Universiteit Leiden

Master Computer Science

AutoML for Hydraulic Head Forecasting in Dikes:
A Selective Pooling and Peak-Aware Approach

Name:	Bram van Eerden
Student ID:	s3726991
Date:	25/09/2025
Specialisation:	Data Science
1st supervisor:	Mitra Baratchi
2nd supervisor:	Jan van Rijn

Master's Thesis in Computer Science

Leiden Institute of Advanced Computer Science (LIACS)
Leiden University
Niels Bohrweg 1
2333 CA Leiden
The Netherlands

Abstract

Hydraulic head forecasting in dikes is essential for calculating the probability of failure of a flood defense mechanism, as sudden peaks or prolonged extreme values in groundwater levels could induce direct or indirect processes that may compromise stability. Forecasting these time series in a short-horizon setting is challenging due to nonlinear hydrological dynamics and site-specific subsurface heterogeneity. Automated Machine Learning (AutoML) offers a way to automate algorithm and hyperparameter selection, but its potential for hydraulic head forecasting in dikes has not yet been systematically investigated. In this thesis, we evaluate AutoGluon-TimeSeries (AG-TS) against classical models and alternative AutoML frameworks across three settings: local univariate, global univariate, and global multivariate. To improve global models, we propose Bayesian Optimization for Selective Pooling (BOSP), which adaptively identifies informative subsets of series, and we extend the framework with a peak-aware covariate augmentation (BOSP+Peak) designed to improve performance during hydraulic head peaks. Using over four years of data from 51 piezometers across 10 Dutch dikes, we show that AG-TS outperforms baselines in the local setting reducing the average error by 7%, achieving the lowest error on 67% of all time series. BOSP achieves significant improvements in global univariate forecasting by reducing the average error by 16% and improving upon the baseline in 90% of all time series. BOSP+Peak reduces the average error during peak events by 18% and outperforms the baseline in 75% of all cases while preserving overall performance outside peak periods. Together, these results demonstrate that AutoML, when combined with domain-specific extensions, provides a scalable and effective approach to hydraulic head forecasting in dikes, with clear potential for risk assessments of dikes in an operational setting.

Acknowledgements

I would like to thank Dr Mitra Baratchi and Dr Jan van Rijn for their time investment, feedback and insightful discussions throughout the project, which helped shape this research and bring it to a successful conclusion.

I would also like to extend my gratitude to Erik Vastenburg for providing the initial idea for this thesis, and with him all colleagues at HHNK for their guidance, support and sharing their expertise during this project.

Finally, I would like to thank my family, friends and partner for their interest in my work and support throughout this project.

Contents

Chapter 1

Introduction

Despite efforts by countries worldwide to limit global warming to 1.5–2.0 °C, sea levels keep rising and longer periods of extreme weather occur more frequently. These changing conditions create an environment where flooding is expected to be one of the most damaging consequences of climate change [1]. In the Netherlands, flooding is not only a threat from the sea, but also a significant threat further inland, as 26% of the country's land mass lies below sea level and an estimated 29% is at risk of flooding [2]. This risk is particularly high in polders, which are low-lying areas surrounded by dikes and equipped with internal drainage systems [3]. These polders are dependent on the dikes surrounding them for protection against flooding. The Netherlands maintains over 17.700 km of dikes in total [4].

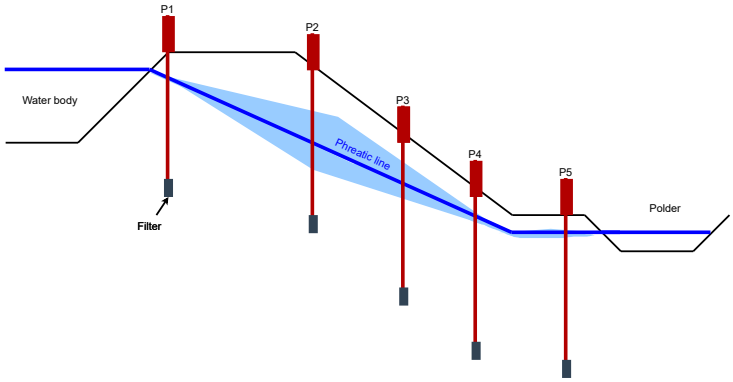


FIGURE 1.1: Schematization of a cross-sectional profile of a dike. P1-P5 are the piezometers, which measure the hydraulic head at their locations, allowing an approximation of the phreatic line. Illustration by the author.

An important indicator for dike safety is dike stability. Dike stability is mainly impacted by changes in water pressures - the hydraulic head changes - impacting the subsurface balance. High groundwater levels in dikes could induce direct or indirect processes that may compromise dike stability, causing a breach in the worst-case scenario. The hydraulic head (groundwater level) within the dike is measured by piezometers placed at multiple depths and locations. From these measurements, the phreatic line (the highest internal water level) can be inferred [5]. Sudden rises, anomalous patterns, or prolonged extremes in hydraulic head may indicate damaging processes such as slope instability or internal erosion, which could cause shear

of the dike body, and are therefore important indicators [6]. Modern monitoring systems collect daily time series across many sites; the key challenge is turning this data into short-horizon forecasts that support operational decisions such as targeted inspection and taking precautionary emergency measures [7].

Forecasting hydraulic head levels is difficult because of the complex temporal patterns inherent to these time series, influenced by precipitation, evaporation, seasonal variation, and upward seepage from deeper soil layers [8]. Numerous studies have examined forecasting methods for hydraulic head data, albeit primarily in wells [9], [10]. While the literature on wells provides important insights into the domain of hydraulic head forecasting, dikes differ in boundary conditions, layered structure, and responsiveness to rainfall. As hydraulic head time series from dikes are collected using a comparable setup (several piezometers in a cross-sectional profile of the dike), their time series may contain informative patterns beyond the target series itself. Some of these patterns are direct, such as multiple piezometers at the same site responding similarly to rainfall, while others are more subtle, such as comparable recharge dynamics observed in dike sections in other geographical locations. An effective forecasting approach should therefore be able to capture site-specific behavior while also learning from related series. This motivates evaluating models trained only on the target series alongside models trained on multiple series In this work, we methodologically distinguish two modeling dimensions: First, *local* models are trained per site, while *global* models share parameters across sites. Second, *univariate* indicates a single target series, whereas *multivariate* denotes joint prediction of multiple targets [11]. Local models can suffer from limited per-site data; global models exploit cross-site regularities but risk negative transfer when sites behave differently.

Beyond the local–global and univariate-multivariate distinctions in modeling approach, selecting an appropriate forecasting algorithm type remains a challenge as well. Many approaches are available for time series forecasting, ranging from traditional methods such as ARIMA and exponential smoothing to more recent techniques including Long Short-Term Memory (LSTM) networks, Convolutional Neural Networks (CNNs), and Transformers [12]. Most studies on hydraulic head forecasting, however, focus on a single model class, often a neural network [13], [14], and manually tune its architecture and hyperparameters for a specific dataset. While this can result in good performance within that study, the results are mainly relevant to the chosen model and optimization procedure. As a consequence, the literature has produced a wide variety of potentially strong but highly heterogeneous solutions. The solutions found are difficult to compare across studies and rarely generalize to other sites or datasets, limiting their practical transferability [15]. Selecting an appropriate forecasting method therefore continues to require substantial domain expertise and effort, with no guarantee that a model optimized in one setting will be effective elsewhere.

Automated Machine Learning (AutoML) offers a way to automate algorithm and hyperparameter selection, and can potentially find high-performing configurations more efficiently. By automating model selection and hyperparameter optimization , an AutoML framework could provide a unified and generalizable approach to forecasting hydraulic head time series across different sites and applications. Among currently available frameworks, AutoGluon-TimeSeries (AG-TS) has recently been shown to be state of the art on a large variety of domains [16], [17]. AG-TS shifts the AutoML focus away from extensive hyperparameter optimization (HPO), and instead relies on robust default configurations and strong model ensembling. While

AutoML has thus far shown promising results in other domains like finance [18], production engineering [19] and energy demand [20], its effectiveness for hydraulic head forecasting in dikes has not been systematically evaluated. Additionally, high risk in dike stability comes from sharp *peaks* in the hydraulic head of a dike, which are under-represented in training data and therefore under-predicted by standard models.

In this thesis, we investigate the use of AutoML for short-horizon forecasting in hydraulic head time series in dikes. We evaluate local univariate, global univariate, and global multivariate strategies. We introduce a Bayesian-optimization method to select informative subsets of series for global modeling and we develop a peak-aware augmentation method that supplies future peak indicators to improve performance during surge events.

Our main contributions are as follows:

- We evaluate AG-TS on hydraulic head time series under three modeling scenarios: local univariate, global univariate, and global multivariate.
- For global univariate modeling, we introduce a Bayesian optimization approach to automatically select related time series for training.
- We propose a method to improve forecasting of sudden peaks by forecasting peak probability and including these probabilities as future covariates.

The remainder of this thesis is organized as follows. Chapter 2 introduces the fundamental definitions and formalizes the forecasting problem studied in this work. Chapter 3 reviews related literature on time series forecasting, hydraulic head data and AutoML. Chapter 4 presents the proposed methods and their implementation details. The experimental setup and dataset are presented in Chapter 5, and the results are reported in Chapter 6. Finally, Chapter 7 concludes the thesis and outlines directions for future research.

Chapter 2

Problem statement

This chapter will introduce concepts and definitions used in this work. We first describe preliminary concepts and notation, before formalising the forecasting problem in the context of hydraulic head data.

2.1 Preliminaries

A *time series* is a sequence of data points, or observations, collected at consecutive and uniform time intervals. We consider real-valued time series with a constant, daily sampling rate. We denote the time series consisting of hydraulic head measurements from piezometer i as $\mathbf{z}_i = [z_{i,1}, \ldots, z_{i,\ell_i}] \in \mathbb{R}^{\ell_i}$ where ℓ_i denotes the length of that series. Each site contains multiple piezometers, indexed by i , placed on a different location of the dike in a cross-sectional profile. Exogenous covariates (e.g., precipitation, evaporation) are collected as $\mathbf{X}$ with the convention that $\mathbf{X}_{\cdot,t+1:t+H}$ are *known in advance* at forecast origin t, where we assume perfect foresight for evaluation.

A forecasting model f at origin t maps past targets and covariates to future targets over horizon H: $f : (\mathbf{Z}_{S,1:t}, \mathbf{X}_{S,1:t+H}) \mapsto \hat{\mathbf{z}}_{\cdot,t+1:t+H}$, where $S_i \subseteq \{1, ..., N\}$ specifies the modeling context. We distinguish: (i) *local univariate* ($S = \{i\}$, one model per piezometer), (ii) *global univariate* ($S_i \subseteq \{1, ..., N\}$, a single model trained across S_i but predicting one target i at a time), and (iii) *global multivariate* S_i contains all piezometers at the same site, and forecasts are produced jointly for every series in S_i.

The forecasting function f is obtained from a *learning algorithm* A, trained on historical observations D_{train} and evaluated on unseen data D_{test}. For a given time series $z_{i,1:\ell}$, we define D_{test} as the actual values in the forecast horizon(s) of length H. With n backtesting windows, the evaluation set consists of the final $p = n \cdot H$ points of the series, i.e., $D_{\text{test}} = z_{i,\ell-p+1:\ell}$. The training set is the preceding part, $D_{\text{train}} = z_{i,1:\ell-p}$, used by A to fit the forecasting map. Performance is quantified by a loss function $L(A, D_{\text{train}}, D_{\text{test}})$, which measures the discrepancy between predictions and actual observations.

A forecasting function can be built from many algorithms $A^{(j)} \in \mathcal{A}$, with each algorithm adapting internal parameters during training on D_{train}. In contrast, hyperparameters $\lambda \in \Lambda^{(j)}$, such as model depth or regularization strength, must be set externally and have a large impact on the performance of the algorithm. The joint task of choosing both the algorithm and its hyperparameters is known as *combined*

algorithm selection and hyperparameter optimization (CASH):

$$A^*_{\lambda^*} \in \arg \min_{A^{(j)} \in \mathcal{A}, \lambda \in \Lambda^{(j)}} L(A^{(j)}_{\lambda}, D_{\text{train}}, D_{\text{val}}) \tag{2.1}$$

with D_{val} being a subset of D_{train} as designated validation set that is inaccessible to the model during training, and providing as an estimate of the performance of the algorithm. AutoML frameworks automate this process.

The *hydraulic head* is the top of the water column within a piezometer, referenced to the Dutch sea level (NAP). The observed values in a time series in this work represent the hydraulic head at a given date, measured in centimeters. Concretely, the hydraulic head $z_{i,t}$ represents the water level at piezometer i for timestep t. A higher hydraulic head implies greater water pressure within the dike caused by internal or external stressors. Rapid surges in $z_{i,t}$ are known as *peaks*. Peaks are local maxima in the time series exceeding a certain baseline by a minimum rise, and are separated by a minimum distance. Forecasting accuracy at extreme values is particularly important, as these surges are critical for dike safety assessments.

2.2 Problem statement

The forecasting problem addressed in this thesis is to predict future hydraulic head values measured by piezometers in dikes. Each piezometer i produces a univariate time series $\mathbf{z}_{i,1:\ell} = [z_{i,1}, \ldots, z_{i,\ell}]$ of length ℓ, with daily observations indexed by t. Let $\mathbf{Z} = \{\mathbf{z}_j\}_{j=1}^{N}$ denote the collection of all time series and $\mathbf{X}$ the associated covariates.

At each forecast origin t, the objective is to predict the next H steps $\mathbf{z}_{i,t+1:t+H}$ given historical observations and covariates for the specified target using a subset S_i as modeling context. The target series may either be a single series i or the site-level stack containing i. This includes local models (using only i), global univariate models (pool across multiple series j but forecast only for target i), and global multivariate models (jointly forecast all models at a site).

Formally, we aim to learn a forecasting function Ψ that produces a predictive distribution

$$p(\mathbf{z}_{i,t+1:t+h} \mid \mathbf{Z}_{S_i,1:t}, \mathbf{X}_{S_i,1:t+H}; \theta_i, \Phi) \tag{2.2}$$

where $\mathbf{X}_{1:t+H}$ are covariates, θ_i are parameters specific to series i, and Φ are the learnable parameters of the algorithm. The optimal function is defined as the configuration of Ψ that minimizes the expected loss

$$\Psi^* = \arg\min_{\Psi} \mathbb{E}_t[L(\mathbf{z}_{i,t+1:t+H}, \hat{\mathbf{z}}_{i,t+1:t+H})], \tag{2.3}$$

where L is a loss function measuring the discrepancy between predicted and observed values over rolling backtest windows.

Chapter 3

Related work

This chapter discusses the current state of the literature regarding the three main topics in this work. First, we review the general time series forecasting literature, exploring classical approaches as well as state of the art techniques and algorithms. Second, we investigate the literature on hydraulic head forecasting in both dikes and wells. Third, we discuss AutoML for time series forecasting with an emphasis on the AutoGluon-TimeSeries framework.

3.1 Time series forecasting

Time series forecasting occurs in many domains, from economics to energy and climate science. Traditional forecasting methods relied on *local* models, which train a separate model for each time series. Well-known approaches include ARIMA [21] and exponential smoothing [11]. In a local univariate model, each series is treated as an independent regression task, and patterns learned for one series are not transferred to another. In contrast, *global* models fit a single model across multiple time series, sharing parameters and learned representations across the group. This global approach enables the discovery of cross-series patterns, potentially improving performance when appropriate strategies are used to select time series on which the models are trained [22], [23]. Global models can be separated into two categories: (1) *global univariate*, which share parameters across series but still predict one series at a time, and (2) *global multivariate*, which jointly predict multiple series at once [11]. Global univariate models may learn from multiple related time series but still output forecasts for each series independently: for example, rising hotel bookings in summer can help improve forecasts of restaurant reservations in the same area. Global multivariate models instead produce joint forecasts across series, such as predicting traffic flows at multiple road segments simultaneously during rush hour, when they are strongly interdependent.

Interest in global models has grown over time, and global models have become prominent in large-scale forecasting competitions such as the M-series [24], [25]. In these competitions, participants forecast a large collection of time series, with submissions evaluated on an unseen test set. Because these competitions attract hundreds of innovative approaches, their results are widely regarded as benchmarks for the state of the art. While the M3 competition [26] was dominated by local statistical methods, M4 [24] marked a turning point: despite the dataset's heterogeneity, the top two models were global univariate approaches [24]. Deep learning–based global models such as DeepAR [27] and TFT [28] further accelerated this shift by using shared representations across many series. In contrast, the M5 [25] accuracy

competition was based on a highly correlated retail dataset, and all the top 50 submissions contained global models [25]. In addition, most submissions did not train one model on all time series in the dataset, but partitioned the time series into subsets, on which multiple global models were then trained. Finally, both competitions showed that ensembling was crucial: the best submissions used many different models simultaneously and combined the forecasts of these models to produce the best results. These three insights from the M5 competition strongly influence our experimental design. After we validate that global approaches do indeed outperform local models on our dataset, we investigate the best way to partition the time series in our dataset to train our global models. Ensembling is an essential part of the most important library used in this thesis, AutoGluon-TimeSeries, which we will discuss in Section 3.3.

Beyond the competition setting, recent years have seen major advances in time-series forecasting through transformer-based architectures. The Multi-resolution Time-Series Transformer (MTST) introduces a multi-branch patch-based architecture that captures both long-term trends as well as high-frequency local dynamics, and achieves state-of-the-art accuracy on several benchmark datasets [29]. Another recent transformer-based model is Chronos, a foundation model that tokenizes time series for zero-shot probabilistic forecasting, with its Chronos-Bolt variant reaching up to 250x faster inference and improved accuracy compared to the base models [30].

