Multi-step Ahead Inflow Forecasting for a Norwegian Hydro-Power Use-Case, Based on Spatial-Temporal Attention Mechanism

Magnus Børseth

Supervisor: Lars Arne Jordanger

Co-Supervisor: Reza Arghandeh and Mojtaba Yousefi

Master's thesis in Software Engineering at

Department of Computer science, Electrical engineering and Mathematical sciences,
Western Norway University of Applied Sciences

Department of Informatics,
University of Bergen

June 2023

Western Norway
University of
Applied Sciences

UNIVERSITAS BERGENSIS

Acknowledgements

I would like to express my gratitude to my supervisors Lars Arne Jordanger, Reza Arghandeh, and Mojtaba Yousefi for their guidance, patience, and support throughout the course of this project. Gratitude extends to my peers, friends, and family whose encouragement has been invaluable.

Contents

I Background **1**

1 Hydrological Forecasting **2**
1.1 Importance of Hydrological Forecasting . . . 2
1.2 Approaches to Hydrological Forecasting . . . 3
1.3 Physically-Based Models . . . 3
1.4 Conventional Data-Driven Models: Statistical Methods . . . 4
1.5 Data-Driven Forecasting Methods: Machine Learning Models . . . 5

2 Machine Learning for Hydrological Forecasting **6**
2.1 Introduction to Machine Learning . . . 6
2.1.1 Machine Learning in Hydrology . . . 7
2.1.2 Popular Machine Learning Algorithms in Hydrology . . . 7
2.2 Feature Extraction and Selection in Machine Learning for Hydrology . . . 11
2.3 Hyperparameter Tuning in Machine Learning . . . 12

3 Spatial-Temporal Modeling in Hydrological Forecasting **13**
3.1 Importance of Spatial and Temporal Dependencies in Hydrology . . . 13
3.2 Techniques for Spatial and Temporal Modeling . . . 14
3.2.1 Spatial Modeling . . . 14
3.2.2 Temporal Modeling . . . 14
3.2.3 Spatial-Temporal Modeling . . . 14
3.3 Attention Mechanisms in Spatial and Temporal Modeling . . . 15
3.3.1 Spatial Attention . . . 15
3.3.2 Temporal Attention . . . 15
3.3.3 Spatial-Temporal Attention . . . 15

4 Methodology **16**
4.1 Literature Review Methodology . . . 16
4.2 Case Study Methodology . . . 16
4.2.1 Data and Study Area . . . 16
4.2.2 Data Cleaning . . . 17
4.2.3 Model Development . . . 18

5 Overview of Part II **22**
5.1 Literature Review of Data-Driven Methods Used for Forecasting Hydrological Events . . . 22
5.1.1 Context . . . 22
5.1.2 Results . . . 22
5.2 A Hydrological Case Study of Short-Term Inflow Forecasting with Spatial-Temporal Attention-Based LSTM . . . 23
5.2.1 Context . . . 23
5.2.2 Results . . . 23

II Papers **24**

6 Literature Review of Data-Driven Methods Used for Forecasting Hydrological Events **25**

7 A Hydrological Case Study of Short-Term Inflow Forecasting with Spatial-Temporal Attention-Based LSTM **34**

III Application **43**

8 Application: An Interactive Tool for Hydrological Forecasting **44**
8.1 Development Process . . . 44
8.2 Implementation and Technical Details . . . 45
8.3 Overview of the Application . . . 46
8.4 Application Design and Workflow . . . 46
8.4.1 Data Upload and Pre-Processing . . . 46
8.4.2 Model Selection and Training . . . 47
8.4.3 Data and Result Visualization . . . 47
8.5 Further Work: Future Potential and Limitations . . . 48

A Source code **50**
A.1 Model Training . . . 50
A.2 Model Evaluation . . . 53

B Data description **55**

C Application **58**

List of Figures

1.1 The two broad approaches to hydrological forecasting: Physically-based models and Data-driven models, the latter further classified into Statistical and ML models. 3

2.1 A flowchart depicting the iterative process of ML, from data collection and processing to model development, training, and testing. 7
2.2 Artificial Neural Network (ANN) diagram showcasing interconnected nodes arranged in input (x), two hidden (h), and output (y) layers. 9
2.3 A visualization of a Recurrent Neural Network (RNN). The RNN's structure is defined by three key nodes: input (x), hidden (h), and output (y). Notably, the hidden layer has a recurrent connection that feeds its output back as an additional input for the next time step. 10
2.4 Diagram of an LSTM cell, highlighting its gating mechanisms: the forget gate, input gate, and output gate. These gates control the flow and modification of information within the cell, enabling LSTM's long-term memory capability. 11

