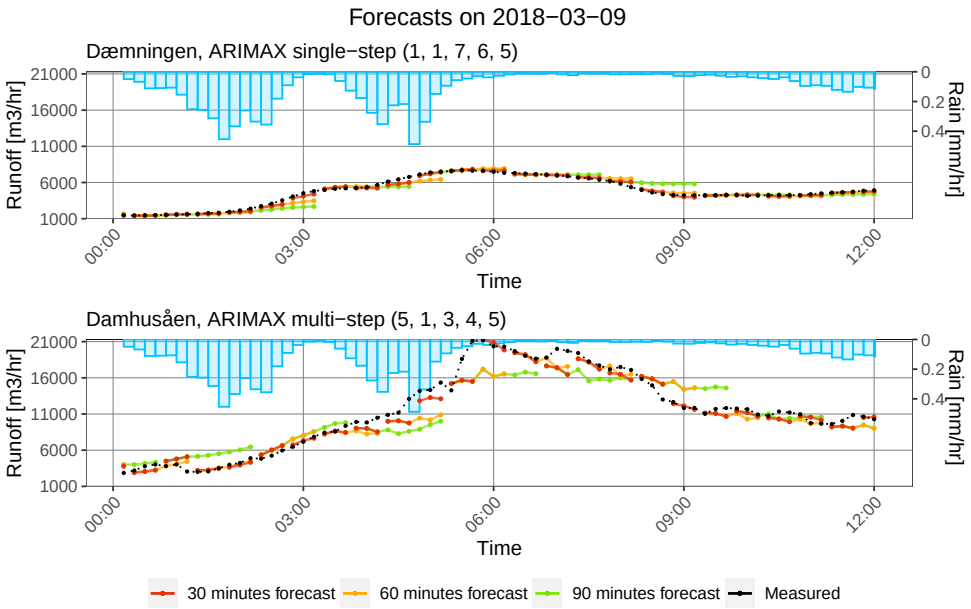


Figure B.14: Real-world forecasts of model number 5 for both catchments from Table 6.3

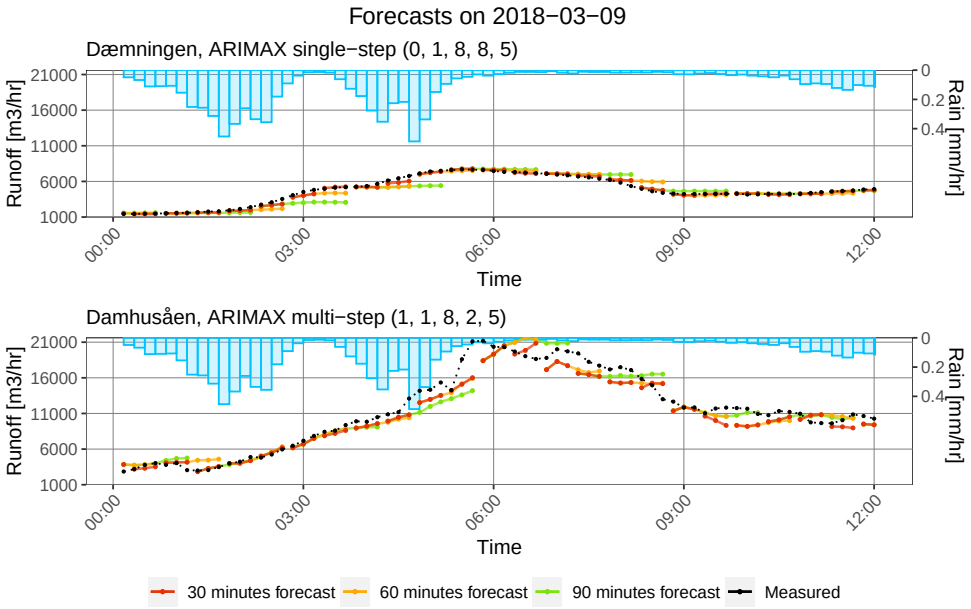


Figure B.15: Real-world forecasts of model number 6 for both catchments from Table 6.3