3.2 Hydraulic head forecasting

Forecasting hydraulic head time series is difficult due to their complicated dynamics, which are influenced by the external water level, subsurface processes, seasonal cycles, evaporation, and precipitation [3]. Forecasting hydraulic heads has several applications. For example, they help manage groundwater resources in wells, and they are essential for determining the failure probability of a dike. Despite these variations, the fundamental problem of predicting groundwater levels is comparable, and methods from both domains are relevant to this work.

Early studies on hydraulic heads in wells relied on classical time series models or simple machine learning approaches (e.g., multiple linear regression; early ANNs) [13], [31], [32]. These classical approaches performed well on less complex time series, but struggled on time series with nonlinear and long-term dependencies [33]. In recent years, research has shifted toward the use of more advanced machine learning and hybrid techniques, including support vector machines (SVM), neuro-fuzzy systems, and deep neural networks [34]. Recent work shows that deep learning models such as LSTM and CNN architectures often outperform shallower methods in modeling nonlinear dynamics and long-term dependencies [35]. For instance, CNN-based models outperformed a large set of traditional and deep learning models for Iranian groundwater levels [36], SVMs best captured regional aquifer behavior in South Africa [37], and an LSTM architecture achieved the lowest error in India [38].

In the context of dikes, hydraulic head forecasting is often used to predict the *phreatic line* (the saturated–unsaturated zone interface), which is a critical input for dike stability calculations and safety assessments. Most of the current literature on hydraulic head in dikes has focused on physics-based or conceptual models, such as Pastas

[39], a Python library that models the response of groundwater to subsurface conditions and hydrological stresses. On the other hand, software like Plaxis [40] is a geotechnical finite-element suite for coupled flow–deformation used to create a 2D or 3D model of a specific dike section, mapping the subsurface conditions of a dike to detailed groundwater flow simulations. These approaches require detailed site characterisation and repeated calibration, making them costly and hard to scale beyond localised sections. Recent studies focusing more on data-driven methods in this domain show the possibilities of different models. In a study on the subsurface conditions of dikes in one of the provinces of the Netherlands, nonlinear time series models have been shown to better capture hydraulic head responses to heavy rainfall than simpler approaches [8]. In another study, LSTM models trained on hydrodynamic model outputs have been shown to predict flood inundation patterns after dike breaches with high accuracy (MAE=0.045 m) [41]. Together, these studies demonstrate that data-driven approaches are becoming feasible alternatives for short-term scenario evaluations, even though traditional dike-related modeling is still centered on physics-based models and observational monitoring.

Overall, the literature presents a wide range of models applied to hydraulic head forecasting, mainly applied to wells, but no single approach consistently outperforms others. The large variety of different high-performing models across locations highlights the strong dependence of groundwater dynamics on local environmental and subsurface conditions. A large-scale survey confirms that the effectiveness of machine learning methods for groundwater level forecasting is highly dataset-specific, recommending the testing of multiple methods and the use of ensembling [34]. However, this leaves selection and ensembling largely manual and expert-driven, opening up the opportunity for AutoML to systematize and scale this process.

3.3 Automated Machine Learning

Automated Machine Learning (AutoML) aims to reduce the human effort in selecting models, features, and hyperparameters by automating the search for high-performing modeling pipelines. In time series forecasting, AutoML tools aim to handle tasks such as algorithm selection, hyperparameter tuning, and ensemble construction without requiring deep forecasting expertise from the user. Over the last few years, a number of open-source AutoML frameworks have introduced support for time series tasks. Examples include `NeuralProphet` [42], `FLAML` [43], `AutoTS` [44], and research projects like `HyperTS` [45]. Many of these frameworks experiment with a diverse set of traditional and deep learning models, and attempt to find the best model for a given problem. Additionally, these frameworks rely on some sort of hyperparameter optimization, often either Bayesian Optimization or an evolutionary approach. This combination of model selection and hyperparameter optimization can be computationally expensive, and in many frameworks is not combined with an ensembling approach, which, given the results of the M-challenges, is an important aspect in the state of the art of time series forecasting.

Among all other frameworks, *AutoGluon–Timeseries* (AG-TS) has established itself as a state-of-the-art framework for time series forecasting [16]. The core idea behind AG-TS is to focus on ensembling over hyperparameter optimization. AG-TS uses

a broad selection of models, and initializes these models with robust hyperparameter presets that are expected to perform well across domains. The models included in AG-TS range from local methods like ARIMA [21] and ETS [46], to global methods, including tabular methods like LightGBM [47] and deep learning methods like DeepAR [27], PatchTST [48] and Temporal Fusion Transformer [28]. AG-TS fits each model on the given dataset, and then ensembles the models using the forward selection algorithm for up to K steps (default K=100) [49]:

$$\hat{y}^{ensemble}_{i,T+1:T+H} = \sum_{m=1}^{M} w_m \cdot \hat{y}^{(m)}_{i,T+1:T+H} \quad \text{with} \quad w_m \geq 0, \sum_{m=1}^{M} w_m = 1. \tag{3.1}$$

Where weights w_m are added greedily to minimize the validation loss.

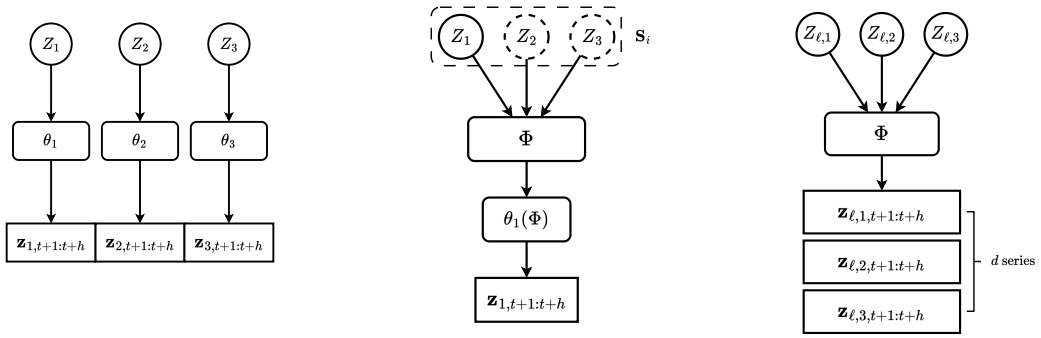
In a benchmark across 29 diverse open-source forecasting datasets, AG-TS outperformed traditional models as well as deep learning methods and AutoML frameworks [16]. Additionally, AG-TS was also found to perform best among seven other AutoML tools on a benchmark of 17 "smart city" datasets [17]. This consistent performance across various domains shows the generalizability and robustness of its ensemble-based approach.

In hydraulic head forecasting, AutoML remains under-explored; most prior work relies on custom, site-tailored pipelines. We therefore systematically evaluate AG-TS on hydraulic head data and introduce domain-specific extensions (e.g., selective pooling and peak-aware features) tailored to groundwater and dike monitoring.

Chapter 4

Methods

This chapter outlines the methods used to investigate AutoML for time series forecasting in dikes. We start by investigating the performance of AG-TS in a local univariate setting against classical methods and AutoML frameworks. We then build on this by exploring selective pooling methods to increase the performance of AG-TS in a global univariate setting. We then test the AG-TS approach in a multivariate setting by reframing the methodology through another package, and compare the performance to the previous approaches. Finally, we propose a method to improve the forecasting performance of AG-TS around peaks in the hydraulic head data.



(A) Local univariate (B) Global Univariate (C) Multivariate

FIGURE 4.1: Outputs by model class. (a) Local univariate: each model outputs an h-step vector for its own series. (b) Global univariate: shared Φ with a head $\theta_i(\Phi)$ that produces a *single* h-step vector for target z_i. (c) Multivariate: shared Φ produces a joint $d \times h$ output (multiple targets over the horizon).

4.1 Local univariate

Local univariate models train a separate model for each of the N time series independently, forecasting only its future values. Patterns learned for one series are not transferred to others unless explicitly provided through shared covariates. This approach has been widely used in classical statistical methods and early applications of neural networks [11].

Formally, for the i-th time series, a local univariate model estimates the predictive distribution:

$$p(\mathbf{z}_{i,t+1:t+h} \mid \mathbf{z}_{i,1:t}, \mathbf{X}_{i,1:t+h}; \theta_i), \quad \theta_i = \Psi(\mathbf{z}_{i,1:t}, \mathbf{X}_{i,1:t+h}), \tag{4.1}$$

where Ψ is a function mapping input features to the parameters θ_i of the probabilistic model, local to the i-th series. Multidimensional covariates $\mathbf{x}_{i,t}$ may be included, but the task remains univariate since only one time series is forecasted.

4.2 Global univariate

While local models can capture patterns specific to a single site, they ignore patterns that may occur across locations. In practice, hydraulic head series often share seasonalities and hydrological response dynamics, which motivates pooling across sites. A global univariate model uses this shared information while still producing forecasts for individual series [11].

The global univariate model is formalized as:

$$p(\mathbf{z}_{i,t+1:t+h} \mid \mathbf{Z}_{1:t}, \mathbf{X}_{1:t+h}; \theta_i), \quad \theta_i = \Psi(\mathbf{z}_{i,1:t}, \mathbf{X}_{i,1:t+h}, \Phi), \tag{4.2}$$

where Φ are the shared parameters of the model across the N time series in $\mathbf{z}$. This allows the model to learn information across all series, while the output for each time series through the parameters θ_i remains specific to that series. The model can thus learn cross-site features, which can improve overall performance. Although the model may see all series jointly, the forecasts for each are still produced independently.

A problem can arise if we train the global model on all N series. For some N, if a subset of series follows a distribution very different from the majority, the model will also attempt to optimize for these, which can lead to conflicting learned features. To address this, we apply *selective pooling*: instead of training on the full set $\mathbf{Z}$, we select a subset $\mathbf{S}_i \subseteq \mathbf{Z}$ for each target series $\mathbf{z}_i$. The selection is given by a function

$$S : \{\mathbf{z}_1, \ldots, \mathbf{z}_N\} \to \{\mathbf{S} \subseteq \mathbf{Z} \mid \mathbf{z}_i \in \mathbf{S}\} \tag{4.3}$$

which maps the set of all series to an informative subset $\mathbf{S}_i$ to be used when forecasting $\mathbf{z}_i$. Different definitions of $S(\cdot)$ correspond to different selection strategies.

We propose a novel selective pooling method: Bayesian Optimization for Selective Pooling (BOSP), and compare it against methods proposed in other research. The proposed algorithm is inspired by the Sequential Model Based Optimization (SMBO) algorithm as proposed by Hutter et al. [50]. Where the main function of SMBO is to automatically tune the hyperparameters of a given algorithm, our proposed method automatically optimizes the subset of time series to train a global univariate model on. The procedure is summarized in Algorithm 1 and explained step by step below, with references to the corresponding pseudocode lines.

The algorithm begins by extracting features from each time series using TSFresh [51] at line 3, creating a compressed representation of the temporal patterns of each series. TSFresh is a widely used library in academic work and provides a large feature set. It has been shown that Bayesian Optimization suffers when the dimensions of the input increases over 20 [52]. We therefore use Principal Component Analysis (PCA) [53] to reduce the dimensionality of the feature representation to 19 dimensions at line 4. Each subset of time series is encoded through three key representations: the mean and variance of its dimensionality-reduced time series feature vector representations, and the minimum distance to the target series in feature space. We

start by generating initial subsets of time series at line 7 following a structured sampling scheme. We first create a fixed number of pair- and triplet-sized subsets to ensure coverage of very small subsets. The remaining subsets are then drawn according to a predefined size distribution: 40% with 4–7 series, 40% with 8–15 series, and 20% with more than 15 series. The initial subsets are then evaluated by AG-TS at line 9, generating pairs of feature vectors and performance scores. As evaluating many subsets using AG-TS is computationally expensive, we limit the models used by BOSP in subset evaluation to three models: DirectTabular, Temporal Fusion Transformer and DeepAR. We use these three models as they are SOTA in their respective model types (tabular, transformer and neural network).

The observations of error-subset pairs are fed into a Gaussian Process Regressor (GPR) at line 16, which maintains a posterior distribution $p(F|D)$ over the function that maps subset features to performance. Since our subset representation now exceeds 20 dimensions (mean and variance of PCA components, minimum distance to the target, and subset size), we employ an ARD kernel that assigns separate length scales to each dimension, allowing the GPR to down-weight irrelevant features. After collecting 30 initial observations, the GPR is used for principled exploration through Expected Improvement (EI) [54]:

$$EI(x) = (f^* - \mu - \xi)\Phi(\frac{f^* - \mu - \xi}{\sigma}) + \sigma(\phi(\frac{f^* - \mu - \xi}{\sigma})) \tag{4.4}$$

where f^* represents the best observed score, μ and σ are the predictive mean and standard deviation of the candidate subset. ξ is an exploration parameter, and Φ, ϕ are the cumulative and probability density functions of the standard normal distribution. For each iteration, candidate subsets are generated at line 17, their expected improvement computed at line 18, and the most promising candidate selected at line 19. This acquisition function balances exploitation of promising regions with exploration of uncertain areas in the subset space.

The process continues until the maximum number of iterations is reached, with each evaluation updating the GPR posterior. After BOSP has reached the maximum iterations it returns the best found subset. This subset is then used in the full AG-TS evaluation, which uses all global models provided in AG-TS. The full experimental setup is described in Chapter 5, and the hyperparameters associated with BOSP can be found in Appendix C.

Algorithm 1: Full pipeline for BOSP

Input: All time series S, target series s_t, number of initial points n_{init}, maximum iterations *max_iter*
Output: Best subset S^* and best score y^*
$F \leftarrow \text{extract\_TSFresh\_features}(S)$
$P \leftarrow \text{PCA}(F)$
$X \leftarrow \emptyset, \quad Y \leftarrow \emptyset$
$y^* \leftarrow \infty \quad S^* \leftarrow \emptyset$
$init\_subsets \leftarrow \text{generate\_diverse\_subsets}(n_{\text{init}})$
foreach $v \in init\_subsets$ **do**
    $y \leftarrow \text{evaluate\_subset}(v)$
    $X \leftarrow X \cup \{v\}, \quad Y \leftarrow Y \cup \{y\}$
    if $y < y^*$ **then**
        $y^* \leftarrow y$
        $S^* \leftarrow \text{subset}(v)$
for $iter \leftarrow 1$ **to** max_iter **do**
    $X_{\text{repr}} \leftarrow \{\text{feature\_repr}(v) \mid v \in X\}$
    Fit GP on $(X_{\text{repr}}, \log Y)$
    $candidates \leftarrow \text{sample\_candidates}(v^*, 1000)$
    $EI \leftarrow \{\text{expected\_improvement}(c) \mid c \in candidates\}$
    $v_{\text{next}} \leftarrow \arg\max_c EI[c]$
    $y \leftarrow \text{evaluate\_subset}(v_{\text{next}})$
    $X \leftarrow X \cup \{v_{\text{next}}\}, \quad Y \leftarrow Y \cup \{y\}$
    if $y < y^*$ **then**
        $y^* \leftarrow y$
        $S^* \leftarrow \text{subset}(v_{\text{next}})$
return $\{s_t\} \cup S^*,\ y^*$

We compare BOSP to several other methods for selective pooling. Two of these strategies, Dynamic Time Warping and feature-based clustering, have been shown to improve forecasting accuracy in other literature [55], [56]. In the domain of hydraulic head forecasting in dikes, it is logical that piezometers in the same dike may share similarities, we therefore also experiment with global models for all piezometers in a site. Additionally, we use greedy forward selection, which iteratively adds time series to the subset and evaluates on a proxy model. Finally, we include a full global model, which trains a single model on all time series in the dataset. We will now provide further explanation of these strategies.

- *Feature-based clustering:* We automatically extract features using TSFresh [51] and apply the K-means clustering algorithm on a maximum of 5 clusters to select the best subset.
- *Dynamic Time Warping similarity*: We select the k series with the smallest Dynamic Time Warping distance $d_{\text{DTW}}(\mathbf{z}_i, \mathbf{z}_j)$ to the target $\mathbf{z}_i$, computed on first-differenced values to remove low-frequency trends [57].
- *Greedy forward selection*: Given a target series $\mathbf{z}_t \in \mathbf{Z}$ and a proxy evaluation function $M(S)$ for any subset $S \subseteq \mathbf{Z}$, we first compute the pairwise improvement $\Delta_i = M(\mathbf{z}_t) - M(\mathbf{z}_t, \mathbf{z}_i)$ for each $\mathbf{z}_i \in \mathbf{Z} \setminus \mathbf{z}_t$, retaining only those with $\Delta_i > 0$. We initialize $S = \mathbf{z}_t, \arg\max \mathbf{z}_i \Delta_i$ and iterate over remaining candidates in descending Δ_i order, adding $\mathbf{z}_j$ only if $M(S \cup \mathbf{z}_j) < M(S)$.

Iteration stops when no $\mathbf{z}_j$ added to S results in further improvement, returning S as the selected subset.

- *Time series from the same location*: We select all series measured at the same site as $\mathbf{z}_i$, i.e., $\mathbf{S}_i = \{\mathbf{z}_j \in \mathbf{Z} \mid \ell(\mathbf{z}_j) = \ell(\mathbf{z}_i)\}$, where $\ell(\cdot)$ maps a series to its location identifier.
- *Full global model*: We train on the complete set $\mathbf{S}_i = \mathbf{Z}$ for all i, corresponding to no selective pooling.