4.1 Study area map showing the catchment area in the Stavanger region, southwestern coast of Norway. 17
4.2 Visualization of missing values in the dataset, with the marked area indicating the filtered dataset used in the case study. 18

8.1 Flowchart illustrating the workflow of the interactive hydrological forecasting application, showcasing the process of data upload, model training, and result visualization. 47

All figures in this thesis were generated by the author using various software tools, including Miro for diagramming, Plotly for data visualization, Missingno for handling visualization of missing data, and NVE Temakart for geographic maps

List of Papers

1 Literature Review of Data-Driven Methods Used for Forecasting Hydrological Events 25
2 A Hydrological Case Study of Short-Term Inflow Forecasting with Spatial-Temporal Attention-Based LSTM 34

List of Acronyms

R^2 Coefficient of Determination.

ANN Artificial Neural Networks.

ARIMA Auto Regressive Integrated Moving Average.

ARMA Auto Regressive Moving Average.

CNN Convolutional Neural Networks.

GRU Gated Recurrent Units.

HA Historical Average.

HBV Hydrologiska Byråns Vattenbalansavdelnin.

LSTM Long Short-Term Memory.

MAE Mean Absolute Error.

MAPE Mean Absolute Percentage Error.

ML Machine Learning.

MLR Multiple Linear Regression.

MSE Mean Squared Error.

PBT Population Based Training.

RMSE Root Mean Squared Error.

RNN Recurrent Neural Networks.

SVM Support Vector Machines.

Summary

Hydrological forecasting has been an ongoing area of research due to its importance to improve decision making on water resource management, flood management, and climate change mitigation. With the increasing availability of hydrological data, Machine Learning (ML) techniques have started to play an important role, enabling us to better understand and predict complex hydrological events. However, some challenges remain. Hydrological processes have spatial and temporal dependencies that are not always easy to capture with traditional ML models, and a thorough understanding of these dependencies is essential when developing accurate predictive models.

This thesis explores the use of ML techniques in hydrological forecasting and consists of an introduction, two papers, and an application developed alongside the case study. The motivation for this research is to enhance our understanding of the spatial and temporal dependencies in hydrological processes and to explore how ML techniques, particularly those incorporating attention mechanisms, can aid in hydrological forecasting.

The first paper is a chronological literature review that explores the development of data-driven forecasting in hydrology, and highlighting the potential application of attention mechanisms in hydrological forecasting. These attention mechanisms have proven to be successful in various domains, allowing models to focus on the most relevant parts of the input for making predictions, which is particularly useful when dealing with spatial and temporal data.

The second paper is a case study of a specific ML model incorporating these attention mechanisms. The focus is to illustrate the influence of spatial and temporal dependencies in a real-world hydrological forecasting scenario, thereby showcasing the practical application of these techniques.

In parallel with the case study, an application has been developed, employing the principles and techniques discovered throughout the course of this research. The application aims to provide a practical demonstration of the concepts explored in the thesis, contributing to the field of hydrological forecasting by introducing a tool for hydropower suppliers.

Part I

Background

Chapter 1

Hydrological Forecasting

1.1 Importance of Hydrological Forecasting

Hydrological forecasting is an important tool in water resource management, flood management, and climate change mitigation. It involves predicting future water values such as streamflow and inflow, based on past and present historical data. These forecasts play a significant role in different fields, and influence decision making and strategic planning [14, 21].

An area where hydrological forecasting displays its importance is disaster management, particularly in relation to floods [13] and droughts [8]. By providing precise and accurate hydrological forecasts, disaster preparedness and mitigation can be improved, resulting in reduced damages and potentially saving lives. For example, flood forecasts can provide an early warning to areas which are flood-prone, allowing them to evacuate and reduce the risk of casualties.

In resource management, hydrological forecasts are critical for optimizing the allocation of water resources. These forecasts can help inform the decision making related to the distribution of water between different uses, such as domestic consumption, industry, and maintaining ecological flows [22].

In the hydropower sector, accurate hydrological forecasts help optimize power generation [10]. By creating accurate inflow forecasts into reservoirs, hydropower suppliers can optimize the generation schedule to match peak demand, which would maximize efficiency and revenue.

However, it is important to note that hydrological forecasting is dependent on its accuracy. Errors or inaccuracies in forecasts can have significant consequences. As an example, an overestimation of precipitation can lead to unnecessary discharge of water from a reservoir, wasting valuable water resources. Similarly, underestimation of a flood event can delay necessary evacuations, putting many lives at risk. Therefore, advancements in hydrological forecasting methodologies are important for improving accuracy and reducing uncertainty, leading to better informed decision-making and resource allocation.