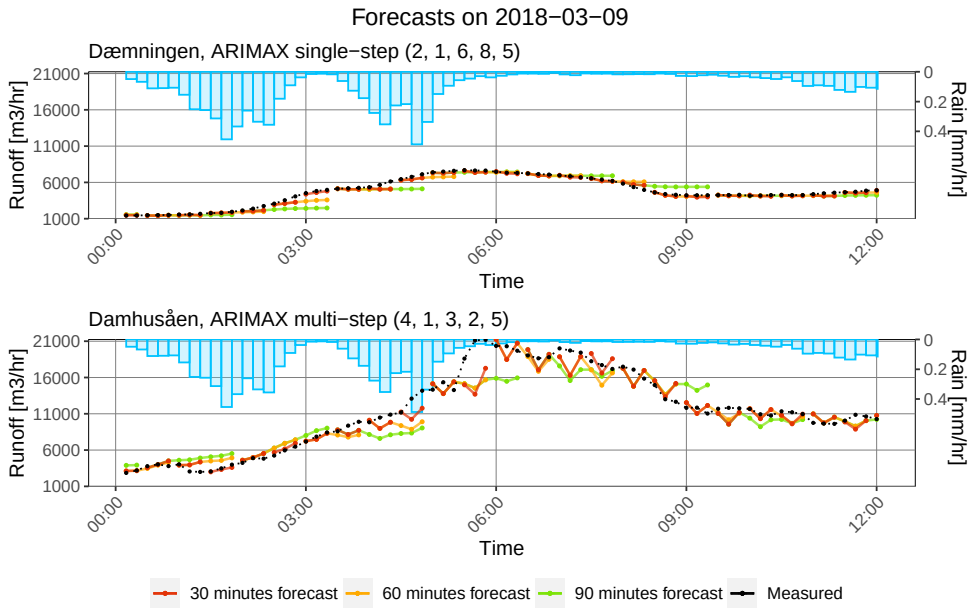


Figure B.16: Real-world forecasts of model number 7 for both catchments from Table 6.3

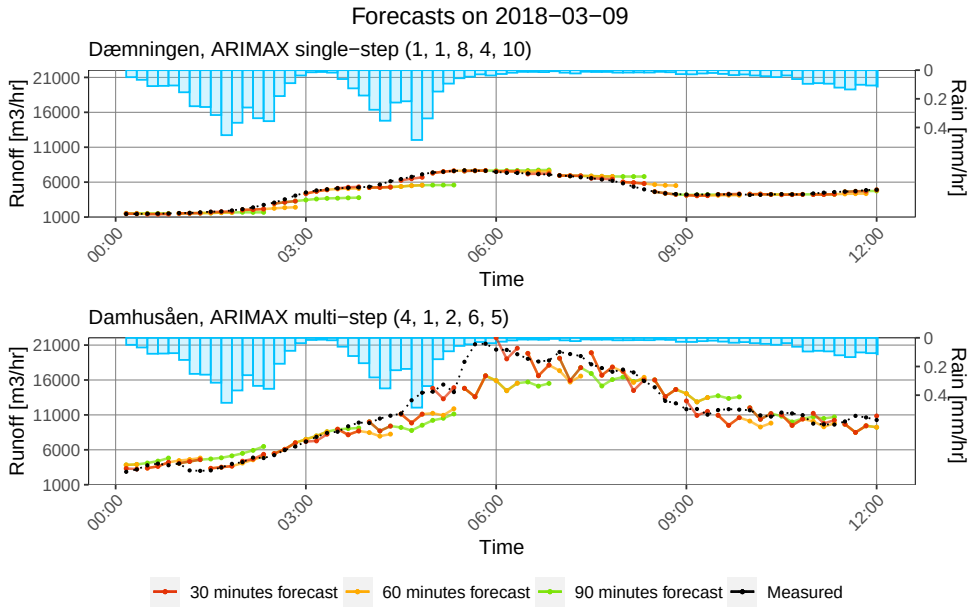


Figure B.17: Real-world forecasts of model number 8 for both catchments from Table 6.3