4.3 Global multivariate

For the third modeling approach, we test whether multivariate models can improve upon the best found models in previous experiments. To keep comparisons across models as fair as possible, we replicate AG-TS's ensemble-based strategy in a multivariate setting using Darts [58]. Darts is a Python library for time series forecasting that supports both statistical and deep learning models. Darts is one of the few Python libraries that natively handles multivariate targets with a unified API, making it suitable for replicating AG-TS's ensemble strategy in a joint forecasting setting.

To ensure comparability with AG-TS, we implemented multivariate counterparts of its main models in Darts: DeepAR (as `DeepVAR`), Temporal Fusion Transformer (`TFT`), and TiDE (`TiDEModel`). Since AG-TS does not support multivariate forecasting, we reproduced its approach by matching hyperparameters, using the same feature generator and scaling strategy, and applying the same greedy ensemble selection procedure. Formally, the multivariate setup extends the predictive distribution to

$$p(\mathbf{Z}_{i,t+1:t+h} \mid \mathbf{Z}_{1:t}, \mathbf{X}_{1:t+h}; \theta), \quad \theta = \Psi(\mathbf{Z}_{1:t}, \mathbf{X}_{1:t+h}, \Phi), \tag{4.5}$$

where $\mathbf{Z}$ contains all series from the same location, as dependencies in our case only exist within a dike. This ensures that differences in performance between univariate and multivariate experiments stem from model structure rather than implementation details, allowing a fair comparison across approaches.

4.4 Peak-aware forecasting

In addition to BOSP for selective pooling, we propose a second contribution: a peak-aware forecasting strategy designed to address the frequent sharp rises in hydraulic head data. Dealing with sudden peaks is a common challenge in modeling and forecasting hydraulic head data [10], [59]. The hydraulic response of dikes to external stimuli can change significantly over time. For instance, during a prolonged dry period, the upper soil layers of a dike may dry out, reducing their hydraulic conductivity or even becoming hydrophobic. In such conditions, rainfall infiltrates only slowly, and much of the water is temporarily stored in the unsaturated zone. By contrast, when the soil is already wet or near saturation, infiltration occurs rapidly and groundwater heads can rise sharply as the infiltration pathways are open and rainfall rapidly transmits downward. These peaks are difficult to model, as they can be triggered by drought–rainfall interactions but also by a variety of other hydrological and geotechnical processes [3].

We define a peak at time t in the hydraulic head time series $\mathbf{z}_{1:T}$ as a local maximum within a lookback window L, where the rate of change is larger than a certain threshold η, defined as the change of hydraulic head (m/day). We set $\eta = 0.8$ by calibrating on a set of representative time series and verifying that the detected events aligned with the desired peaks. To ensure the detected peaks are distinct local maxima, we additionally require a minimum spacing D from the previously identified peak. Formally,

$$y_t = \mathbb{1}\left[\frac{z_t - z_{t-L}}{L} > \eta,\ z_t = \max_{s\in[t-\lfloor W/2\rfloor,\, t+\lfloor W/2\rfloor]} z_s,\ z_t > \tfrac{1}{r}\sum_{s=t-r}^{t-1} z_s,\ t - t_{\text{last}} \geq D\right], \tag{4.6}$$

where W is the peak-window size, r is the length of the recent-past comparison, and t_{last} is the time of the most recent peak.

To improve the performance of our model at these peak locations, we use a separate model that is trained to forecast peaks. The motivation for introducing a separate peak classifier is that peaks arise from a variety of hydrological processes, as described earlier. These dynamics are hard for a forecaster like AG-TS to capture, as it mainly relies on future covariates and aims to model the average behavior of the time series, rather than sparse peaks. By integrating peak probabilities as covariates, we provide the forecaster with an explicit signal as to where sharp upwards rises are likely to occur. This allows the model to relate high peak probabilities to sharp rises in the hydraulic head, improving the predictive accuracy at these timesteps, while also potentially refining point forecasts by avoiding to predict sharp rises in the hydraulic head when no peak is expected.

We train an XGBoost model [60] to perform a binary classification task, predicting whether a point within the forecast window will be a peak or not. The full configuration for the model is described in Appendix D, and the procedure of integrating the predicted peaks into the AG-TS forecaster is described in Algorithm 2. The pipeline begins by training a multi-output XGBoost classifier on lagged hydraulic head, precipitation, and evaporation values, as well as cumulative and trend statistics of these values (line 6). The model outputs peak probabilities for the next H timesteps. The integration process at line 13 starts by augmenting the AG-TS training set with predicted peak probabilities, treated as additional covariates alongside hydraulic head, precipitation, and evaporation. AG-TS is then fitted on the enriched data to learn both hydro-meteorological drivers and peak likelihood. At inference, the classifier predicts peak probabilities for each forecast window at line 14, which are passed to AG-TS as covariates for the final hydraulic head forecasts.

Algorithm 2: Peak-aware covariate pipeline (train on train set, then backtest)

Input: Set of time series S, forecast horizon H, feature builder $\mathcal{F}$, number of backtest windows K

Output: $\widehat{\mathbf{z}}_{\tau_k+1:\tau_k+H}$ for all windows K

0. Define train/backtest split

Let $\{\tau_1, \ldots, \tau_K\}$ be the forecast origins.

Let $\mathcal{D}_{\text{train}} = \{t \mid t \leq \tau_1\}$ and, for window k, $\mathcal{D}_{\text{val}}^{(k)} = \{\tau_k+1, \ldots, \tau_k+H\}$.

1. Train peak classifier on $\mathcal{D}_{\text{train}}$

Build $X_{\text{train}} = \mathcal{F}(S|_{\mathcal{D}_{\text{train}}})$ and multi-horizon targets $Y_{\text{train}} = \text{peaks}|_{\mathcal{D}_{\text{train}}}$.

Train multi-output XGBoost $\mathcal{C}$ on $(X_{\text{train}}, Y_{\text{train}})$ with class-imbalance handling.

2. Train AG-TS forecaster on $\mathcal{D}_{\text{train}}$

Augment the training set with in-sample peak probabilities $\widehat{p}_{t+1|t}$ predicted by $\mathcal{C}$

Known covariates: $\mathbf{x}_t = (\text{precip}_t,\ \text{evap}_t,\ \widehat{p}_{t+1|t})$.

Fit forecaster $\mathcal{A}$ on $\left(z_{1:T}|_{\mathcal{D}_{\text{train}}},\ \mathbf{x}_{1:T}|_{\mathcal{D}_{\text{train}}}\right)$ with prediction length H.

3. Rolling backtest (no refit)

$\mathcal{Y} \leftarrow [\,]$

for $k = 1$ **to** K **do**

`// At forecast origin` τ_k

Predict future peaks $\widehat{p}_{\tau_k+1:\tau_k+H} = \mathcal{C}\big(\mathcal{F}(S|_{\leq \tau_k})\big)$.

Set future covariates $\mathbf{x}_{\tau_k+1:\tau_k+H} = (\text{precip},\ \text{evap},\ \widehat{p})$.

Forecast $\widehat{\mathbf{z}}_{\tau_k+1:\tau_k+H} \leftarrow \mathcal{A}.\text{predict}(\text{history } \leq \tau_k,\ \mathbf{x}_{\tau_k+1:\tau_k+H})$.

Append $\widehat{\mathbf{z}}_{\tau_k+1:\tau_k+H}$ to $\mathcal{Y}$

return $\mathcal{Y}$

Chapter 5

Experimental setup

In this chapter we will outline the experimental setup to evaluate the methods discussed in Chapter 4. Our experiments are set up to address the following four queries:

- What is the best-performing method for local univariate hydraulic head forecasting among classical approaches and AutoML frameworks?
- Can global univariate models outperform local univariate models, and which strategies are most effective?
- Does incorporating location specific multivariate information improve forecasting accuracy over the best global univariate model?
- How can forecasting performance during sudden hydraulic head peaks be improved?

We will start by addressing the baselines used in our experiments, and how the relevant baselines evolve throughout the experiments. Next, we will discuss the dataset we use in our experiments. After this, we will discuss the evaluation protocol, and highlight how we test the performance of the models. Finally, we will discuss the evaluation metrics we use to quantify this performance.

5.1 Baselines

In line with the research questions, the set of baselines against which we compare evolves across experiments. The local univariate experiments serve two purposes: (1) to assess which AutoML framework can outperform classical forecasting methods and alternative AutoML frameworks, and (2) to establish a performance baseline against which subsequent global and multivariate models can be compared. Specifically, we compare the following set of models and AutoML frameworks: `Naive`, `Seasonal Naive`, `Linear Trend`, `Exponential Smoothing`, `AutoRegressive`, `ARIMA`, `SARIMA`, `HyperTS`, `AutoTS`, and `AutoGluon-TimeSeries`. The classical models were selected as they are well-represented in time series forecasting literature and provide interpretable baselines. The three AutoML frameworks were chosen as they provide complementary optimisation strategies; `HyperTS` uses Monte Carlo tree search and adaptive grid search, `AutoTS` employs evolutionary algorithms, and `AutoGluon-TimeSeries` emphasizes ensemble learning.

In the global univariate experiments, the comparison is extended to whether pooling across time series can improve upon the strongest local baseline. The global

multivariate setting further builds on this by testing whether incorporating multivariate dependencies leads to gains over the best-performing global univariate approach. Finally, the peak-aware experiments focus specifically on improving predictions during sudden hydraulic head surges, where the best-performing method from the earlier stages serves as the baseline.

5.2 Data

The dataset used in our experiments consists of measurements from 10 monitoring sites on dikes in the province of South-Holland, the Netherlands and is publicly available through the open-data portal of the TU Delft [61]. The original dataset contains 65 time series, but 14 of these were excluded due to missing values or erroneous measurements, resulting in a total of 51 time series. Each site contains between 4 and 6 piezometers placed along a cross-sectional profile of a dike. A typical cross-sectional profile is illustrated in Figure 1.1. Each piezometer is equipped with a filter at the bottom, allowing groundwater to enter the tube. A diver placed inside the piezometer records the water level, which rises as pore water pressure increases. The measurements are expressed as hydraulic head values relative to Normaal Amsterdams Peil (NAP), the Dutch national reference for approximated mean sea level, and serve as the target variable in our forecasting tasks.

All series were resampled to daily resolution. To enrich the dataset, we include two covariates provided by the Royal Netherlands Meteorological Institute (KNMI): daily precipitation (mm/day) measured at a South-Holland weather station, and potential evaporation (mm/day) estimated using the Makkink method [62]. Both precipitation and evaporation datasets are publicly available through the Meteobase portal [63].

The placement of piezometers along the cross-section leads to distinct behaviors: sensors near the canal exhibit relatively stable signals, whereas those located on the slope of the dike show larger variability in hydraulic head fluctuations. Eight sites were recorded between February 2020 and January 2025, while two sites were recorded between May 2020 and January 2025. To make sure the performance across all time series is comparable, we fixed the observation window of each time series to start at 2020-07-08, and to end at 2024-12-31 resulting in time series of 1638 data points each. The dataset contains very few missing values or outliers (<0.1%), which were handled through linear interpolation. A full overview of the time series included in this work can be found in Appendix A.

5.3 Evaluation protocol

All experiments follow a standard evaluation protocol to ensure comparability across approaches and research questions. We use a multi-window backtesting approach to generate forecasts, with a prediction horizon of three days. The three day forecast horizon was chosen to mimic practical application; As we include precipitation and evaporation as covariates, these values are forecasted in practice, and long term forecasts lose most of their reliability after three days. For each experiment, we define K rolling forecast origins, and evaluate the predictions over the according K prediction windows. To maintain comparability across sites, all test sets are taken as the final K windows of each time series. As all time series end on the same calendar day, this ensures that differences in performance are due to modeling choices, and

not differences in temporal coverage. To account for stochasticity in model training, each experiment is run on three different random seeds. The results are always reported as the mean across runs, and for completeness we also provide the standard deviation.

5.4 Evaluation metrics

Performance is assessed using two metrics: mean absolute error (MAE) and mean absolute percentage error (MAPE). MAE provides a measure that is directly interpretable in physical units (centimeters relative to NAP), which directly indicates how far predictions differ from actual hydraulic heads. MAPE complements this by providing a scaled measure through normalizing the errors relative to the magnitude of the time series. MAPE provides extra insight into the relative performance across piezometers, as piezometers placed in different sections on a cross profile have different levels of variability.

Formally, for a forecast horizon of length H (here $H = 3$) and K backtest windows with forecast origins $\tau_1, \ldots, \tau_K$, the metrics are defined as:

$$\text{MAE} = \frac{1}{KH} \sum_{k=1}^{K} \sum_{h=1}^{H} \left| z_{\tau_k+h} - \hat{z}_{\tau_k+h} \right|, \tag{5.1}$$

$$\text{MAPE} = \frac{100}{KH} \sum_{k=1}^{K} \sum_{h=1}^{H} \left| \frac{z_{\tau_k+h} - \hat{z}_{\tau_k+h}}{z_{\tau_k+h}} \right|, \tag{5.2}$$

where z_{τ_k+h} denotes the observed value at horizon h after forecast origin τ_k, and $\hat{z}_{\tau_k+h}$ is the corresponding model prediction. Both metrics aggregate errors across all horizons and all backtest windows, ensuring a robust evaluation of forecasting performance. Additionally, we provide the 'wins' for each model. The wins correspond to the amount of time series on which a listed model returned the lowest MAE.

5.5 Statistical significance testing

To assess whether observed differences in forecasting performance are statistically significant, we apply the Wilcoxon signed-rank test [64] with the *less* alternative hypothesis, testing whether the first method achieves lower errors than the second. The Wilcoxon test is a nonparametric paired test that compares distributions of errors across time series.

For each site, model errors are first averaged across the three random seeds to obtain a single score. This ensures that stochasticity in training does not inflate the number of paired samples. The number of pairs, n, is therefore equal to the number of time series ($n = 51$). In this test, the statistic W denotes the sum of the ranks of positive differences (i.e., cases where the first method performs worse). A small W value indicates consistent improvement of the first method across sites.

Throughout the results chapter, we report the Wilcoxon test outcome as (W, n, p) when comparing a candidate method to the best-performing baseline. Statistical significance is evaluated at the conventional $\alpha = 0.05$ level.

Additionally, we provide a critical distance (CD) diagram of the Nemenyi test [65]. The Nemenyi test is a nonparametric post-hoc procedure that evaluates whether differences between average ranks of multiple strategies are statistically significant. The resulting CD diagram visualizes the mean rank of each strategy across all time series, while a horizontal bar connects methods whose rank differences fall below the critical distance. In this way, the plot provides an overview of which methods perform comparably and which are significantly different. The Wilcoxon signed rank test is applied to all experiments, and the Nemenyi test with CD diagram is applied to all experiments containing more than two models in the comparison.

Chapter 6

Results

In this chapter we report the experimental results and answer the research questions posed in Chapter 5. Throughout this chapter, values are reported as *mean* $\pm$ *standard deviation* across the 51 piezometer time series, averaged over three random seeds. A *win* denotes the number of sites on which a method achieved the lowest MAE. Boldface values indicate the best mean performance across methods; they do not imply statistical significance, which is assessed separately using the procedure described in Chapter 5. We proceed in four steps. First, we benchmark AutoGluon-TimeSeries (AG-TS) against classical local models and alternative AutoML frameworks in the *local univariate* setting. Second, we extend AG-TS to a *global univariate* approach and compare several pooling strategies. Third, we reframe forecasting as a *global multivariate* problem at the site level. A full per-site comparison of the results of these three models, as well as boxplots for the distribution of the errors, is provided in Appendix B. Finally, we evaluate BOSP+Peak designed to improve performance specifically during sudden hydraulic head surges. The full results table of BOSP+Peak can be found in Appendix D.

6.1 Local univariate results

Table 6.1 shows the performance of the local univariate models across all piezometers. Among the classical baselines, the AutoRegressive model achieved the best average results. This suggests that for some time series the dynamics are not too complex, allowing a relatively simple model to capture them well. ARIMA and SARIMA only provided small improvements over the Naive and exponential smoothing forecasts, and their large error variances show that these methods were not reliable across sites.

The two alternative AutoML frameworks (AutoTS, HyperTS) rely on evolutionary search to pick a single model family before hyperparameter tuning. This often produced unstable selections across runs and sites: when the candidate space includes many comparable options, small differences in search trajectories lead to different winners and varying errors. HyperTS, in particular, underperformed on average relative to the Naive baseline but did achieve the best score on four sites. This is evidence that it can occasionally discover strong configurations, but is unstable overall. AutoTS behaved more consistently but rarely achieved site-level wins. Overall, both approaches show that "finding one good model" is possible in principle but lacks robustness across time series.

AutoGluon-TimeSeries (AG-TS) showed the best performance overall. It obtained the lowest mean absolute error (3.12) and mean absolute percentage error (1.99),

and it produced the lowest error on 34 of the 51 sites. Its error variance was also much lower than the other methods, which indicates that it is more reliable across different conditions. The ensemble strategy of AG-TS, which combines multiple models rather than relying on a single one, likely contributes to both its accuracy and its stability.