1.2 Approaches to Hydrological Forecasting

Hydrological forecasting has had a rapid development over the years, due to increased data availability and computational power [10]. This has lead to various methodologies and approaches being developed for creating accurate predictions. As seen in Figure 1.1, these approaches can be classified into two broad categories: physically-based models and data-driven models.

Physically-based models are deterministic, leveraging mathematical equations based on the laws of physics that describe the movement and storage of water in a hydrological cycle. These models are highly dependent on the understanding and representation of physical processes. On the other hand, data-driven methods take advantage of computational advancements and the increasing availability of data to learn relationships directly from data.

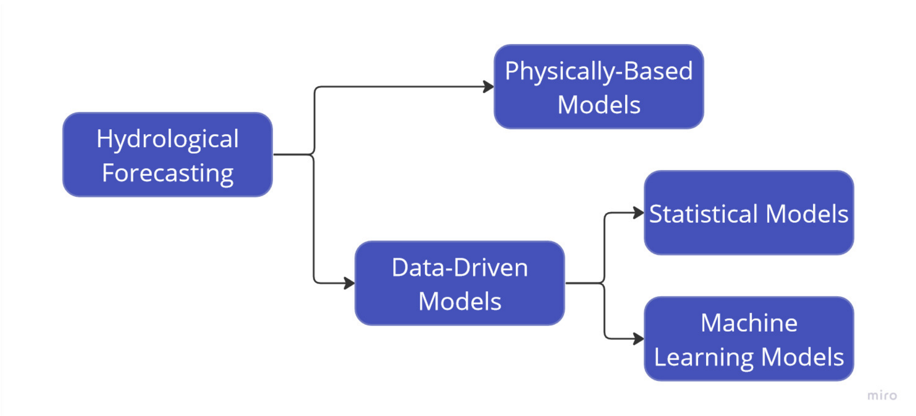

Figure 1.1: The two broad approaches to hydrological forecasting: Physically-based models and Data-driven models, the latter further classified into Statistical and ML models.

1.3 Physically-Based Models

Physically-based models in hydrology attempt to simulate the intricate processes of the water cycle [25]. These models rely on deterministic methods that use physical principles and mathematical equations to understand the behavior of the hydrological cycle. Therefore, a deep understanding of the physical processes involved is crucial for their successful implementation and operation.

An example of physically-based models is the Hydrologiska Byråns Vattenbalansavdelnin (HBV) model, first introduced by the Swedish Meteorological and Hydrological Institute [24]. The HBV model contains several components, each representing a specific aspect of the hydrological cycle, including snow accumulation and melt, soil moisture, and water runoff [25]. It applies a mix of empirical and physically-based relationships to simulate the hydrological responses within a catchment area. The model's advantage lies in its simplicity, robustness, and flexibility to handle varied geographical and climatic conditions, making it a widely adopted tool in hydrological forecasting.

However, it is essential to note that the performance and reliability of physically-based models, including the HBV, often depend on the quality of the hydro-meteorological data as well as the accurate representation of the complex physical processes within the model. Any inaccuracies in the input data or deficiencies in the model representation could lead to significant errors in the forecast.

1.4 Conventional Data-Driven Models: Statistical Methods

Statistical methods are data-driven and rely on identifying and exploiting statistical patterns in the data. They typically require assumptions about the statistical properties of the data and the relationships between variables. Despite some limitations, particularly with non-linear, these conventional models have been widely used in hydrological forecasting due to their simplicity, interoperability and ease of implementation. Some often used models in hydrological forecasting are the Historical Average (HA), Multiple Linear Regression (MLR) [7], Auto Regressive Moving Average (ARMA) [23], and Auto Regressive Integrated Moving Average (ARIMA) [28] models.

Historical Average (HA)

The HA model is a simplistic but occasionally effective forecasting tool. It works on the premise that the predicted future value is calculated by the historical mean of the previous sequence of data. In the context of hydrological forecasting, the HA model often functions as a benchmark against which the performance of more intricate models is measured (see, e.g., [12]). Its benefits lie in its straightforwardness, both in terms of understanding and implementation, and that it does not make any assumptions about the statistical properties of the data. Despite these advantages, HA does have significant limitations. Its simplicity makes it unable to capture more complex data patterns such as trends, seasonality, and non-linear relationships.

Multiple Linear Regression (MLR)

MLR is a statistical technique used to model the relationship between two or more variables and a response variable by fitting a linear equation on the historical data. In the context of hydrological forecasting, the response variable can be the hydrological variable we want to predict, such as inflow, while the explanatory variables can be other parameters that influence it, such as precipitation and temperature.

The advantages of MLR include its simplicity, interoperability, and speed of computation [7]. It also provides a measure of the strength of the relationship between variables through the regression coefficients. However, MLR assumes a linear relationship between variables, which is often not the case in hydrological processes.