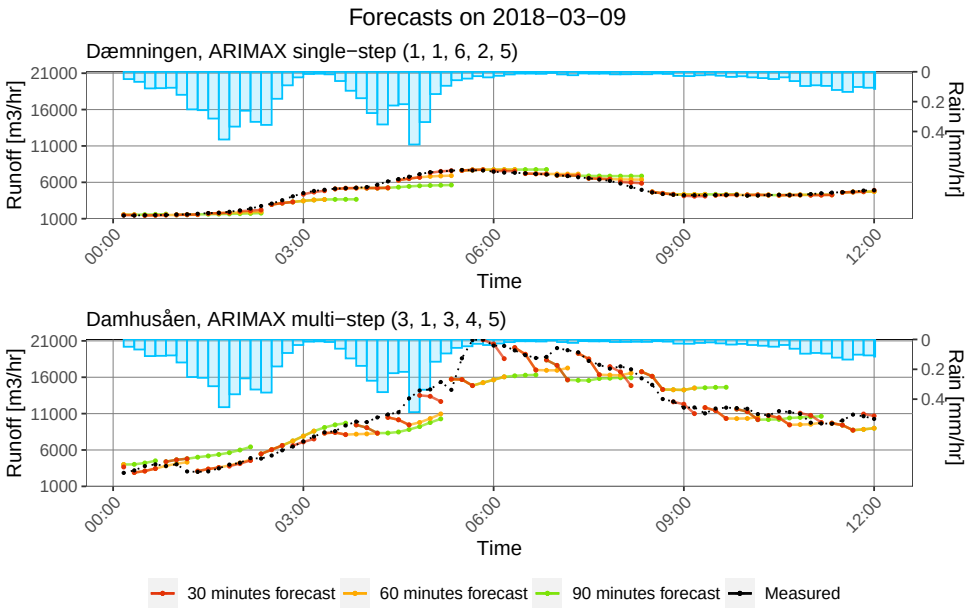


Figure B.18: Real-world forecasts of model number 9 for both catchments from Table 6.3

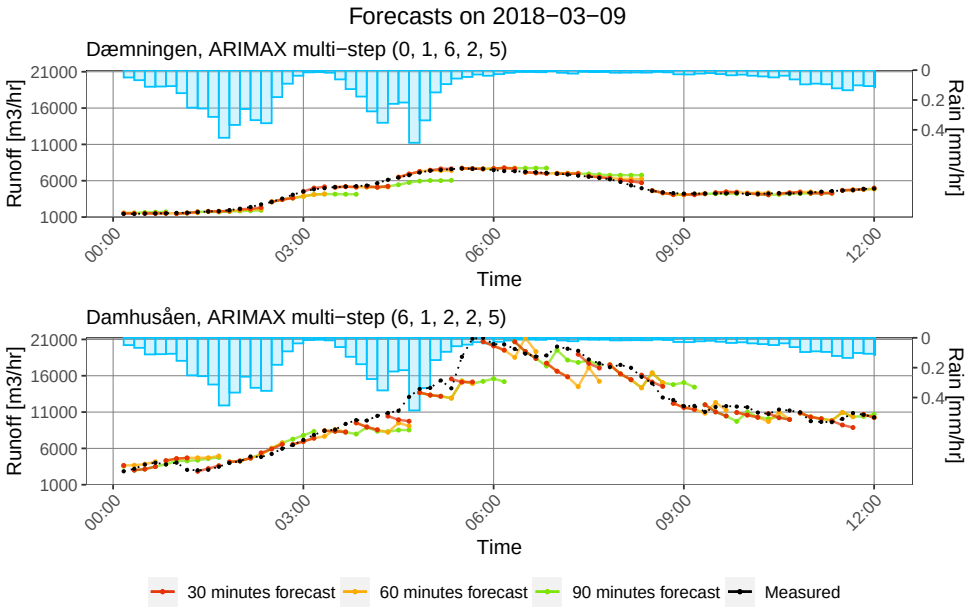


Figure B.19: Real-world forecasts of model number 10 for both catchments from Table 6.3

Bibliography

[1] Allan H. Murphy. "Skill Scores Based on the Mean Square Error and Their Relationships to the Correlation Coefficient". In: *Monthly Weather Review* 116.December 1988 (Feb. 1099), pp. 2417–2424. DOI: https://doi.org/10.1175/1520-0493(1988)116<2417:SSBOTM>2.0.CO;2. URL: https://journals.ametsoc.org/doi/pdf/10.1175/1520-0493%281988%29116%3C2417%3ASSBOTM%3E2.0.CO%3B2.