TABLE 6.1: Local univariate model performance comparison

Model	MAE	MAPE	Wins
AG-TS†	**3.121 ± 0.681**	**1.99 ± 0.38**	**34**
AutoRegressive	3.367 ± 3.138	2.22 ± 3.66	10
ARIMA	3.559 ± 3.801	2.35 ± 4.29	2
AutoTS†	3.678 ± 3.422	2.33 ± 3.01	1
SARIMA	3.737 ± 3.577	2.39 ± 3.17	0
Naive	3.930 ± 3.785	2.60 ± 4.22	0
Exponential Smoothing	4.014 ± 3.839	2.64 ± 4.24	0
HyperTS†	4.183 ± 4.712	2.78 ± 4.65	4

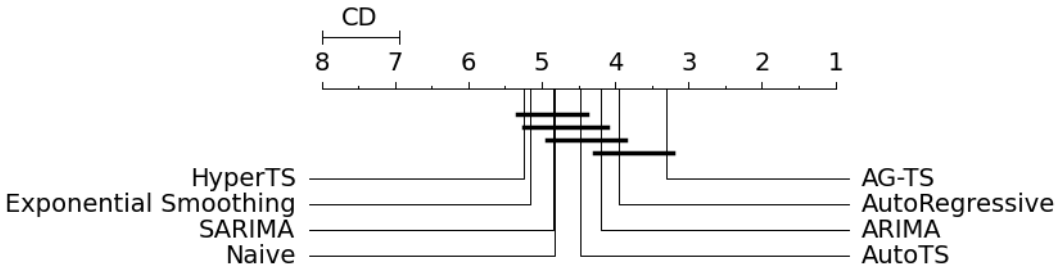


FIGURE 6.1: CD diagram of the Nemenyi test for the local univariate models. The numbers represent the mean ranks, where lower is better. Ranks with non-significant difference are connected with a horizontal line.

To test whether the improvement of AG-TS over the classical baselines was statistically significant, we applied a Wilcoxon signed-rank test across all time series. The test comparing AG-TS to the best performing baseline (AutoRegressive) resulted in $W = 549$, $n = 51$, and $p = 0.14$. This indicates that the difference is not statistically significant at the conventional $\alpha = 0.05$ level.

With these results, we can answer the first research question: *What is the best-performing method for local univariate hydraulic head forecasting among classical approaches and AutoML frameworks?* AG-TS achieved the best overall performance, with the lowest mean error (MAE 3.12, MAPE 1.99), the most site-level wins (34 of 51), and substantially lower variance than the alternatives. The Wilcoxon test comparing AG-TS to the next-best model (AutoRegressive) did not confirm a statistically significant improvement ($W = 549$, $n = 51$, $p = 0.14$), but AG-TS nevertheless provides the most consistent and reliable results. We therefore adopt it as the local univariate reference model for subsequent experiments.

6.2 Global univariate results

In the global univariate setting (Table 6.2), strategies based on simple pooling did not improve over the local AG-TS baseline. The full global model, which trains on all series jointly, showed high error variance. This is a classic negative-transfer effect: sites differ in hydrological response (e.g., lag and gain with respect to precipitation/evaporation), so parameters that help one subset can degrade others. Unable to separate or weight incompatible series, the model fits an "average" relationship that fits no site particularly well.

TABLE 6.2: Global univariate strategy performance comparison averaged across all time series. All strategies train an AG-TS model on the subset proposed by that strategy; The 'Local univariate' row is the best found model in Section 6.1

Strategy	MAE	MAPE	Wins
BOSP	**2.615 ± 0.071**	**1.87 ± 0.04**	**46**
Local univariate AG-TS	3.121 ± 0.681	1.99 ± 0.38	5
Full global model	3.773 ± 2.166	2.64 ± 3.52	0
DTW similarity	3.807 ± 0.406	2.69 ± 0.41	0
Same location	3.831 ± 0.409	2.69 ± 0.59	0
Greedy forward selection	3.867 ± 0.314	2.67 ± 0.38	0
Feature-based clustering	3.893 ± 0.492	2.73 ± 0.59	0

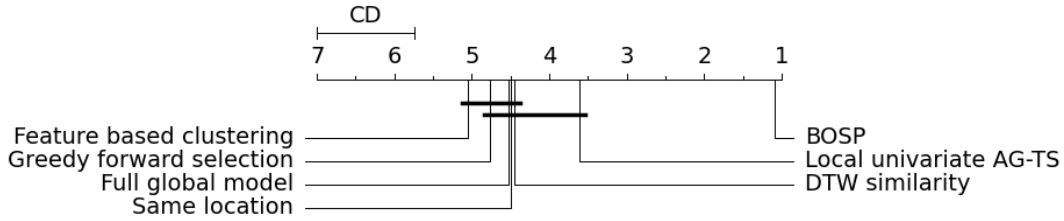


FIGURE 6.2: CD diagram of the Nemenyi test for the global univariate models. The numbers represent the mean ranks, where lower is better. Ranks with non-significant difference are connected with a horizontal line.

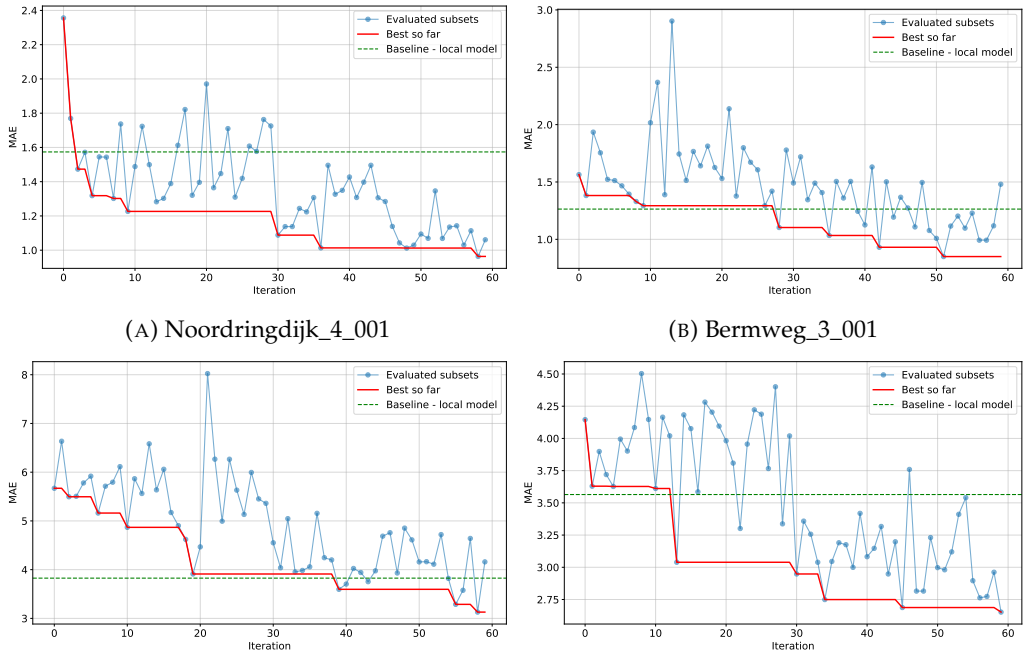
The simple selective pooling strategies also struggled. Feature-based clustering showed the worst performance on average, while this approach has had success in other domains. Selective pooling through DTW similarity, Greedy forward selection and taking time series from the same location also did not result in any significant improvement over the local univariate baseline. Clearly, these methods struggle to find subsets of time series that are informative to the global univariate approach using AG-TS.

BOSP overcomes this by learning which subsets are informative. We embed each candidate subset into a feature vector (summarizing the series in the subset) and model the mapping from subset features to validation loss with a Gaussian Process (GP). After initial random exploration of subsets to shape the GP posterior, we select new subsets via the Expected Improvement (EI) acquisition, which trades off exploring uncertain regions and exploiting areas with predicted lower loss. As shown in Figure 6.3, once the posterior stabilizes (typically after $\sim$30 evaluations), EI repeatedly proposes subsets that beat the local baseline and progressively improves

the best error. This explains BOSP's low mean error (MAE 2.62; MAPE 1.87) and dominant site-level wins (46/51).

A paired analysis across all time series confirms that BOSP provides a statistically significant improvement over the local baseline. The Wilcoxon signed-rank test comparing BO global univariate to the Local univariate model resulted in $W = 215$, $n = 51$, one-sided $p < 10^{-8}$. This shows that the performance gains of BOSP are both consistent across sites and statistically robust.

This allows us to answer our second research question: *Can global univariate models outperform local univariate models, and which strategies are most effective?* The results showed that a full global model has to deal with too much noise to improve performance overall, and that simple selective pooling strategies do not partition the time series into subsets that are informative for AG-TS. Using Bayesian Optimization to optimize the subset selection did prove informative, resulting in a clear improvement over the local univariate baseline.



(A) Noordringdijk_4_001

(B) Bermweg_3_001

(C) Molenlaan_3_001

(D) Duifpolderzuid_2_001

FIGURE 6.3: Bayesian optimization progress for four representative piezometers. Blue markers denote evaluated subsets, the red line the best subset found so far, and the dashed green line the baseline performance of a local run. The baseline is the best found error by the local univariate model.

6.3 Global multivariate results

Table 6.3 compares the performance of the multivariate models with the best global univariate approach (BOSP) and the local univariate baseline. The multivariate models did not outperform the global univariate strategy. On average, the multivariate error was higher (MAE 3.44, MAPE 1.85) but did win on 12 sites, compared to 37 site-specific wins for BOSP.

TABLE 6.3: Multivariate strategy performance comparison averaged across all time series

Strategy	MAE	MAPE	Wins
BOSP	**2.615 ± 0.071**	1.87 ± 0.12	**37**
Local univariate AG-TS	3.121 ± 0.681	1.99 ± 0.38	2
Multivariate	3.444 ± 0.187	**1.85 ± 0.10**	12

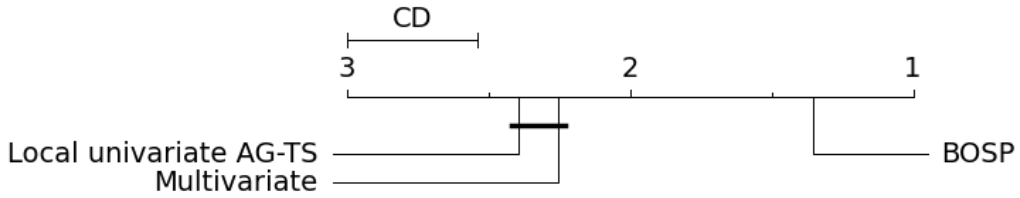


FIGURE 6.4: CD diagram of the Nemenyi test for the multivariate models. The numbers represent the mean ranks, where lower is better. Ranks with non-significant difference are connected with a horizontal line.

The fact that the multivariate model got the lowest error on 12 sites requires additional analysis. The multivariate experiments are set up so that all time series from a site are modeled jointly, as the piezometers at the same site experience the same internal and external stressors, possibly allowing for informative features from these nearby piezometers. Figure 6.5 shows that the wins were spread across five locations, showing improvements in at most half of the sensors of a single location. This indicates that multivariate modeling can be useful under certain local conditions, but the benefits are not consistent across the entire dataset. This also suggests that while multivariate forecasting is not competitive as a general strategy, it remains relevant in specific settings, and further work could explore which site characteristics drive these gains.

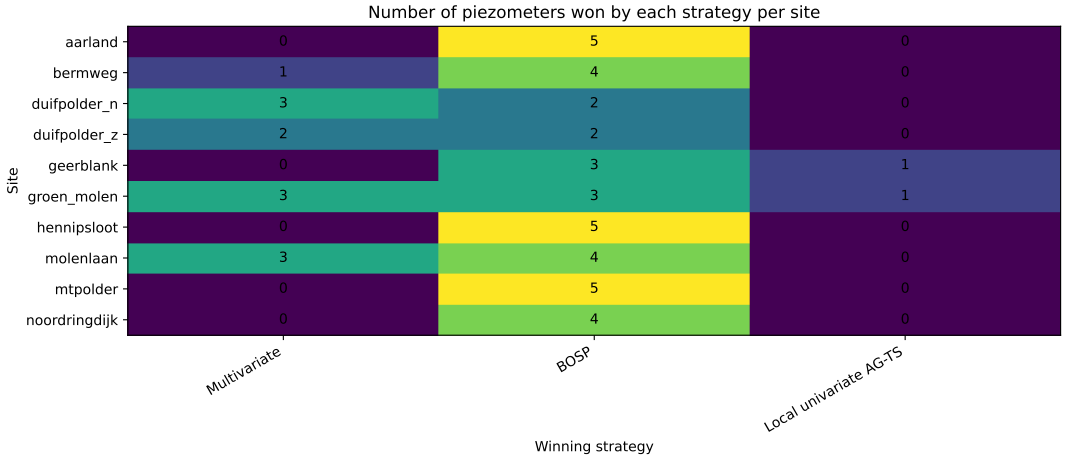


FIGURE 6.5: Total amount of piezometer wins per site for each strategy. Wins are calculated based on MAE. The amount of piezometers varies from 4 to 6 per site, and each piezometer represents a time series.

Additionally, although the multivariate models perform worse in terms of MAE, Table 6.3 shows that they perform better in terms of MAPE. This can occur because MAPE normalizes errors by the magnitude of the series, and several of the piezometers where the multivariate model performs well are those with smaller hydraulic head values. This results in a lower relative error, while the absolute error remains higher.

We can now answer our third research question: *Does incorporating location specific multivariate information improve forecasting accuracy over the best global univariate model?* While the results suggest that there may be specific settings in which a multivariate approach is beneficial, it is generally outperformed by the global univariate approach. After assessing the three modeling approaches (local univariate, global univariate and global multivariate) on our dataset, we found that the global univariate approach performs best. We will use this approach in the final set of experiments, which focuses on peaks in the hydraulic head time series.

6.4 Hydraulic head peaks results

We augment BOSP with a peak predictor that estimates, for each forecast window, the probability of a peak. The predicted probabilities are passed as covariates to AG-TS so the forecaster can condition on peak risk. Because peaks are rare but critical, an effective peak signal can help indicate peak periods.

Table 6.4 summarizes the results of adding our proposed peak predictor (BOSP+Peak) compared to the original BOSP strategy. While the overall MAE slightly increased compared to BOSP (2.642 vs 2.615) the performance at peaks increased notably, with the peak MAE reduced from 9.44 to 7.74. This shows that the BOSP+Peak helps the model adjust to possible peaks, even if it comes at a slight cost in overall accuracy.

TABLE 6.4: Performance comparison with and without peak predictor averaged across all time series

Strategy	Total MAE	Peak MAE
BOSP	**2.615 ± 0.071**	9.44 ± 0.57
BOSP+Peak	2.642 ± 0.062	**7.74 ± 0.46**

Figure 6.6 compares the results at peak moments of BOSP+Peak to the original BOSP approach. BOSP+Peak improves peak performance at 75% of the sites. The fact that the peak predictor rarely results in much worse performance shows that it is informative, but not harmful to the model.

Statistical analysis across all time series supports this conclusion. Peak MAE improved significantly with BOSP+Peak (Wilcoxon signed-rank: $W = 215$, $n = 51$, one-sided $p < 10^{-5}$). In contrast, the change in total MAE was not significant (Wilcoxon signed-rank: $W = 668$, $n = 51$, one-sided $p = 0.42$), confirming that the overall error difference remained statistically indistinguishable from BOSP.

To illustrate the range of outcomes, Figure 6.7 shows differences in forecasts for three sites representing the 25th, 50th and 75th percentile of improvement in peak MAE.

The results answer our final research question: *How can forecasting performance during sudden hydraulic head peaks be improved?* As we have shown, including a dedicated

peak predictor and passing the predicted probability of peaks in the prediction window as covariates improves the performance of the model at peak moments, while keeping overall accuracy mostly the same.

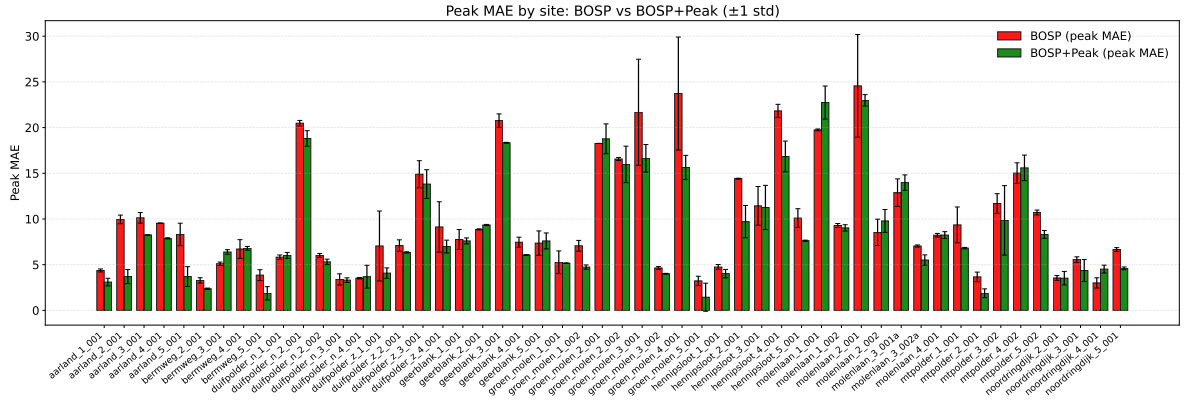


FIGURE 6.6: Comparison of errors for all time series during peak moments of BOSP (red) to BOSP+Peak (green). Bars show the mean MAE across three different seeds; error bars denote $\pm$ 1 standard deviation across seeds. MAE is computed only at timestamps labeled as peaks for each time series; both variants use the same data splits and evaluation protocol.