Auto Regressive Moving Average (ARMA)

The ARMA model is a combination of two stochastic processes: Autoregressive (AR) and Moving Average (MA). ARMA models are often used with time series data, which is the case for hydrological forecasting. The AR component models the current value of the time series as a linear combination of past values, while the MA component models the current value as a linear combination of past error. The order of the AR and MA components, i.e. how many past values or errors are considered, are selected based on the data.

ARMA models are capable of modeling many different time series patterns and are simple to understand and implement. However, they are based on the assumption that the time series data is stationary, i.e. that the statistical properties of the data do not change over time, which is often not the case for hydrological data [23]. The ARMA models also struggle with modeling long-term trends and seasonal patterns.

Auto Regressive Integrated Moving Average (ARIMA)

The ARIMA model is an extension of the ARMA model, which includes an *integrated* component to account for non-stationary in the data. The *I* in ARIMA refers to the number of times the data have been differenced to make the time series stationary. ARIMA models are more flexible than ARMA models and can handle a wider range of time series patterns, including non-stationary data. They are also capable of modeling long-term trends. However, like ARMA models, they are based on the assumption that there is a linear relationship between variables and struggle with non-linear data [28]. Additionally, they also require a significant amount of data to accurately estimate the models parameters.

1.5 Data-Driven Forecasting Methods: Machine Learning Models

Machine Learning (ML) has emerged as a powerful alternative to the traditional forecasting methods, significantly advancing hydrological predictions [10]. Unlike conventional approaches, ML models are characterized by their data focused approaches [19], which do not require an established understanding of the relationships between variables and outcomes. This creates a flexibility that enables the ML models to uncover intricate patterns and non-linear relationships from the data, which conventional methods struggle with. Some of the commonly used models employed in hydrological forecasting include Support Vector Machines (SVMs) [20], Artificial Neural Networks (ANNs) [17], and Recurrent Neural Networks (RNNs) [9], among others.

The strength of ML models lies in their ability to learn and adapt from data. Moreover, they are highly versatile and scalable, meaning they can handle large datasets and high-dimensional inputs. However, the main disadvantage of ML models is their lack of interpretability, often referred to as the *black box* problem [30].

Chapter 2

Machine Learning for Hydrological Forecasting

This chapter serves as a comprehensive introduction providing the necessary background knowledge required to further investigate the application of ML in the field of hydrological forecasting. Understanding the fundamental concepts of ML and its application in hydrology is an important step in identifying the benefits and challenges in creating different techniques for hydrological forecasting.

2.1 Introduction to Machine Learning

ML is a sub-field of Artificial Intelligence, and is centered around the idea of enabling computers to learn from data and to improve their performance without being explicitly programmed [21]. Essentially, these algorithms can identify patterns and learn from experience. The learning process is guided by a variety of concepts, such as the representation of data, the parameters used, the evaluation of models, and the optimization of algorithms.

In ML, there are primarily three types of learning categories, each of which is characterized by its learning process, as described in [11]:

- **Supervised Learning**: In this category, the algorithm is trained on labeled data, meaning all data in the training set are paired with an expected output. The algorithm learns a function that maps the input to the output. The objective of supervised learning is to accurately predict the output for unseen data. It is commonly used for tasks such as regression (predicting continuous outputs) and classification (predicting discrete outputs).
- **Unsupervised Learning**: This category involves training algorithms on data without predetermined labels. Because the data is unlabeled during training, there is no correct answer, which can make it challenging to evaluate the performance of the ML algorithm. The goal of unsupervised learning is to identify patterns, correlations, or clusters in the input data.

- **Reinforcement Learning**: Reinforcement learning is a type of ML where the algorithm learns to make decisions by interacting with an environment. The algorithm receives feedback in the form of rewards or penalties and aims to maximize the total reward over time. Unlike supervised and unsupervised learning, reinforcement learning focuses more on learning which actions to take in specific situations.

The choice of these learning categories depend on the specific problem to be solved and the data at disposal, with each category having its unique capabilities and relevance.

2.1.1 Machine Learning in Hydrology

ML has emerged as a valuable tool in hydrological forecasting and modeling, offering a data-driven approach to predict complex hydrological processes. This is achieved mainly through supervised learning methods, given the predictive requirements of the tasks and the existence of historical data. The increasing availability of data, alongside advances in computational power, have given the adoption of ML techniques to uncover hidden relationships and patterns in hydrological data, which may not be easily captured by traditional physically-based or statistical methods.