[2] Anders Breinholt and Anitha K. Sharma. *Case area baseline report: Copenhagen and Lynette Fællesskabet*. Mar. 2010.

[3] Bellis, Mary. *The History of Plumbing*. Feb. 2020. URL: https://www.thoughtco.com/history-of-plumbing-1992310.

[4] Bryan A. Tolson and Christine A. Shoemaker. "Dynamically dimensioned search algorithm for computationally efficient watershed model calibration". In: *Water Resources Research* 43.W01413 (Jan. 2017). DOI: 10.1029/2005WR004723.

[5] ByoungSeon Choi. *ARM A Model Identification*. 1st. Springer, 1992. ISBN: ISBN-13: 978-1-4613-9747-2.

[6] Duo Zhang et al. "Hydraulic modeling and deep learning based flow forecasting for optimizing inter catchment wastewater transfer". In: *Journal of Hydrology* 567.2018 (Nov. 2017), pp. 792–802. DOI: https://doi.org/10.1016/j.jhydrol.2017.11.029.

[7] Emili Balaguer et al. "Predicting service request in support centers based on nonlinear dynamics, ARMA modeling and neural networks". In: *Expert Systems with Applications* 34.2008 (2006), pp. 665–672. DOI: doi:10.1016/j.eswa.2006.10.003.

[8] *Evaluation of KRG2+4*. Tech. rep.

[9] Fong-Lin Chu. "Forecasting tourism: a combined approach". In: *Tourism Management* 19.6 (), pp. 515–520.

[10] Gavin Boyd, Dain Na, and Zhong Li. "Influent Forecasting for Wastewater Treatment Plants in North America". In: *Sustainability* 2019.11, 1764 (Mar. 2019). DOI: 10.3390/su11061764.

[11] George E. P. Box et al. *Time Series Analysis: Forecasting and Control, 5th Edition*. 5th edition. Wiley, June 2015. ISBN: ISBN: 978-1-118-67502-1.

[12] H. Jonsdottir, H. Aa Nielsen, and H. Madsen. "Conditional parametric models for storm sewer runoff". In: *Water Resources Research* 43.W05443 (May 2007). DOI: 10.1029/2005WR004500.

[13] Henrik Bechmann et al. "Grey-box modelling of aeration tank settling". In: *Water Research* 36 (Aug. 2001), pp. 1887–1895.

[14] Jacob Carstensen, Marinus K. Nielsen, and Helle Strandæk. "Prediction of hydraulic load for urban storm control of a municipal WWT plant". In: *Wat. Sci. Tech* 37.12 (1998), pp. 363–370.

[15] Jan G De Gooijer and Rob J Hyndman. *25 Years of Time Series Forecasting*. Jan. 2006. URL: https://robjhyndman.com/papers/ijf25.pdf.

[16] Jonas Kjeld Kirstein et al. "A semi-automated approach to validation and error diagnostics of water network data". In: *Urban Water Journal* 16.1 (Apr. 2019), pp. 1–10. DOI: 10.1080/1573062X.2019.1611884. URL: https://doi.org/10.1080/1573062X.2019.1611884.

[17] Jonathan D. Cryer and Kung-Sik Chan. *Time Series Analysis with Applications in R*. 2nd. Springer.

[18] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. 2nd edition. Springer, 2006. ISBN: ISBN-13: 978-0387-30303-1.

[19] J.R. Kim et al. "Forecasting influent flow rate and composition with occasional data for supervisory management system by time series model". In: *Water Science & Technology* 53.4-5 (), pp. 185–192. DOI: doi:10.2166/wst.2006.123.

[20] Lisbet Sneftrup Hansen and Morten Borup. "Flow Forecasting using Deterministic Updating of Water Levels in Distributed Hydrodynamic Urban Drainage Models". In: *Water* 2014,6, 2195-2211 (July 2014), pp. 2195–2211. DOI: 10.3390/w6082195.