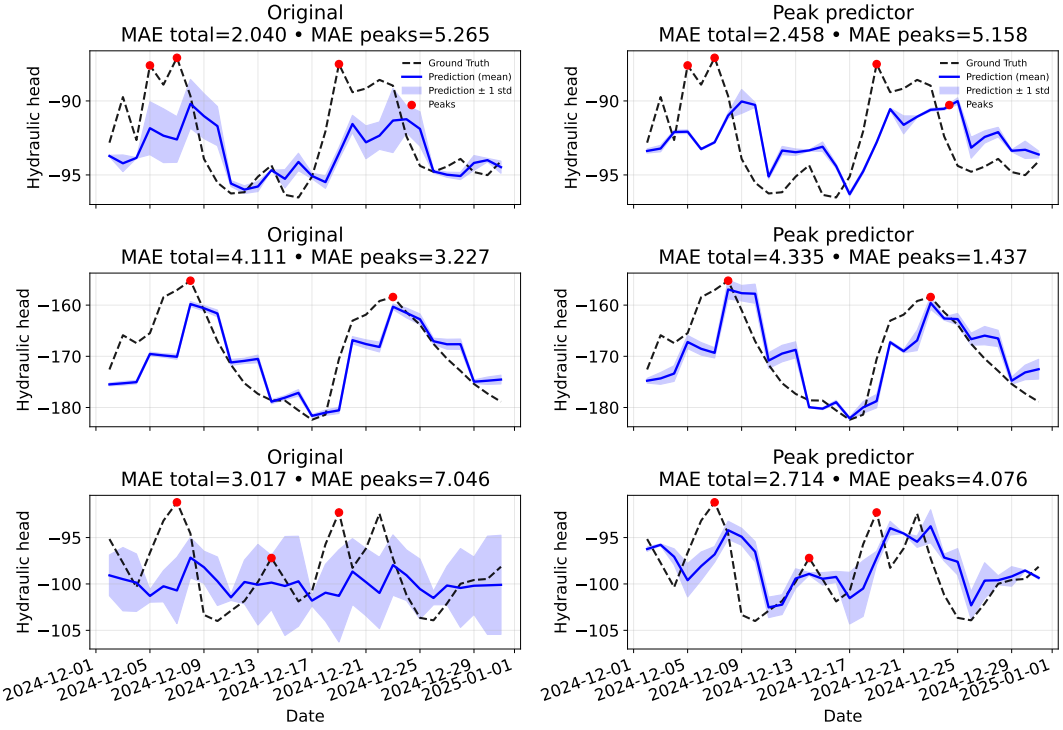


FIGURE 6.7: Comparison of forecast of BOSP (left) to BOSP+Peak (right). In order from top to bottom: (i) groen_molen_1, 25th percentile (ii) groen_molen_5, 50th percentile (iii) duifpolder_z_1, 75th percentile. Note that not all local maxima are marked; peaks are only highlighted if they satisfy the criteria as described in equation 4.6

Chapter 7

Conclusions and future work

In this thesis, we investigated the use of AutoML for hydraulic head forecasting in dikes. Though AutoML techniques have been used for hydraulic head forecasting in wells, this is, to the best of our knowledge, the first application of AutoML in the domain of dikes. This domain is unique due to its layered structures, site-specific boundary conditions and variable response to external and internal stressors, while its short-term forecasting accuracy is directly tied to the dike stability and flood defense safety. The goal was to investigate whether AutoML can provide improvements over traditional forecasting approaches, and to explore extensions to make the AutoML approach more suitable to our domain. Our experiments addressed four research questions, each focusing on a different modeling strategy.

First, we evaluated the performance of AutoGluon-TimeSeries against classical time series forecasting methods, as well as other AutoML frameworks in a local univariate setting. Though the results of AG-TS were not significantly better than the second best performing model (simple autoregressive) the results did show that AG-TS achieved lower error metrics compared to all other strategies, and achieved the lowest error on 34 out of 51 time series. This established AG-TS as a good baseline for further experiments.

Second, we investigated AG-TS in a global univariate setting, where the model learns information across multiple time series. Training the model on all time series in the dataset introduces too much noise to improve performance, so we used several strategies to partition the time series into smaller subsets. The selective pooling strategies we used came from either relevant literature (DTW, feature-based clustering) or domain knowledge (series from the same location). As these strategies were unable to partition the dataset in meaningful subsets, we proposed Bayesian Optimization for Selective Pooling (BOSP). BOSP adaptively searched for informative subsets through Bayesian Optimization, which did result in clear improvements. It achieved the lowest error overall, and consistently outperformed the local univariate baseline across sites. Our results showed that global models can outperform local models, but only if the time series are partitioned into meaningful subsets on which the model can be trained.

Third, we tested whether multivariate models could exploit dependencies between piezometers at the same site. We implement the AG-TS approach with robust model presets and ensembling using the multivariate counterparts of the AG-TS models in DARTS. While the multivariate models achieved lower error on certain time series, they did not outperform the BOSP approach from the previous experiments. Their relative performance in mean absolute percentage error (MAPE) combined with good performance at certain sites does suggest they may be useful for certain

specific situations, especially when the magnitude of the values is smaller. Their general performances maintained inconsistent, showing that multivariate forecasting is not the best solution in this context.

Finally, we developed a peak-aware forecasting strategy to improve predictions during sudden hydraulic head peaks. We introduced a separate model to predict peaks during the time series, and use its predictions as covariates within AG-TS. This reduced peak errors significantly, while only slightly increasing the overall error. This shows that our peak-aware forecasting method is a useful complement to previous approaches.

7.1 Limitations

While we have shown results that could be useful in practice, the work has certain limitations. We evaluated all models using 10 backtesting windows to ensure comparability across experiments. While this design choice provides a consistent basis for comparison, it may not fully capture longer-term variability in model performance. Larger test sets or more extensive backtesting would have provided additional insight into the robustness of the methods.

This extends into the size of our dataset; though we use 51 time series, these come from a limited 10 measuring locations, all located in the same province. Due to the computational cost of AG-TS and BOSP, including more experiments and/or time series was unfortunately not possible.

Additionally, our experiments assumed perfect knowledge of covariates such as precipitation and evaporation. In practice, these values come from weather forecasts that are inherently uncertain. As we did not include uncertainty around these covariates, some questions in this regard remain unanswered.

7.2 Future work

Although this work presents only the first step into including AutoML methods into the domain of hydraulic head forecasting in dikes, we have shown the potential of this direction. Therefore several directions remain open for future research.

In this work, we observed a large improvement in performance through our proposed BOSP method. Though this method showed promise, it is computationally expensive. We see several options to improve its efficiency, for instance through surrogate assisted optimization, different dimensionality reduction techniques or smarter exploration of the search space. As the work presented in this thesis has a large potential for practical application, improving efficiency could make it more attractive for operational use. Another interesting direction regarding BOSP is to investigate whether the informative subsets defined by BOSP stay stable over time, or if they evolve over time. This could provide insight into the temporal dynamics of these time series and the cross-series relationships that are important in the use of global univariate models.

Our peak-aware forecasting approach could also be extended in multiple ways. Our current classifier had to deal with a large class imbalance, as peaks are relatively rare compared to non-peaks. Exploring alternative models or imbalance-aware strategies could improve the peak forecasting performance.

Finally, we assumed perfect knowledge of precipitation and evaporation at prediction time for our covariates. As uncertainty is an important factor of hydraulic head forecasting, further research could focus on including probabilistic forecast or scenario testing to improve these models for practical use in dike monitoring and flood defense.

Together, these directions highlight the potential to further strengthen AutoML-based approaches for hydraulic head forecasting, both in methodological development and in preparing them for practical use in dike monitoring and flood defense management.

7.3 Code and data availability

The code used for all experiments in this thesis, along with preprocessing scripts and sample data is available at github.com/bramvaneerden/AutoML-Hydraulic-Head. The hydraulic head dataset used in this thesis contains measurements until January 1, 2025; an updated version is available through the data portal of TU Delft [61]. Precipitation and evaporation data are available through the Meteobase platform [63].

Appendix A

Dataset overview

This appendix gives a full overview of all time series included in this work. This section is intended to provide additional information on the dataset as described in Chapter 5 Each time series is recorded through a piezometer at a measuring site (e.g. Aarland), and the position of the piezometer on the cross-sectional profile of the dike is indicated by the integer following the site name (e.g. aarland_1). The piezometers are placed in ascending order from closest to the water body to furthest away. Some piezometers contain a (additional) diver that measures the groundwater level at a lower depth, meant for specific stability calculations. These piezometers are indicated with '_002' at the end of the id.

A.1 Distribution of hydraulic head

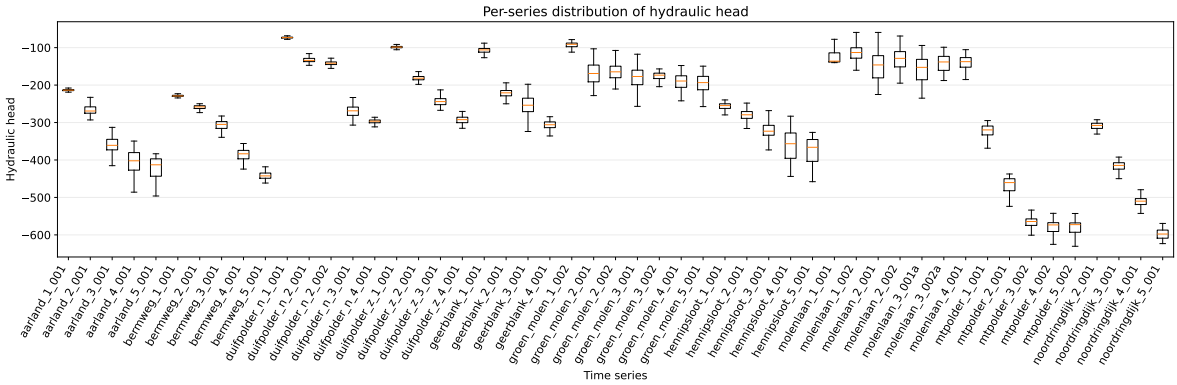


FIGURE A.1: Distribution of hydraulic head per time series, shown through boxplots. It is often the case that piezometer placed closest to the waterbody (e.g. aarland_1) show lower variance and piezometers placed further inland (e.g. aarland_5) show higher variance. Seepage from the water body itself keep the water level relatively stable at nearby piezometers, but piezometers placed further away are influenced more heavily by external stressors.

A.2 Statistics per series

id	date min	date max	mean	variance	min	max	Missing %
aarland_1_001	2020-07-08	2024-12-31	-213.335465	4.973381	-219.342542	-204.005375	0.0
aarland_2_001	2020-07-08	2024-12-31	-265.159148	212.766304	-293.039500	-218.567333	0.0
aarland_3_001	2020-07-08	2024-12-31	-360.415716	421.350907	-415.173667	-312.891625	0.0
aarland_4_001	2020-07-08	2024-12-31	-405.365180	936.619754	-485.915583	-349.507208	0.1
aarland_5_001	2020-07-08	2024-12-31	-421.172355	817.924478	-496.143083	-383.367292	0.0
bermweg_1_001	2020-07-08	2024-12-31	-228.708083	4.413103	-234.750333	-219.685708	0.0
bermweg_2_001	2020-07-08	2024-12-31	-259.101232	24.422168	-274.461708	-249.764542	0.0
bermweg_3_001	2020-07-08	2024-12-31	-306.955125	142.415548	-339.448917	-282.596583	0.0
bermweg_4_001	2020-07-08	2024-12-31	-385.856732	221.333109	-424.298333	-356.002833	0.0
bermweg_5_001	2020-07-08	2024-12-31	-441.482702	85.421827	-461.568333	-418.218000	0.0
duifpolder_n_1_001	2020-07-08	2024-12-31	-73.203981	4.386675	-83.273792	-64.350417	0.0
duifpolder_n_2_001	2020-07-08	2024-12-31	-131.555723	84.508124	-147.236958	-93.179667	0.0
duifpolder_n_2_002	2020-07-08	2024-12-31	-141.819577	29.925816	-161.737458	-122.970875	0.0
duifpolder_n_3_001	2020-07-08	2024-12-31	-270.405178	237.507569	-306.874167	-233.169708	0.0
duifpolder_n_4_001	2020-07-08	2024-12-31	-297.176917	25.552147	-313.454583	-285.919542	0.0
duifpolder_z_1_001	2020-07-08	2024-12-31	-98.533921	9.014223	-109.130750	-85.636750	0.0
duifpolder_z_2_001	2020-07-08	2024-12-31	-181.293835	43.866921	-202.013750	-158.967625	0.0
duifpolder_z_3_001	2020-07-08	2024-12-31	-243.792905	128.812129	-267.363542	-212.811833	0.0
duifpolder_z_4_001	2020-07-08	2024-12-31	-292.940836	89.334139	-315.368958	-270.339208	0.0
geerblank_1_001	2020-07-08	2024-12-31	-107.380242	75.909170	-133.254125	-80.553625	0.0
geerblank_2_001	2020-07-08	2024-12-31	-223.488591	212.392498	-281.862500	-194.004625	0.0
geerblank_3_001	2020-07-08	2024-12-31	-254.539731	684.963291	-325.737708	-197.812500	0.0
geerblank_4_001	2020-07-08	2024-12-31	-307.035310	141.392835	-349.608958	-284.453500	0.0
groen_molen_1_002	2020-07-08	2024-12-31	-93.963190	88.014378	-136.433500	-78.526375	0.0
groen_molen_2_001	2020-07-08	2024-12-31	-168.077153	789.468883	-228.080042	-103.160792	0.0
groen_molen_2_002	2020-07-08	2024-12-31	-164.045595	411.293681	-210.497417	-107.503083	0.0
groen_molen_3_001	2020-07-08	2024-12-31	-179.650399	832.621087	-262.867875	-117.605542	0.0
groen_molen_3_002	2020-07-08	2024-12-31	-176.046787	112.589941	-208.940333	-156.804792	0.1
groen_molen_4_001	2020-07-08	2024-12-31	-191.494376	441.678178	-241.998417	-147.803125	0.0
groen_molen_5_001	2020-07-08	2024-12-31	-196.199767	619.116677	-257.423458	-149.514042	0.0
hennipsloot_1_001	2020-07-08	2024-12-31	-256.477542	68.111757	-284.952417	-239.628500	0.0
hennipsloot_2_001	2020-07-08	2024-12-31	-283.370358	353.767860	-350.272792	-248.109000	0.0
hennipsloot_3_001	2020-07-08	2024-12-31	-322.872221	602.935054	-404.737667	-261.572500	0.0
hennipsloot_4_001	2020-07-08	2024-12-31	-360.838635	1598.840535	-443.873917	-283.081750	0.1
hennipsloot_5_001	2020-07-08	2024-12-31	-375.061039	1146.017540	-457.905625	-326.363292	4.3
molenlaan_1_001	2020-07-08	2024-12-31	-123.476442	420.136394	-140.076500	-42.069125	0.0
molenlaan_1_002	2020-07-08	2024-12-31	-114.229331	359.787286	-160.372875	-53.414083	0.0
molenlaan_2_001	2020-07-08	2024-12-31	-150.105333	1523.985715	-225.153667	-59.407917	0.0
molenlaan_2_002	2020-07-08	2024-12-31	-131.976129	736.793369	-194.847208	-68.984417	0.0
molenlaan_3_001	2020-07-08	2024-12-31	-159.175207	1192.449267	-234.947833	-94.194542	0.0
molenlaan_3_002	2020-07-08	2024-12-31	-141.637604	517.684803	-187.867792	-98.980583	0.0
molenlaan_4_001	2020-07-08	2024-12-31	-140.187631	282.332822	-185.245750	-105.436042	0.0
mtpolder_1_001	2020-07-08	2024-12-31	-323.083018	303.297824	-389.750042	-294.777000	0.0
mtpolder_2_001	2020-07-08	2024-12-31	-465.431308	313.732699	-523.803167	-437.374458	0.0
mtpolder_3_002	2020-07-08	2024-12-31	-569.477592	428.016149	-661.436042	-529.975125	0.0
mtpolder_4_002	2020-07-08	2024-12-31	-580.179098	429.743225	-670.346208	-542.444833	0.0
mtpolder_5_002	2020-07-08	2024-12-31	-580.296347	387.922806	-651.302042	-542.847125	0.0
noordringdijk_2_001	2020-07-08	2024-12-31	-308.799206	72.806281	-330.774708	-292.481375	0.0
noordringdijk_3_001	2020-07-08	2024-12-31	-416.177757	151.839550	-458.688333	-392.309458	0.0
noordringdijk_4_001	2020-07-08	2024-12-31	-514.143224	338.610753	-592.728083	-479.396167	0.0
noordringdijk_5_001	2020-07-08	2024-12-31	-597.926313	155.531258	-623.308583	-569.423750	0.0

TABLE A.1: Statistics for each time series in the dataset. All time series span the same period, but vary in hydraulic head distribution. Missing values are presented as percentage of the total points in the time series.

Appendix B

Per-series model errors

This appendix gives an overview of the model errors of the local univariate, global univariate, and global multivariate models for each time series. Figure B.1 shows the total amount of wins per strategy and Figure B.2 shows the difference between the best error found and the errors produced per strategy. Tables B.1 and B.2 show the complete set of errors for all time series and models. Figure B.3 shows the distribution of the error values through boxplots. These per-series results complement the aggregate results presented in Chapter 6 by highlighting site-level variation and identifying cases where a strategy performs better or worse.