Figure 2.1 presents a flowchart depicting the iterative process of ML in hydrology, beginning with data collection and pre-processing. The data used in hydrological forecasting can include historical records of inflow, precipitation, temperature, and other relevant variables. After processing, the data is divided into training and testing sets. The training set is used to develop and train the ML model, while the testing set is employed to evaluate its performance and generalization capabilities.

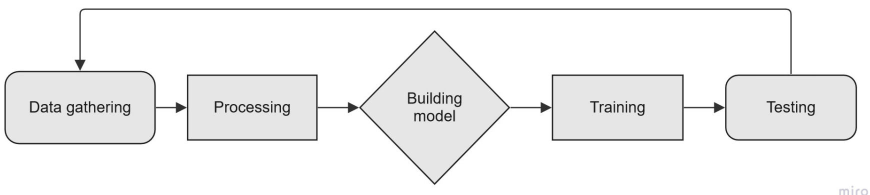

Figure 2.1: A flowchart depicting the iterative process of ML, from data collection and processing to model development, training, and testing.

2.1.2 Popular Machine Learning Algorithms in Hydrology

In the field of hydrological forecasting, ML models such as Support Vector Machines (SVM), Artificial Neural Networks (ANN), Recurrent Neural Networks (RNN), and Long Short-Term Memory (LSTM) have demonstrated promising results. See Paper 1 in Chapter 6 for a more detailed introduction on how these have been used in hydrological forecasting.

Support Vector Machines (SVM)

The SVM is a widely used ML method for both classification and regression problems. Although SVMs are primarily used for classification, where they optimize the differences between groups, they can also be adapted for regression by transforming the data. Essentially, this approach allows the SVM to predict continuous rather than categorical variables, making it a versatile tool in the field of ML [21].

Despite their versatility, SVMs are not without limitations. Their efficiency tends to decline when confronted with large datasets, as the training time increases exponentially with the size of the data, making them less suitable for big data applications. Similarly, SVMs might encounter difficulties in addressing complex nonlinear problems, where other methods such as deep learning might outperform them. Additionally, SVMs are sensitive to missing data and require a comprehensive pre-processing step to handle such scenarios, which may add to the complexity of the model building process [15]. Careful consideration should be given to these limitations when selecting SVMs as the forecasting tool for hydrological predictions.

Artificial Neural Networks (ANN)

ANNs were designed to emulate the biological neural systems in the human brain, aiming to make decisions in a human-like manner. As illustrated in Figure 2.2, an ANN comprises of interconnected computational nodes, also known as neurons, categorized into three types of layers: input, hidden, and output layers.

Data propagates through these layers in a process called forward propagation, where each neuron applies an activation function to the input it receives and passes the result to the neurons in the next layer. This activation function introduces non-linearity into the system, giving the ANN the capability to model and solve complex, non-linear problems.

After the forward propagation, an error is calculated using a loss function that measures the difference between the network's prediction and the actual output. This error is then propagated backward through the network, a process known as back-propagation, adjusting the weights and biases of the neurons in a way that minimizes the error.

Through this architecture and the ability to learn from the input-output mappings during a training process, ANNs are equipped to perform sophisticated pattern recognition, approximation, and predictive tasks, which mirror the human decision-making process [16].

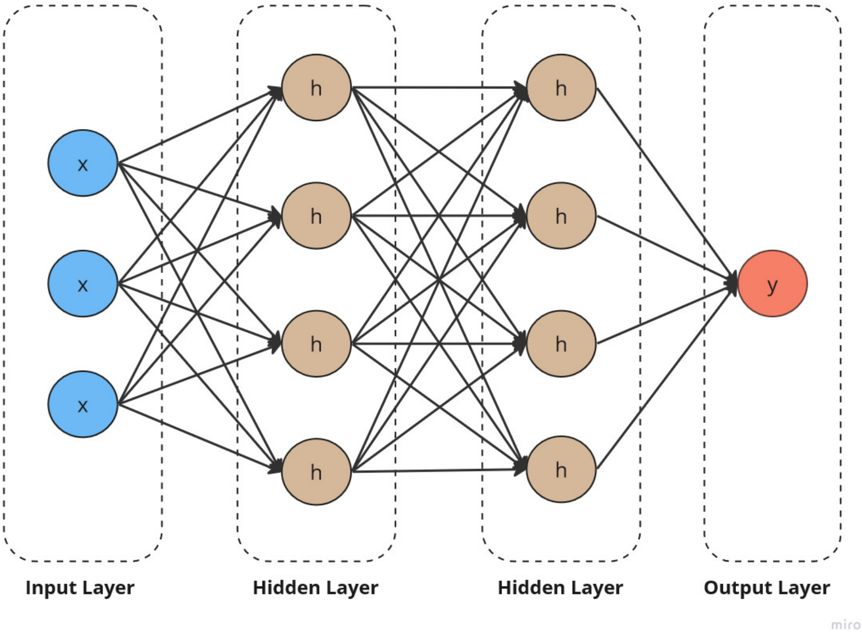

Figure 2.2: Artificial Neural Network (ANN) diagram showcasing interconnected nodes arranged in input (x), two hidden (h), and output (y) layers.