[21] Roland Löwe, Luca Vezzaro, and Peter Steen Mikkelsen. "Probabilistic runoff volume forecasting in risk-based optimization for RTC of urban drainage systems". In: *Environmental Modelling & Software* 80 (Feb. 2016), pp. 143–158.

[22] Luca Vezzaro and Morten Grum. *A generalized Dynamic Overflow Risk Assessment (DORA) for urban drainage RTC*. 2012.

[23] M.K. Nielsen, H. Bechmann, and M. Henze. "Modelling and test of aeration tank settling (ATS)". In: *Water Science and Technology* 41.9 (2000), pp. 179–184. URL: https://iwaponline.com/wst/article-pdf/41/9/179/427819/179.pdf.

[24] Mohammad Valipour, Mohammad Ebrahim Banihabib, and Seyyed Mahmood Reza Behbahani. "Comparison of the ARMA, ARIMA, and the autoregressive artificial neural network models in forecasting the monthly inflow of Dez dam reservoir". In: *Journal of Hydrology* 476.2013 (June 2012), pp. 433–441. DOI: http://dx.doi.org/10.1016/j.jhydrol.2012.11.017.

[25] S. R. Mounce et al. “Predicting combined sewer overflows chamber depth using artificial neural networks with rainfall radar data”. In: *Water Science & Technology* 2014.69.6 (), pp. 1326–1333. DOI: doi:10.2166/wst.2014.024.

[26] Neil D. Bennett et al. “Characterising performance of environmental models”. In: *Environmental Modelling & Software* 40.2013 (Sept. 2012), pp. 1–20. DOI: http://dx.doi.org/10.1016/j.envsoft.2012.09.011.

[27] *Radar based 24-hr Precipitation from Tropical Storm Fay*. URL: http://kejian1.cmatc.cn/vod/comet/hydro/flash_flood/navmenu.php_tab_1_page_4.2.2.htm.

[28] Rob J Hyndman and George Athanasopoulos. *Forecasting: Principles and Practice*. 2nd edition. OTexts: Melbourne, Australia, Feb. 2020. URL: https://otexts.com/fpp2/.

[29] Rob J. Hyndman and Yeasmin Khandakar. “Automatic Time Series Forecasting: The forecast Package for R”. In: *Journal of Statistical Software* 27.3 (July 2008).

[30] Robert H. Shumway and David S. Stoffler. *Time Series Analysis and Its Applications*. 4th. Springer, 2016.

[31] Roland Löwe. “Probbabilistic Forecasting for On-line Operation of Urban Drainage Systems”. PhD thesis. DTU, 2014.

[32] Roland Löwe, Peter Steen Mikkelsen, and Henrik Madsen. “Stochastic rainfall-runoff forecasting: parameter estimation, multi-step prediction, and evaluation of overflow risk”. In: *Stochastic Environmental Research and Risk Assessment* 28.3 (July 2014), pp. 505–5016. ISSN: 1436-3240. DOI: DOI10.1007/s00477-013-0768-0.

[33] S. Thorndahl and M. R. Rasmussen. “Short-term forecasting of urban storm water runoff in real-time using extrapolated radar rainfall data”. In: *Journal of Hydroinformatics* 15.3.2013 (2013), pp. 897–912.

[34] Søren Thorndahl et al. “Weather radar rainfall data in urban hydrology”. In: *Hydrology and Earth System Sciences* 2017.21 (Mar. 2017), pp. 1359–1380. DOI: 10.5194/hess-21-1359-2017.

[35] Tetiana Stadnytska, Simone Braun, and Joachim Werner. “Comparison of automated procedures for ARMA model identification”. In: *Behaviour Research Methods* 2008.40 (), pp. 250–262. DOI: dol:10.3738/BRM.40.I.250.

[36] Wayne F. Velicer and John Harrop. “The Reliability and Accuracy of Time Series Model Identification”. In: *Evaluation Review* 7.4 (Aug. 1983), pp. 551–560.

[37] Zahra Alizadeh et al. “Assessment of Machine Learning Techniques for Monthly Flow Prediction”. In: *Water* 2018.10 (Nov. 2018). DOI: doi:10.3390/w10111676.

{2.7182818284}ϑφερτυθιοπσδφγηξκλ