B.1 Wins per strategy

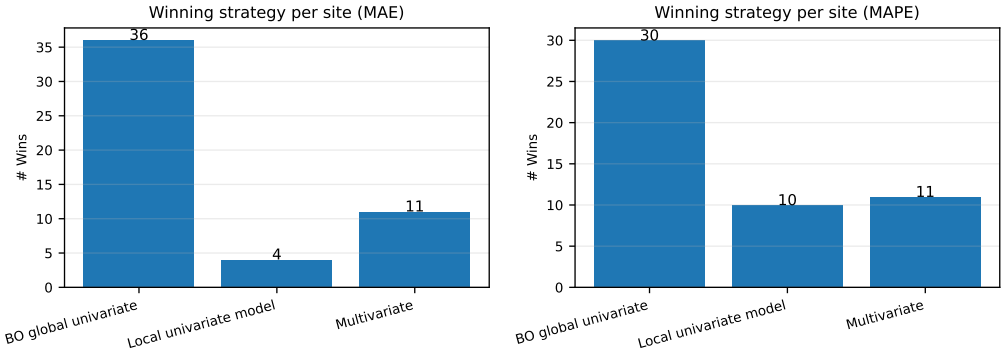


FIGURE B.1: Winning strategy per site for MAE (left) and MAPE (right). Bars show how often each strategy achieves the lowest per-series error across the 51 time series.

B.2 Average error comparison

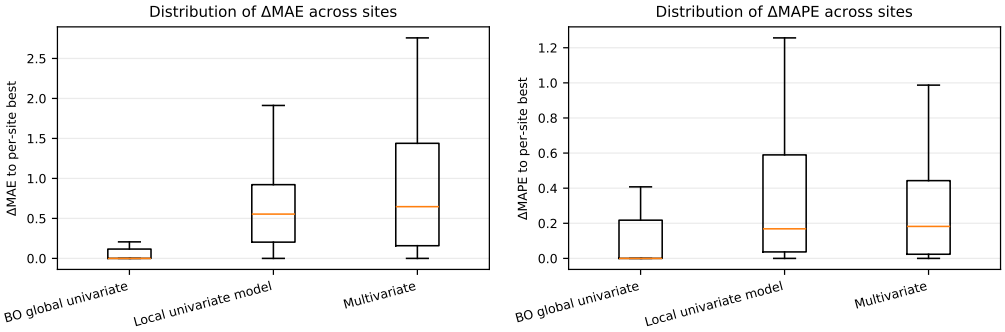


FIGURE B.2: Distributions of Δ to the per-site best for MAE (left) and MAPE (right). For each time series, Δ is computed as the strategy's error minus the best error achieved at that series.

B.3 Per-location MAE

TABLE B.1: Per-location MAE (mean ± std) per strategy, averaged across three seeds. Boldface entries denote the lowest mean error among the three strategies for that series.

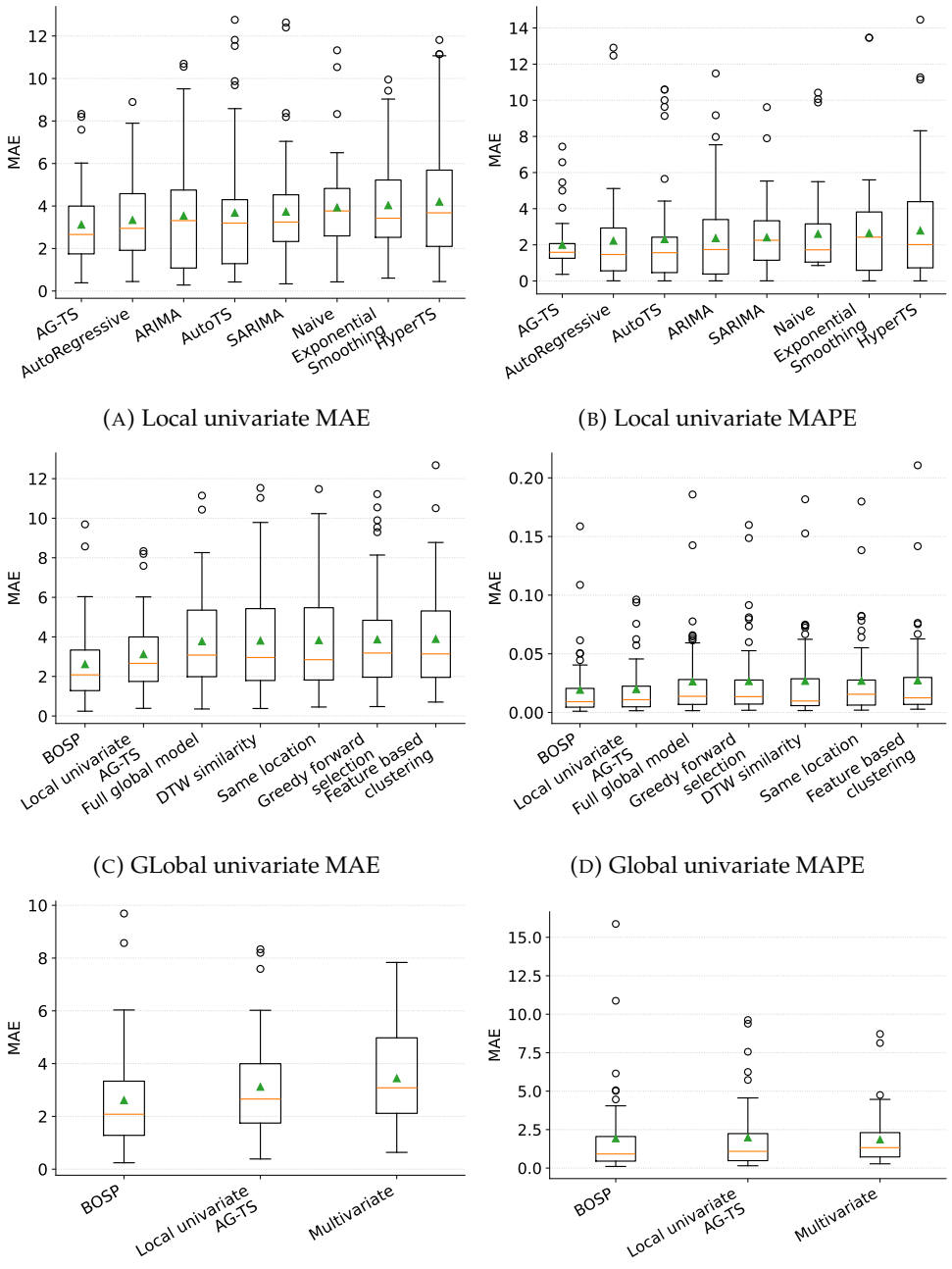
location	BOSP	Local	Multi
aarland_1_001	**1.341 ± 0.008**	1.387 ± 0.287	1.351 ± 0.007
aarland_2_001	**2.130 ± 0.064**	4.042 ± 0.417	7.017 ± 0.542
aarland_3_001	**2.899 ± 0.046**	3.381 ± 0.289	4.212 ± 0.417
aarland_4_001	**2.419 ± 0.009**	2.765 ± 0.562	6.375 ± 0.526
aarland_5_001	**1.140 ± 0.177**	1.776 ± 0.634	3.083 ± 0.602
bermweg_1_001	**0.243 ± 0.013**	0.386 ± 0.140	0.636 ± 0.001
bermweg_2_001	**0.392 ± 0.009**	0.945 ± 0.364	0.871 ± 0.039
bermweg_3_001	**0.827 ± 0.077**	2.264 ± 0.472	1.577 ± 0.036
bermweg_4_001	**1.271 ± 0.188**	1.990 ± 0.708	1.813 ± 0.033
bermweg_5_001	2.081 ± 0.095	2.241 ± 0.656	**1.887 ± 0.281**
duifpolder_n_1_001	1.177 ± 0.022	1.816 ± 0.713	**1.008 ± 0.018**
duifpolder_n_2_001	4.791 ± 0.226	4.952 ± 0.293	**2.494 ± 0.043**
duifpolder_n_2_002	2.742 ± 0.129	3.024 ± 0.696	**1.917 ± 0.038**
duifpolder_n_3_001	**1.382 ± 0.193**	2.949 ± 1.168	2.557 ± 0.194
duifpolder_n_4_001	**0.762 ± 0.036**	0.923 ± 0.375	0.901 ± 0.024
duifpolder_z_1_001	2.295 ± 0.016	2.858 ± 1.020	**2.291 ± 0.038**
duifpolder_z_2_001	**2.793 ± 0.171**	3.565 ± 1.002	3.198 ± 0.008
duifpolder_z_3_001	2.622 ± 0.030	2.660 ± 0.228	**2.415 ± 0.036**
duifpolder_z_4_001	**1.588 ± 0.004**	1.727 ± 0.526	2.097 ± 0.118
geerblank_1_001	**2.017 ± 0.064**	2.725 ± 1.214	2.494 ± 0.086
geerblank_2_001	**1.231 ± 0.036**	1.923 ± 1.437	2.422 ± 0.242
geerblank_3_001	3.768 ± 0.240	**2.391 ± 1.197**	5.148 ± 0.072
geerblank_4_001	**1.493 ± 0.059**	1.648 ± 0.232	2.610 ± 0.372
groen_molen_1_002	**1.761 ± 0.226**	1.991 ± 0.386	2.297 ± 0.107
groen_molen_2_001	6.033 ± 0.313	**4.236 ± 0.592**	4.518 ± 0.233
groen_molen_2_002	5.892 ± 0.089	6.014 ± 0.218	**4.852 ± 0.308**
groen_molen_3_001	5.285 ± 0.285	5.458 ± 0.430	**5.115 ± 0.140**
groen_molen_3_002	**1.422 ± 0.032**	1.619 ± 1.026	1.599 ± 0.070
groen_molen_4_001	**4.366 ± 0.127**	8.335 ± 1.821	5.500 ± 0.142
groen_molen_5_001	**3.275 ± 0.088**	3.391 ± 0.504	3.922 ± 0.256
hennipsloot_1_001	**1.257 ± 0.094**	1.613 ± 0.202	1.711 ± 0.201
hennipsloot_2_001	**2.221 ± 0.023**	3.330 ± 1.080	3.623 ± 0.180
hennipsloot_3_001	**4.312 ± 0.133**	4.455 ± 1.440	5.436 ± 0.271
hennipsloot_4_001	4.565 ± 0.701	**4.267 ± 1.190**	5.803 ± 0.059
hennipsloot_5_001	**1.293 ± 0.151**	1.715 ± 0.520	5.846 ± 0.565
molenlaan_1_001	9.689 ± 0.552	**6.021 ± 0.443**	6.688 ± 0.242
molenlaan_1_002	4.326 ± 0.037	5.460 ± 1.116	**4.191 ± 0.153**
molenlaan_2_001	8.570 ± 0.249	8.199 ± 1.259	**7.835 ± 0.325**
molenlaan_2_002	3.600 ± 0.357	5.074 ± 1.336	**3.502 ± 0.394**
molenlaan_3_001a	**3.263 ± 0.234**	5.696 ± 0.330	3.826 ± 1.186
molenlaan_3_002a	**1.216 ± 0.026**	1.768 ± 0.651	3.589 ± 0.591
molenlaan_4_001	**1.665 ± 0.207**	2.638 ± 0.227	2.134 ± 0.033
mtpolder_1_001	**1.824 ± 0.262**	2.034 ± 0.340	2.523 ± 0.047
mtpolder_2_001	**1.182 ± 0.008**	1.379 ± 0.276	3.610 ± 0.096
mtpolder_3_002	**3.197 ± 0.075**	3.892 ± 1.034	5.102 ± 0.438
mtpolder_4_002	**2.144 ± 0.139**	3.950 ± 0.197	3.620 ± 0.083
mtpolder_5_002	**3.395 ± 0.169**	7.591 ± 0.739	3.809 ± 0.130
noordringdijk_2_001	**0.601 ± 0.029**	1.471 ± 0.287	1.308 ± 0.200
noordringdijk_3_001	**1.330 ± 0.081**	1.954 ± 0.774	3.016 ± 0.749
noordringdijk_4_001	**0.914 ± 0.050**	1.374 ± 0.596	6.039 ± 0.073
noordringdijk_5_001	**1.298 ± 0.009**	2.566 ± 0.241	2.403 ± 0.217

B.4 Per-location MAPE

TABLE B.2: Per-location MAPE (mean ± std) per strategy, averaged across three seeds. Boldface entries denote the lowest mean error among the three strategies for that series.

location	BOSP	Local	Multi
aarland_1_001	0.634 ± 0.004	**0.626 ± 0.523**	0.637 ± 0.003
aarland_2_001	**1.027 ± 0.030**	2.143 ± 0.345	2.976 ± 0.220
aarland_3_001	**0.967 ± 0.015**	1.024 ± 0.327	1.268 ± 0.129
aarland_4_001	0.706 ± 0.032	**0.648 ± 0.260**	1.635 ± 0.136
aarland_5_001	**0.320 ± 0.046**	0.443 ± 0.166	0.768 ± 0.152
bermweg_1_001	**0.106 ± 0.006**	0.151 ± 0.061	0.278 ± 0.000
bermweg_2_001	**0.154 ± 0.019**	0.318 ± 0.144	0.336 ± 0.015
bermweg_3_001	**0.316 ± 0.026**	0.500 ± 0.239	0.515 ± 0.011
bermweg_4_001	**0.374 ± 0.051**	0.543 ± 0.105	0.473 ± 0.008
bermweg_5_001	0.486 ± 0.023	**0.360 ± 0.154**	0.434 ± 0.065
duifpolder_n_1_001	1.772 ± 0.032	2.116 ± 0.197	**1.364 ± 0.025**
duifpolder_n_2_001	4.026 ± 0.153	3.440 ± 0.249	**1.921 ± 0.036**
duifpolder_n_2_002	1.952 ± 0.092	2.166 ± 0.327	**1.365 ± 0.026**
duifpolder_n_3_001	**0.568 ± 0.074**	1.088 ± 0.820	0.962 ± 0.073
duifpolder_n_4_001	**0.261 ± 0.013**	0.282 ± 0.199	0.303 ± 0.008
duifpolder_z_1_001	2.333 ± 0.128	2.915 ± 0.091	**2.322 ± 0.035**
duifpolder_z_2_001	**1.638 ± 0.103**	2.223 ± 0.151	1.854 ± 0.007
duifpolder_z_3_001	1.254 ± 0.015	**0.953 ± 0.288**	1.036 ± 0.019
duifpolder_z_4_001	0.589 ± 0.001	**0.464 ± 0.192**	0.715 ± 0.039
geerblank_1_001	**2.058 ± 0.046**	2.515 ± 0.229	2.211 ± 0.075
geerblank_2_001	**0.662 ± 0.016**	0.906 ± 0.625	1.122 ± 0.106
geerblank_3_001	1.849 ± 0.117	**1.569 ± 0.607**	2.289 ± 0.034
geerblank_4_001	0.549 ± 0.020	**0.393 ± 0.182**	0.852 ± 0.122
groen_molen_1_002	**2.047 ± 0.221**	2.166 ± 0.247	2.533 ± 0.114
groen_molen_2_001	5.061 ± 0.361	3.522 ± 0.475	**3.296 ± 0.157**
groen_molen_2_002	5.014 ± 0.096	4.559 ± 0.409	**3.644 ± 0.207**
groen_molen_3_001	4.046 ± 0.207	3.918 ± 0.260	**3.335 ± 0.100**
groen_molen_3_002	**0.919 ± 0.019**	0.977 ± 0.323	0.921 ± 0.039
groen_molen_4_001	**2.761 ± 0.066**	5.727 ± 0.522	3.064 ± 0.078
groen_molen_5_001	2.087 ± 0.058	**1.938 ± 0.350**	2.080 ± 0.132
hennipsloot_1_001	**0.510 ± 0.106**	0.574 ± 0.367	0.687 ± 0.082
hennipsloot_2_001	**0.888 ± 0.008**	1.282 ± 0.825	1.326 ± 0.066
hennipsloot_3_001	1.587 ± 0.054	**1.187 ± 0.068**	1.811 ± 0.084
hennipsloot_4_001	**1.485 ± 0.231**	1.493 ± 0.922	1.780 ± 0.013
hennipsloot_5_001	**0.412 ± 0.044**	0.450 ± 0.306	1.693 ± 0.164
molenlaan_1_001	15.869 ± 0.571	9.625 ± 0.308	**8.716 ± 0.213**
molenlaan_1_002	6.150 ± 0.040	7.560 ± 0.290	**4.749 ± 0.177**
molenlaan_2_001	10.874 ± 0.332	9.384 ± 0.132	**8.128 ± 0.187**
molenlaan_2_002	4.460 ± 0.407	6.248 ± 0.665	**3.507 ± 0.408**
molenlaan_3_001a	**3.178 ± 0.200**	3.703 ± 0.961	4.465 ± 0.203
molenlaan_3_002a	**1.247 ± 0.032**	1.680 ± 1.026	3.014 ± 0.481
molenlaan_4_001	**1.509 ± 0.184**	2.259 ± 0.935	1.668 ± 0.031
mtpolder_1_001	**0.611 ± 0.083**	0.648 ± 0.498	0.768 ± 0.013
mtpolder_2_001	0.282 ± 0.002	**0.260 ± 0.193**	0.780 ± 0.021
mtpolder_3_002	**0.629 ± 0.014**	0.881 ± 0.675	0.897 ± 0.074
mtpolder_4_002	**0.418 ± 0.024**	0.704 ± 0.434	0.630 ± 0.014
mtpolder_5_002	**0.627 ± 0.029**	1.274 ± 0.664	0.664 ± 0.022
noordringdijk_2_001	**0.232 ± 0.010**	0.360 ± 0.164	0.425 ± 0.064
noordringdijk_3_001	**0.334 ± 0.020**	0.465 ± 0.295	0.735 ± 0.184
noordringdijk_4_001	**0.200 ± 0.010**	0.252 ± 0.237	1.169 ± 0.014
noordringdijk_5_001	**0.241 ± 0.002**	0.408 ± 0.233	0.408 ± 0.037

B.5 Optimization progress



(A) Local univariate MAE

(B) Local univariate MAPE

(C) GLobal univariate MAE

(D) Global univariate MAPE

(E) Multivariate MAE

(F) Multivariate MAPE

FIGURE B.3: Boxplots of the results for the local univariate, global univariate and multivariate experiments. For each experiment, a plot is shown for MAE and MAPE. Models are ordered from left to right based on the mean score across all time series.