Recurrent Neural Networks (RNN)

RNNs are a unique subset of deep learning models, specifically designed for processing sequential or time-series data. This distinct capability comes from their inherent architecture which features *memory*, retaining information from prior inputs to influence future predictions.

The RNN's structure, as shown in Figure 2.3, consists of three key nodes: input (x), hidden (h), and output (y). The defining feature of RNNs is the recurrent connection in the hidden layer, where the output from a hidden node is fed back into the same node as part of the input for the next time step. This looping connection enables the model to propagate information through time, providing essential context for predictions across sequential data points.

However, a noteworthy limitation of standard RNNs is their difficulty in capturing long-term dependencies due to the phenomenon of vanishing and exploding gradients during training [9]. This problem occurs when the gradients of the loss function, used for updating the model's weights, become excessively small (vanish) or large (explode) as they are propagated backward through many time steps.

To mitigate these issues, more sophisticated variations of RNNs have been introduced, such as Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU).

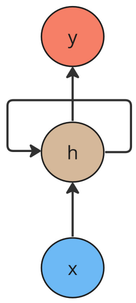

Figure 2.3: A visualization of a Recurrent Neural Network (RNN). The RNN's structure is defined by three key nodes: input (x), hidden (h), and output (y). Notably, the hidden layer has a recurrent connection that feeds its output back as an additional input for the next time step.

Long Short-Term Memory (LSTM)

LSTM is a specialized variant of RNNs, engineered to circumvent the problem of long-term dependency that conventional RNNs encounter. LSTM networks achieve their ability to manage information outside the regular flow of the recurrent network through the incorporation of *gating mechanisms* within each LSTM cell [9]. This mechanism enables LSTM to selectively retain or discard information over extended periods, making it highly effective for tasks with temporal dependencies such as time-series forecasting and natural language processing.

An LSTM cell comprises three key components:

- The *input gate* is responsible for determining the extent of the newly computed state for the current input that should be allowed into the cell.
- The *forget gate* decides how much of the existing state should be preserved.
- The *output gate* determines the amount of the internal state that should be revealed to the external network.

Figure 2.4 shows how these gates interact in an LSTM cell.

The operations of these gates are mathematically represented as follows [18]:

$$
\begin{aligned}
g_t &= \sigma(U_g x_t + W_g h_{t-1} + b_f) && \text{(forget gate)} \\
i_t &= \sigma(U_i x_t + W_i h_{t-1} + b_i) && \text{(input gate)} \\
\tilde{c}_t &= \tanh(U_c x_t + W_c h_{t-1} + b_c) && \text{(new candidate values)} \\
c_t &= g_t \cdot c_{t-1} + i_t \cdot \tilde{c}_t && \text{(cell state update)} \\
o_t &= \sigma(U_o x_t + W_o h_{t-1} + b_o) && \text{(output gate)} \\
h_t &= o_t \cdot \tanh(c_t) && \text{(hidden state update)}
\end{aligned}
$$

These equations illustrate how an LSTM cell operates on the input x_t and the previous hidden state h_{t-1} to generate the current hidden state h_t and cell

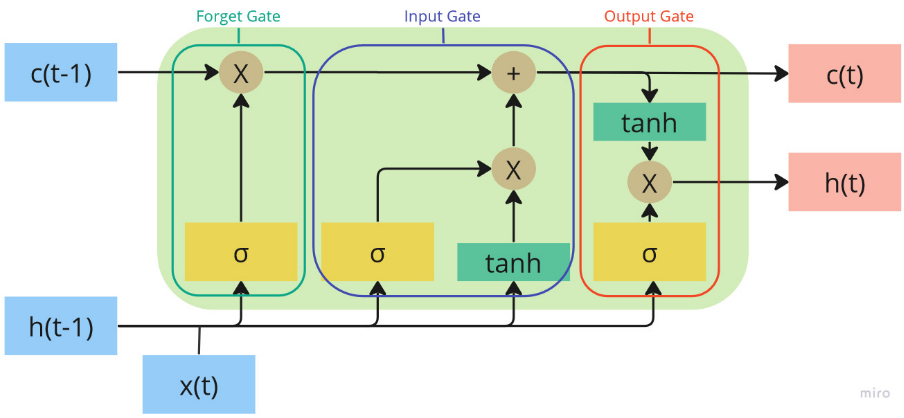

Figure 2.4: Diagram of an LSTM cell, highlighting its gating mechanisms: the forget gate, input gate, and output gate. These gates control the flow and modification of information within the cell, enabling LSTM's long-term memory capability.

state c_t. Here, U and W denote weight matrices for the input and the state respectively, while b represents the bias terms. The symbol σ represents the sigmoid activation function, and tanh denotes the hyperbolic tangent activation function. These combined operations enable the LSTM's ability to learn long-term dependencies.