Appendix C

BOSP implementation details

This appendix provides implementation details of the BOSP method as proposed in Chapter 4, as well as optimization dynamics and concrete subset examples for the Bayesian Optimization for Selective Pooling (BOSP) results as described in Chapter 6. The implementation settings are shown for reproducibility and the examples are intended to provide additional insight into how BOSP selects informative training subsets. The first section outlines the hyperparameters used in the BOSP experiments. The second section illustrates the optimization progress of eight time series from the dataset, and the third section provides information on the size of the found subset and the time series included in it.

C.1 Hyperparameters

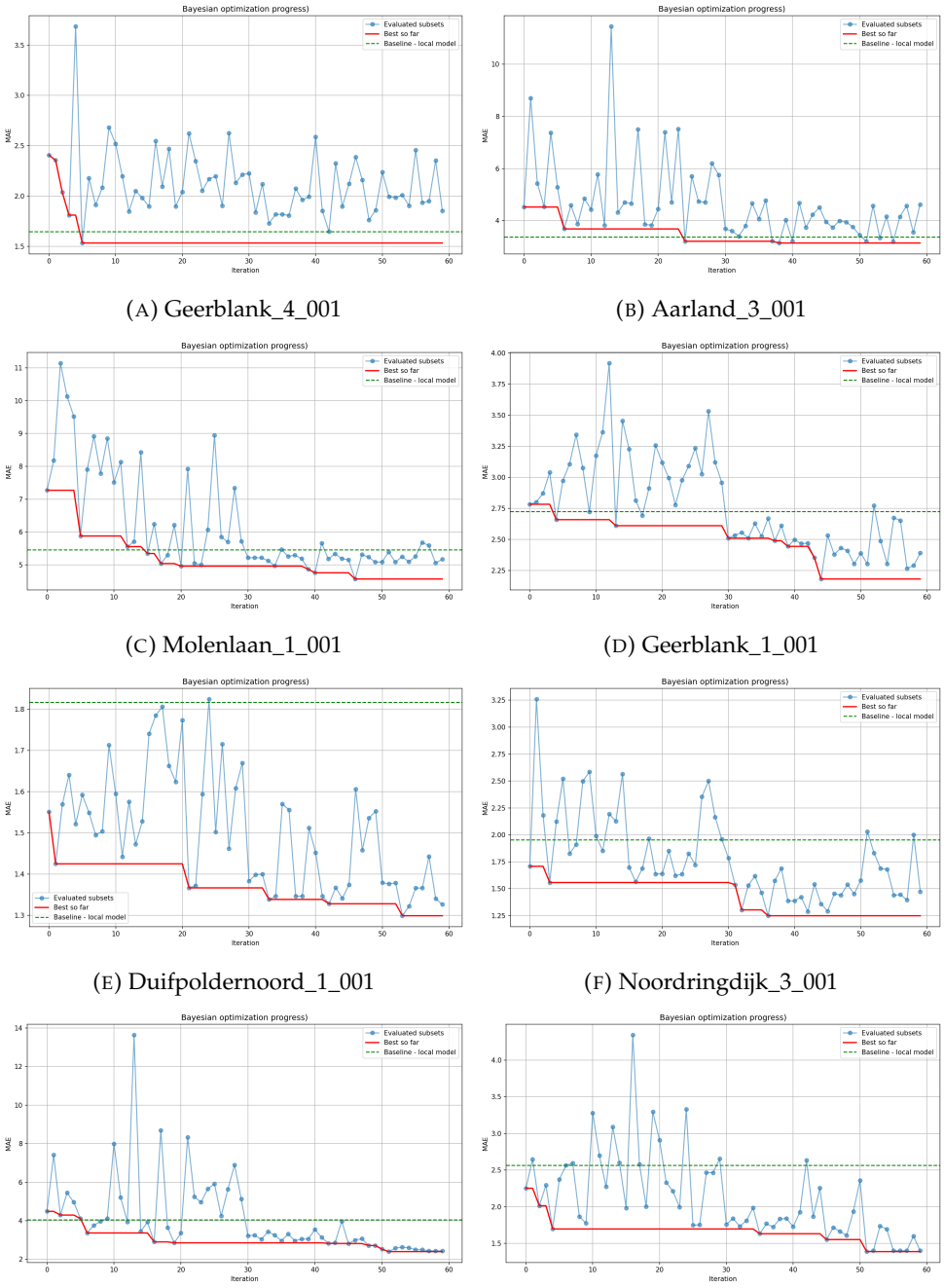
Table C.1 summarizes the fixed hyperparameters used in all BOSP runs.

TABLE C.1: BOSP implementation settings.

Component	**Setting / Value**
Initial random subsets (n_{init})	30
Max BO iterations	30
Prediction horizon (H)	3 days
Validation windows	10 rolling windows
Objective	Mean MAE over validation windows
Response transform	$\log(\text{MAE})$ for GP fit
Kernel	Constant $\times$ Matern(ν=2.5) with length scaling + White noise
Acquisition	Expected Improvement (EI), ξ=0.01
Feature basis	TSFresh $\rightarrow$ Standardize $\rightarrow$ PCA (19 comps)
Subset embedding	mean(PCA), var(PCA), min-dist-to-target, subset size

C.2 Optimization progress

Figure C.1 illustrates the optimization progress for the target time series. Each plot shows the MAE of the evaluated subsets per iteration, the best-so-far subset, and the local univariate AG-TS baseline.



(A) Geerblank_4_001

(B) Aarland_3_001

(C) Molenlaan_1_001

(D) Geerblank_1_001

(E) Duifpoldernoord_1_001

(F) Noordringdijk_3_001

(G) Aarland_2_001

(H) Noordringdijk_5_001

FIGURE C.1: Bayesian optimization progress for 8 piezometers. Blue markers denote evaluated subsets, the red line the best subset found so far, and the dashed green line the baseline performance of the local univariate model.

C.3 Subset composition

Table C.2 reports the final BOSP-selected subsets for the set of representative targets, alongside baseline and final MAE.

Target	Subset size	Baseline MAE	BOSP MAE	% improv.	Selected series
geerblank_4_001	2	1.65	1.53	6.9%	groen_molen_3_002, molenlaan_4_001
aarland_3_001	6	3.38	3.05	9.7%	aarland_4_001, duifpolder_z_4_001, hennipsloot_4_001, molenlaan_1_001, molenlaan_3_002a, molenlaan_4_001
molenlaan_1_002	18	5.46	4.49	17.8%	bermweg_3_001, bermweg_5_001, duifpolder_n_1_001, duifpolder_n_2_002, duifpolder_z_3_001, duifpolder_z_4_001, geerblank_1_001, groen_molen_3_002, …
geerblank_1_001	26	2.72	2.05	24.6%	aarland_3_001, aarland_4_001, bermweg_2_001, bermweg_4_001, duifpolder_n_2_001, duifpolder_n_2_002, duifpolder_n_4_001, duifpolder_z_1_001, …
duifpolder_n_1_001	6	1.82	1.26	30.9%	aarland_1_001, duifpolder_n_2_001, geerblank_1_001, groen_molen_1_002, molenlaan_1_001, mtpolder_5_002
noordringdijk_3_001	32	1.95	1.25	36.1%	aarland_2_001, aarland_3_001, aarland_4_001, bermweg_2_001, bermweg_3_001, bermweg_4_001, duifpolder_n_2_001, duifpolder_n_3_001, …
aarland_2_001	5	4.04	2.27	44.0%	bermweg_5_001, duifpolder_z_1_001, geerblank_3_001, groen_molen_4_001, noordringdijk_3_001
noordringdijk_5_001	5	2.57	1.39	45.9%	aarland_3_001, bermweg_4_001, geerblank_2_001, geerblank_4_001, noordringdijk_4_001

TABLE C.2: Time series contained in the best found subset for eight target series. For each subset the corresponding BOSP MAE is compared to the local univariate model, as well as the percentage improvement.

Appendix D

Peak predictor

This appendix provides implementation details of the peak predictor as described in Chapter 4 and the aggregated results of the peak predictor in Chapter 6. The hyperparameters of the XGBoost model were found using gridsearch and the results are shown in Table D.1. Grid search was performed over 20 random target time series, and the best subsets found by BOSP were used to train the BOSP+Peak model. The model was trained on 80% and evaluated on 20% of the data. Table D.2 shows the top 10 best performing configurations of the grid search. Table D.3 shows the MAE at peak moments for all time series for the BOSP and BOSP+Peak models.

D.1 BOSP+Peak hyperparameters

TABLE D.1: BOSP+Peak XGBoost hyperparameters.

Component	**Setting / Value**
n_estimators	200
max_depth	4
learning_rate	0.05
subsample	0.8
colsample_bytree	0.8

D.2 Best performing configs

TABLE D.2: Ranking of the top 10 XGBoost hyperparameters across targets. Values are mean±std across targets; AP is average precision (AUPRC).

config_id	xgb_config	wins	AUC_{macro}	AP_{macro}	AUC_{h1}	AUC_{h2}	AUC_{h3}
1	n=200, d=4, lr=0.05, ss=0.8, cs=0.8	9	0.731 ± 0.028	0.192 ± 0.025	0.879 ± 0.017	0.715 ± 0.034	0.598 ± 0.046
2	n=200, d=4, lr=0.05, ss=0.8, cs=1.0	6	0.730 ± 0.027	0.192 ± 0.023	0.879 ± 0.018	0.715 ± 0.034	0.595 ± 0.044
8	n=200, d=6, lr=0.05, ss=0.8, cs=1.0	0	0.726 ± 0.030	0.189 ± 0.028	0.876 ± 0.018	0.707 ± 0.037	0.594 ± 0.048
7	n=200, d=6, lr=0.05, ss=0.8, cs=0.8	2	0.725 ± 0.031	0.189 ± 0.029	0.876 ± 0.018	0.707 ± 0.038	0.593 ± 0.048
13	n=400, d=4, lr=0.05, ss=0.8, cs=0.8	0	0.724 ± 0.028	0.189 ± 0.030	0.874 ± 0.018	0.705 ± 0.035	0.593 ± 0.046
14	n=400, d=4, lr=0.05, ss=0.8, cs=1.0	0	0.723 ± 0.028	0.189 ± 0.027	0.874 ± 0.019	0.706 ± 0.035	0.590 ± 0.044
20	n=400, d=6, lr=0.05, ss=0.8, cs=1.0	0	0.721 ± 0.032	0.187 ± 0.032	0.873 ± 0.020	0.701 ± 0.039	0.591 ± 0.050
3	n=200, d=4, lr=0.1, ss=0.8, cs=0.8	0	0.721 ± 0.029	0.187 ± 0.029	0.872 ± 0.018	0.701 ± 0.037	0.591 ± 0.045
19	n=400, d=6, lr=0.05, ss=0.8, cs=0.8	0	0.721 ± 0.032	0.186 ± 0.031	0.873 ± 0.020	0.700 ± 0.039	0.590 ± 0.049
4	n=200, d=4, lr=0.1, ss=0.8, cs=1.0	0	0.720 ± 0.028	0.186 ± 0.028	0.872 ± 0.019	0.700 ± 0.036	0.589 ± 0.044
25	n=600, d=4, lr=0.05, ss=0.8, cs=0.8	0	0.720 ± 0.029	0.186 ± 0.032	0.871 ± 0.019	0.699 ± 0.036	0.590 ± 0.046
32	n=600, d=6, lr=0.05, ss=0.8, cs=1.0	1	0.720 ± 0.032	0.186 ± 0.032	0.871 ± 0.020	0.698 ± 0.040	0.590 ± 0.050

D.3 Performance at peaks per series

TABLE D.3: Overview of the MAE for BOSP and BOSP+Peak at peak moments for all time series in the dataset. Values are shown as mean±std over three runs. Boldface entries represent the model with the lowest error at a time series.

site_id	BOSP	BOSP+Peak	Improvement
aarland_1_001	4.370±0.161	**3.096±0.418**	29.2%
aarland_2_001	9.954±0.472	**3.701±0.770**	62.8%
aarland_3_001	10.127±0.575	**8.247±0.006**	18.6%
aarland_4_001	9.534±0.028	**7.881±0.063**	17.3%
aarland_5_001	8.309±1.228	**3.698±1.084**	55.5%
bermweg_2_001	3.280±0.291	**2.368±0.073**	27.8%
bermweg_3_001	**5.102±0.183**	6.385±0.258	-25.2%
bermweg_4_001	**6.711±1.026**	6.793±0.206	-1.2%
bermweg_5_001	3.855±0.595	**1.869±0.729**	51.5%
duifpolder_n_1_001	**5.822±0.240**	6.017±0.322	-3.3%
duifpolder_n_2_001	20.482±0.300	**18.807±0.853**	8.2%
duifpolder_n_2_002	6.017±0.209	**5.303±0.299**	11.9%
duifpolder_n_3_001	3.391±0.609	**3.309±0.252**	2.4%
duifpolder_n_4_001	**3.522±0.081**	3.686±1.247	-4.7%
duifpolder_z_1_001	7.046±3.825	**4.076±0.571**	42.1%
duifpolder_z_2_001	7.093±0.625	**6.344±0.071**	10.6%
duifpolder_z_3_001	14.895±1.481	**13.817±1.572**	7.2%
duifpolder_z_4_001	9.127±2.754	**6.982±0.702**	23.5%
geerblank_1_001	7.757±1.088	**7.598±0.327**	2.0%
geerblank_2_001	**8.866±0.054**	9.340±0.065	-5.3%
geerblank_3_001	20.762±0.736	**18.341±0.046**	11.7%
geerblank_4_001	7.456±0.553	**6.062±0.022**	18.7%
geerblank_5_001	**7.358±1.336**	7.595±0.874	-3.2%
groen_molen_1_001	5.265±1.242	**5.158±0.039**	2.0%
groen_molen_1_002	7.065±0.590	**4.733±0.234**	33.0%
groen_molen_2_001	**18.264±0.010**	18.769±1.636	-2.8%
groen_molen_2_002	16.562±0.171	**15.970±1.996**	3.6%
groen_molen_3_001	21.674±5.805	**16.635±1.509**	23.2%
groen_molen_3_002	4.640±0.148	**4.006±0.046**	13.7%
groen_molen_4_001	23.725±6.181	**15.639±1.322**	34.1%
groen_molen_5_001	3.227±0.509	**1.437±1.536**	55.5%
hennipsloot_1_001	4.741±0.273	**4.023±0.456**	15.2%
hennipsloot_2_001	14.412±0.047	**9.708±1.767**	32.6%
hennipsloot_3_001	11.436±2.116	**11.258±2.420**	1.6%
hennipsloot_4_001	21.825±0.720	**16.836±1.687**	22.9%
hennipsloot_5_001	10.098±1.026	**7.622±0.063**	24.5%
molenlaan_1_001	**19.746±0.100**	22.739±1.809	-15.2%
molenlaan_1_002	9.285±0.220	**9.022±0.349**	2.8%
molenlaan_2_001	24.562±5.621	**22.979±0.625**	6.4%
molenlaan_2_002	**8.536±1.439**	9.786±1.252	-14.6%
molenlaan_3_001a	**12.882±1.504**	13.978±0.846	-8.5%
molenlaan_3_002a	7.043±0.116	**5.511±0.561**	21.8%
molenlaan_4_001	**8.221±0.199**	8.239±0.372	-0.2%
mtpolder_1_001	9.351±1.961	**6.811±0.079**	27.2%
mtpolder_2_001	3.664±0.522	**1.871±0.481**	48.9%
mtpolder_3_002	11.692±1.076	**9.850±3.799**	15.8%
mtpolder_4_002	**15.017±1.128**	15.591±1.399	-3.8%
mtpolder_5_002	10.711±0.264	**8.313±0.431**	22.4%
noordringdijk_2_001	3.546±0.262	**3.522±0.740**	0.7%
noordringdijk_3_001	5.550±0.309	**4.360±1.210**	21.4%
noordringdijk_4_001	**3.007±0.560**	4.516±0.430	-50.2%
noordringdijk_5_001	6.658±0.208	**4.602±0.146**	30.9%

Bibliography

[1] S. Jevrejeva, L. Jackson, A. Grinsted, D. Lincke, and B. Marzeion, "Flood damage costs under the sea level rise with warming of 1.5 °c and 2 °c," *Environmental Research Letters*, vol. 13, pp. 4–14, 2018. DOI: 10.1088/1748-9326/aacc76.

[2] Netherlands Environmental Assessment Agency (PBL), *Correction wording: Flood risks for the netherlands in ipcc report*, https://www.pbl.nl/en/correction-wording-flood-risks-for-the-netherlands-in-ipcc-report, 2010. (visited on 08/24/2025).

[3] B. Strijker, T. Heimovaara, S. Jonkman, and M. Kok, "Exploring subsurface water conditions in dutch canal dikes during drought periods: Insights from multiyear monitoring," *Water Resources Research*, vol. 60, e2023WR036046, 2024. DOI: 10.1029/2023WR036046.

[4] Hoogwaterbeschermingsprogramma, "Kengetallen hwbp: Hwbp in cijfers (stand per 31 december 2024)," HWBP, 2025. [Online]. Available: https://www.hwbp.nl/projecten/kengetallen-hwbp (visited on 07/11/2025).

[5] I. E. Özer, M. van Damme, and S. N. Jonkman, "International levee performance database: A global, open dataset and insights on levee failures," *Water*, vol. 12, 119, 2020. DOI: 10.3390/w12010119.

[6] W. J. Klerk, F. den Heijer, and T. Schweckendiek, "Value of information in lifecycle management of flood defences," in *Safety and Reliability of Complex Engineered Systems*, 2015, pp. 931–938. DOI: 10.1201/b19094-125.

[7] J.-W. Nieuwenhuis, M. van der Meer, S. Bakkenist, Y. Pluijmers, R. Clemens, and W. Zomer, "Livedijk xl noorderzijlvest: State of the art 2015," FloodControl IJkdijk, 2016. [Online]. Available: https://edepot.wur.nl/473074.

[8] B. Strijker and M. Kok, "The dynamics of peak head responses at dutch canal dikes and the impact of climate change," *Natural Hazards and Earth System Sciences*, vol. 25, pp. 3355–3379, 2025. DOI: 10.5194/nhess-25-3355-2025.

[9] A. Singh, S. Patel, V. Bhadani, V. Kumar, and K. Gaurav, "Automl-gwl: Automated machine learning model for the prediction of groundwater level," *Engineering Applications of Artificial Intelligence*, vol. 127, 107405, 2024. DOI: 10.1016/j.engappai.2023.107405.

[10] R. A. Collenteur, E. Haaf, M. Bakker, T. Liesch, A. Wunsch, J. Soonthornrangsan, J. White, N. Martin, R. Hugman, E. de Sousa, D. Vanden Berghe, X. Fan, T. J. Peterson, J. Bikše, A. Di Ciacca, X. Wang, Y. Zheng, M. Nölscher, J. Koch, R. Schneider, N. Benavides Höglund, S. Krishna Reddy Chidepudi, A. Henriot, N. Massei, A. Jardani, M. G. Rudolph, A. Rouhani, J. J. Gómez-Hernández, S. Jomaa, A. Pölz, T. Franken, M. Behbooei, J. Lin, and R. Meysami, "Data-driven modelling of hydraulic-head time series: Results and lessons learned from the 2022 groundwater time series modelling challenge," *Hydrology and Earth System Sciences*, vol. 28, pp. 5193–5208, 2024. DOI: 10.5194/hess-28-5193-2024.

[11] K. Benidis, S. Rangapuram, V. Flunkert, Y. Wang, D. Maddix, A. C. Türkmen, J. Gasthaus, M. Bohlke-Schneider, D. Salinas, L. Stella, F.-X. Aubet, L. Callot, and T. Januschowski, "Deep learning for time series forecasting: Tutorial and

literature survey," *ACM Computing Surveys*, vol. 55, pp. 1–36, 2022. DOI: 10.1145/3533382.

[12] X. Kong, Z. Chen, W. Liu, K. Ning, L. Zhang, S. Muhammad Marier, Y. Liu, Y. Chen, and F. Xia, "Deep learning for time series forecasting: A survey," *International Journal of Machine Learning and Cybernetics*, vol. 16, pp. 5079–5112, 2025. DOI: 10.1007/s13042-025-02560-w.

[13] V. Nourani, A. A. Mogaddam, and A. O. Nadiri, "An ANN-based model for spatiotemporal groundwater level forecasting," *Hydrological Processes*, vol. 22, pp. 5054–5066, 2008. DOI: 10.1002/hyp.7129.

[14] M. B. Dahl, T. N. Vilhelmsen, T. Bach, and T. M. Hansen, "Hydraulic head change predictions in groundwater models using a probabilistic neural network," *Frontiers in Water*, vol. 5, 1028922, 2023. DOI: 10.3389/frwa.2023.1028922.

[15] A. Wunsch, T. Liesch, and S. Broda, "Groundwater level forecasting with artificial neural networks: A comparison of LSTM, CNNs, and NARX," *Hydrology and Earth System Sciences*, vol. 25, pp. 1671–1687, 2021. DOI: 10.5194/hess-25-1671-2021.

[16] O. Shchur, A. C. Türkmen, N. Erickson, H. Shen, A. Shirkov, T. Hu, and B. Wang, "AutoGluon–TimeSeries: Automl for probabilistic time series forecasting," in *Proceedings of the Second International Conference on Automated Machine Learning*, vol. 224, 2023, pp. 9–21.

[17] P. J. Pereira, N. Costa, P. Mestre, and P. Cortez, "A benchmark of automated multivariate time series forecasting tools for smart cities," in *Progress in Artificial Intelligence: 23rd EPIA Conference on Artificial Intelligence (EPIA 2024), Proceedings, Part III*, 2024, pp. 139–150. DOI: 10.1007/978-3-031-73503-5_12.

[18] M. Ait Al, S. Achchab, and Y. Lahrichi, "Automl driven LSTM and stock price prediction effectiveness: A new frontier for volatile stock markets," in *Proceedings of the 2nd GCC International Conference on Industrial Engineering and Operations Management*, 2024, pp. 640–647. DOI: 10.46254/GC02.20240125.

[19] F. Conrad, M. Mälzer, F. Lange, H. Wiemer, and S. Ihlenfeldt, "Automl applied to time series analysis tasks in production engineering," *Procedia Computer Science*, vol. 232, pp. 849–860, 2024. DOI: 10.1016/j.procs.2024.01.085.

[20] N. Koutantos, M. Fotopoulou, and D. Rakopoulos, "Automated machine learning for optimized load forecasting and economic impact in the greek wholesale energy market," *Applied Sciences*, vol. 14, 9766, 2024. DOI: 10.3390/app14219766.

[21] G. E. P. Box and G. M. Jenkins, *Time Series Analysis: Forecasting and Control*. Holden-Day, 1970, ISBN: 9780816210947.

[22] A.-A. Semenoglou, E. Spiliotis, S. Makridakis, and V. Assimakopoulos, "Investigating the accuracy of cross-learning time series forecasting methods," *International Journal of Forecasting*, vol. 37, pp. 1072–1084, 2021. DOI: 10.1016/j.ijforecast.2020.11.009.

[23] H. Hewamalage, C. Bergmeir, and K. Bandara, "Global models for time series forecasting: A simulation study," *Pattern Recognition*, vol. 124, 108441, 2022. DOI: 10.1016/j.patcog.2021.108441.

[24] S. Makridakis, E. Spiliotis, and V. Assimakopoulos, "The m4 competition: 100,000 time series and 61 forecasting methods," *International Journal of Forecasting*, vol. 36, pp. 54–74, 2020. DOI: 10.1016/j.ijforecast.2019.04.014.

[25] S. Makridakis, E. Spiliotis, and V. Assimakopoulos, "M5 accuracy competition: Results, findings, and conclusions," *International Journal of Forecasting*, vol. 38, pp. 1346–1364, 2022. DOI: 10.1016/j.ijforecast.2021.11.013.

[26] S. Makridakis and M. Hibon, "The m3-competition: Results, conclusions and implications," *International Journal of Forecasting*, vol. 16, pp. 451–476, 2000. DOI: 10.1016/S0169-2070(00)00057-1.

[27] D. Salinas, V. Flunkert, J. Gasthaus, and T. Januschowski, "Deepar: Probabilistic forecasting with autoregressive recurrent networks," *International Journal of Forecasting*, vol. 36, pp. 1181–1191, 2020. DOI: 10.1016/j.ijforecast.2019.07.001.

[28] B. Lim, S. Ö. Arık, N. Loeff, and T. Pfister, "Temporal fusion transformers for interpretable multi-horizon time series forecasting," *International Journal of Forecasting*, vol. 37, pp. 1748–1764, 2021. DOI: 10.1016/j.ijforecast.2021.03.012.

[29] Y. Zhang, L. Ma, S. Pal, Y. Zhang, and M. Coates, "Multi-resolution time-series transformer for long-term forecasting," in *Proceedings of The 27th International Conference on Artificial Intelligence and Statistics*, vol. 238, 2024, pp. 4222–4230.

[30] A. F. Ansari, L. Stella, C. Türkmen, X. Zhang, P. Mercado, H. Shen, O. Shchur, S. S. Rangapuram, S. Pineda Arango, S. Kapoor, J. Zschiegner, D. C. Maddix, H. Wang, M. W. Mahoney, K. Torkkola, A. G. Wilson, M. Bohlke-Schneider, and Y. Wang, *Chronos: Learning the language of time series*, 2024. arXiv: 2403.07815.

[31] S. Sahoo and M. Jha, "On the statistical forecasting of groundwater levels in unconfined aquifer systems," *Environmental Earth Sciences*, vol. 73, pp. 3119–3136, 2015. DOI: 10.1007/s12665-014-3608-8.

[32] I. N. Daliakopoulos, P. Coulibaly, and I. K. Tsanis, "Groundwater level forecasting using artificial neural networks," *Journal of Hydrology*, vol. 309, pp. 229–240, 2005. DOI: 10.1016/j.jhydrol.2004.12.001.

[33] F. Mojtahedi, N. Yousefpour, S. H. Chow, and M. Cassidy, "Deep learning for time series forecasting: Review and applications in geotechnics and geosciences," *Archives of Computational Methods in Engineering*, vol. 32, pp. 3415–3445, 2025. DOI: 10.1007/s11831-025-10244-5.

[34] K. B. W. Boo, A. El-Shafie, F. Othman, M. M. H. Khan, A. H. Birima, and A. N. Ahmed, "Groundwater level forecasting with machine learning models: A review," *Water Research*, vol. 252, 121249, 2024. DOI: 10.1016/j.watres.2024.121249.

[35] M. J. Alam, S. Kar, S. Zaman, S. Ahamed, and K. Samiya, "Forecasting underground water levels: LSTM based model outperforms gru and decision tree based models," in *2022 IEEE International Women in Engineering (WIE) Conference on Electrical and Computer Engineering (WIECON-ECE)*, 2022, pp. 280–283. DOI: 10.1109/WIECON-ECE57977.2022.10151230.

[36] F. Feng, H. Ghorbani, and A. E. Radwan, "Predicting groundwater level using traditional and deep machine learning algorithms," *Frontiers in Environmental Science*, vol. 12, 1291327, 2024. DOI: 10.3389/fenvs.2024.1291327.

[37] N. Igwebuike, M. Ajayi, C. Okolie, T. Kanyerere, and T. Halihan, "Application of machine learning and deep learning for predicting groundwater levels in the west coast aquifer system, south africa," *Earth Science Informatics*, vol. 18, 6, 2024. DOI: 10.1007/s12145-024-01623-w.

[38] S. Thakur and S. Karmakar, "Predicting groundwater levels using advanced deep learning models: A case study of raipur, india," *Transactions of the Indian National Academy of Engineering*, vol. 10, pp. 551–558, 2025. DOI: 10.1007/s41403-025-00536-4.

[39] R. A. Collenteur, M. Bakker, R. Caljé, S. A. Klop, and F. Schaars, "Pastas: Open source software for the analysis of groundwater time series," *Groundwater*, vol. 57, pp. 877–885, 2019. DOI: 10.1111/gwat.12925.

[40] R. B. J. Brinkgreve, S. Kumarswamy, and W. M. Swolfs, *Plaxis 2016: General information*, Plaxis BV, 2016, ISBN: 978-90-76016-20-7. [Online]. Available: http://www.plaxis.nl.

[41] L. S. Besseling, A. Bomers, and S. J. M. H. Hulscher, "Predicting flood inundation after a dike breach using a long short-term memory (LSTM) neural network," *Hydrology*, vol. 11, 152, 2024. DOI: 10.3390/hydrology11090152.

[42] O. Triebe, H. Hewamalage, P. Pilyugina, N. Laptev, C. Bergmeir, and R. Rajagopal, *NeuralProphet: Explainable forecasting at scale*, 2021. arXiv: 2111.15214 [cs.LG].

[43] C. Wang and Q. Wu, *Flaml: A fast and lightweight automl library*, 2019. arXiv: 1911.04706 [cs.LG].

[44] D. Srihith, D. Donald, T. Srinivas, G. Thippanna, and P. Lakshmi, *Python's autots: Your co-pilot for time series analysis*, 2023. DOI: 10.5281/zenodo.8374966.

[45] X. Zhang, H. Wu, and J. Yang, *Hyperts: A full-pipeline automated time series analysis toolkit*, 2022. [Online]. Available: https://github.com/DataCanvasIO/HyperTS (visited on 04/14/2025).

[46] R. J. Hyndman, A. B. Koehler, R. D. Snyder, and S. Grose, "A state space framework for automatic forecasting using exponential smoothing methods," *International Journal of Forecasting*, vol. 18, no. 3, pp. 439–454, 2002. DOI: 10.1016/S0169-2070(01)00110-8.

[47] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "Lightgbm: A highly efficient gradient boosting decision tree," in *Advances in Neural Information Processing Systems*, vol. 30, 2017, pp. 3149–3157.

[48] Y. Nie, N. H. Nguyen, P. Sinthong, and J. Kalagnanam, *A time series is worth 64 words: Long-term forecasting with transformers*, 2023. arXiv: 2211.14730 [cs.LG].

[49] R. Caruana, A. Niculescu-Mizil, G. Crew, and A. Ksikes, "Ensemble selection from libraries of models," in *Proceedings of the Twenty-first International Conference on Machine Learning*, 2004, p. 18. DOI: 10.1145/1015330.1015432.

[50] F. Hutter, H. H. Hoos, and K. Leyton-Brown, "Sequential model-based optimization for general algorithm configuration," in *Learning and Intelligent Optimization*, 2011, pp. 507–523. DOI: 10.1007/978-3-642-25566-3_40.

[51] M. Christ, N. Braun, J. Neuffer, and A. W. Kempa-Liehr, "Time series feature extraction on basis of scalable hypothesis tests (TSFresh: A python package)," *Neurocomputing*, vol. 307, pp. 72–77, 2018. DOI: 10.1016/j.neucom.2018.03.067.

[52] E. Raponi, H. Wang, M. Bujny, S. Boria, and C. Doerr, "High dimensional bayesian optimization assisted by principal component analysis," in *Parallel Problem Solving from Nature – PPSN XVI*, 2020, pp. 169–183. DOI: 10.1007/978-3-030-58112-1_12.

[53] H. Abdi and L. J. Williams, "Principal component analysis," *WIREs Computational Statistics*, vol. 2, pp. 433–459, 2010. DOI: 10.1002/wics.101.

[54] E. Brochu, V. M. Cora, and N. de Freitas, *A tutorial on bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning*, 2010. arXiv: 1012.2599 [cs.LG].

[55] T. Räsänen and M. Kolehmainen, "Feature-based clustering for electricity use time series data," in *Adaptive and Natural Computing Algorithms*, 2009, pp. 401–412. DOI: 10.1007/978-3-642-04921-7_41.

[56] J. Zhang, X. Hu, Y. Xu, and W. Ding, "A fuzzy c-means clustering-based hybrid multivariate time series prediction framework with feature selection," *IEEE Transactions on Fuzzy Systems*, vol. 32, no. 8, pp. 4270–4284, 2024. DOI: 10.1109/TFUZZ.2024.3393622.

[57] R. Godahewa, K. Bandara, G. I. Webb, S. Smyl, and C. Bergmeir, "Ensembles of localised models for time series forecasting," *Knowledge-Based Systems*, vol. 233, 107518, 2021. DOI: 10.1016/j.knosys.2021.107518.

[58] J. Herzen, F. Lässig, S. G. Piazzetta, T. Neuer, L. Tafti, G. Raille, T. Van Pottelbergh, M. Pasieka, A. Skrodzki, N. Huguenin, M. Dumonal, J. Kościsz, D. Bader, F. Gusset, M. Benheddi, C. Williamson, M. Kosinski, M. Petrik, and G. Grosch, "Darts: User-friendly modern machine learning for time series," *Journal of Machine Learning Research*, vol. 23, pp. 1–6, 2022.

[59] H.-Y. Chen, Z. Vojinovic, W. Lo, and J.-W. Lee, "Groundwater level prediction with deep learning methods," *Water*, vol. 15, 3118, 2023. DOI: 10.3390/w15173118.

[60] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 785–794. DOI: 10.1145/2939672.2939785.

[61] B. Strijker, *Multi-year monitoring data of subsurface water conditions for dutch canal dikes*, 4TU.ResearchData, 2025. DOI: 10.4121/136aa5df-1907-43ac-a5b0-e0ea8f2dedf3.v2.

[62] A. Jacobs and H. de Bruin, "Makkink's equation for evapotranspiration applied to unstressed maize," *Hydrological Processes*, vol. 12, pp. 1063–1066, 1998. DOI: 10.1002/(SICI)1099-1085(19980615)12:7<1063::AID-HYP640>3.0.CO;2-2.

[63] H. Hakvoort, S. Bosch, R. Versteeg, and J. Heijker, *Meteobase: Online neerslagen referentiegewasverdampingsdatabase voor het nederlandse waterbeheer*, 2013. [Online]. Available: https://www.meteobase.nl (visited on 02/03/2025).

[64] D. Rey and M. Neuhäuser, "Wilcoxon-signed-rank test," in *International Encyclopedia of Statistical Science*, Springer Berlin Heidelberg, 2011, pp. 1658–1659. DOI: 10.1007/978-3-642-04898-2_616.

[65] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *Journal of Machine Learning Research*, vol. 7, pp. 1–30, 2006.