2.2 Feature Extraction and Selection in Machine Learning for Hydrology

Feature extraction is the process of transforming data into features that represent the underlying patterns, while feature selection involves selecting the most relevant features for the forecasting task. These methods can be a crucial step in the ML pipeline, as the performance of the ML models is highly dependant on the quality and relevance of the input features.

In the context of hydrological forecasting, feature extraction involves deriving new variables from the data that capture the most important aspects of the data. For example, variables representing seasonal trends or flood events may be derived from hydrological time series data. Feature selection, on the other hand, involves selecting the hydrological or meteorological variables that are most relevant for predicting an outcome, such as inflow.

Effective feature extraction can help capture the complex, non-linear relationships that often exist in the hydrological data, while feature selection can help reduce the dimensionality of the data. The methods help improve the models' interpretability, reducing overfitting and improving computational efficiency.

2.3 Hyperparameter Tuning in Machine Learning

Hyperparameter tuning is a critical step in ML, responsible for optimizing the configurations of a model's learning process. The hyperparameter search help identify the optimal parameters learned during training, such as learning rates, regularization factors, the number of hidden layers in the neural network, or other important parameters [27].

Traditional approaches to hyperparameter tuning often involve a tedious and computationally expensive process of trial and error. However, there are several systematic strategies developed to address this, including Grid Search, Random Search, and Population Based Training (PBT), each of which has its unique strengths and drawbacks.

Grid Search

Grid search is the most straightforward method. It systematically works through multiple combinations of parameters, cross-validating as it goes, to determine which parameters provide the best performance [11]. Essentially, it performs a search through a manually specified subset of the hyperparameter space.

Random Search

In contrast to grid search, random search selects random combinations of parameters to train the model [11]. This strategy allows the model to explore a broader range of parameters in the same amount of time. Unlike grid search, which is restricted to a discrete set of possible parameter values, random search can explore a continuous space of possibilities.

Population Based Training

More advanced methods such as Population Based Training (PBT) combine the benefits of both grid search and random search. PBT starts with random hyperparameters, but unlike traditional methods that treat each set of hyperparameters independently, PBT optimizes a population of models concurrently [6]. Periodically, underperforming models clone the hyperparameters of the better performing models and mutate these parameters, exploring new hyperparameter configurations. Over time, the population adapts to favor hyperparameters that lead to higher performance.

Chapter 3

Spatial-Temporal Modeling in Hydrological Forecasting

Spatial-temporal modeling attempts to capture the spatial and temporal dependencies that exist within hydrological processes, and has become a central element in hydrological forecasting. This chapter explores the significance of both spatial and temporal dependencies in hydrology, the various techniques used to model these dependencies, and the growing role of attention mechanisms to capture both the spatial and temporal dependencies.

3.1 Importance of Spatial and Temporal Dependencies in Hydrology

Spatial and temporal dependencies are fundamental to the understanding and prediction of hydrological forecasting. The term *spatial* refers to geographical locations, while *temporal* refers to time. In hydrology, these dependencies show that both the location and timing of hydrological events are dependent on each other, and changes in one can directly influence the other.

Spatial dependencies in hydrology are connected to water related variables, such as precipitation and water level changes, across different geographical locations. Movement and distribution of water in a river-system are influenced by factors such a topography and soil characteristics. These spatial factors can lead to significant variability in hydrological processes across different regions within the same catchment area.

Temporal dependencies capture the changes in hydrological processes over time, reflecting the dynamics of the processes, such as the delay between rainfall and the increase in river flow. These temporal dependant changes can significantly impact the state of a hydrological system at any given moment and influence its future values.

Together, spatial and temporal dependencies capture the complex interaction of variables in hydrological processes. For instance, precipitation in a certain region will eventually influence the water level downstream. Similarly, historical weather patterns can provide important context for forecasting future hydrological conditions. Therefore, understanding and accurately modeling these dependencies are important for making reliable hydrological forecasts.

3.2 Techniques for Spatial and Temporal Modeling

Modeling the spatial and temporal dependencies within hydrological processes is a complex task due to the intrinsic variability and inter-connectivity of water-related variables. This is where ML proves to be beneficial, by providing powerful tools and techniques to accurately model the dependencies [30]. The techniques can be separated into three categories: spatial modeling, temporal modeling, and spatial-temporal modeling.

3.2.1 Spatial Modeling

Spatial modeling techniques focus on capturing these dependencies between different geographical locations. A popular technique for spatial modeling is Convolutional Neural Networks (CNNs). CNNs were originally designed for image processing, but the idea of using them to model spatial dependencies derives from their ability to process and analyze data with a grid-like topology [13]. They effectively learn features in the input data by convoluting small, learnable filters across the spatial dimensions, making them adept at capturing spatial correlations in hydrological data.

3.2.2 Temporal Modeling

To capture the dynamics of hydrological processes over time, temporal modeling techniques are used. RNNs are commonly used due to their ability to deal with sequential data by storing past information in hidden stats, which influence the output at the current time step. This makes them suited for modeling time-dependent changes in hydrological processes.

3.2.3 Spatial-Temporal Modeling

To capture both the spatial and temporal dependencies, hybrid models such as Convolutional Long Short-Term Memory (ConvLSTM) networks have been developed [26]. ConvLSTM integrates the spatial feature learning capabilities of CNNs with the temporal modeling strengths of LSTM. This allows ConvLSTM to capture the spatial-temporal dependencies in hydrological data effectively, providing more accurate predictions than models that only consider spatial or temporal dependencies separately.

3.3 Attention Mechanisms in Spatial and Temporal Modeling

In the field of ML, attention mechanisms are a novel approach that concentrate computational resources on the most significant portions of input data. They are designed to selectively focus on a subset of information, while simultaneously processing a broader range of inputs. Attention mechanisms enable a model to weigh the importance of various data points differently. This selective focus is data-dependent, meaning the model assigns different levels of importance to various inputs based on the specific information contained within the data [13, 29].

Therefore, when modeling spatial and temporal dependencies, attention mechanisms have emerged as a powerful tool in the field of ML, primarily due to their ability to focus on relevant parts of the input data and to ignore irrelevant parts [29]. This capability is particularly useful in modeling the spatial and temporal dependencies in hydrological data, where certain locations or time steps may be more important than others for predicting a specific output.

3.3.1 Spatial Attention

In the context of spatial modeling, attention mechanisms can help identify important regions in the input data that contribute more significantly to the output. For instance, in a hydrological forecasting scenario, certain areas within a catchment may be more influential on the water levels. A spatial attention mechanism can learn to focus on these critical areas and assign them higher weights in the model's decision-making process.

3.3.2 Temporal Attention

Temporal attention mechanisms work similarly but in the temporal dimension. They allow a model to focus on specific time steps that are more crucial for prediction. This is especially relevant in hydrological forecasting, where recent rainfall events or historical seasonal patterns may have a greater impact on future water levels.

3.3.3 Spatial-Temporal Attention

Just as spatial and temporal modeling techniques can be combined to capture both types of dependencies simultaneously, so too can spatial and temporal attention mechanisms. Spatial-temporal attention mechanisms can focus on the most relevant locations and time steps simultaneously, providing a better understanding of the spatial-temporal dependencies in hydrological data.

Chapter 4

Methodology

This chapter provides an overview of the methodology applied in this study, building on the foundation laid in the Papers 1 (refer to Chapter 6) and 2 (refer to Chapter 7). The methodologies used in these papers, ranging from data collection and processing to hyperparameter tuning and model development, were done and fine-tuned to fit the requirements of this research. Although there will be some overlap with the methodologies presented in the papers, this chapter aims to provide a more comprehensive overview. For a more detailed description of the methodologies used in Paper 1 and Paper 2, please refer to Chapter 6 and Chapter 7, respectively.

4.1 Literature Review Methodology

The first phase of the research involved a chronological literature review to provide a theoretical understanding of the development of hydrological forecasting and to identify gaps in the existing research. This was achieved by systematically searching for and analyzing relevant academic papers and reports using databases, specifically Google Scholar and Engineering Village. The search terms were selected based on their relevance to the topic, and included combinations of the following: "data-driven", "spatio-temporal", "spatial–temporal", "inflow", "steamflow", "forecast", "hydrology", "meteorology", and "attention", among others. The selection of these particular search terms was driven by the presumption that they would capture the most relevant published results to this field.

4.2 Case Study Methodology

4.2.1 Data and Study Area

This research centers on a catchment area situated in the Stavanger region, on the southwestern coast of Norway. The data for this study was provided by Lyse Energy, a Norwegian hydroelectric power company based in the study area. As shown in Figure 4.1, this region is characterized by diverse topography and a temperate oceanic climate. Its complex hydrological system, consisting of rivers, streams, and reservoirs, is influenced by several meteorological factors.

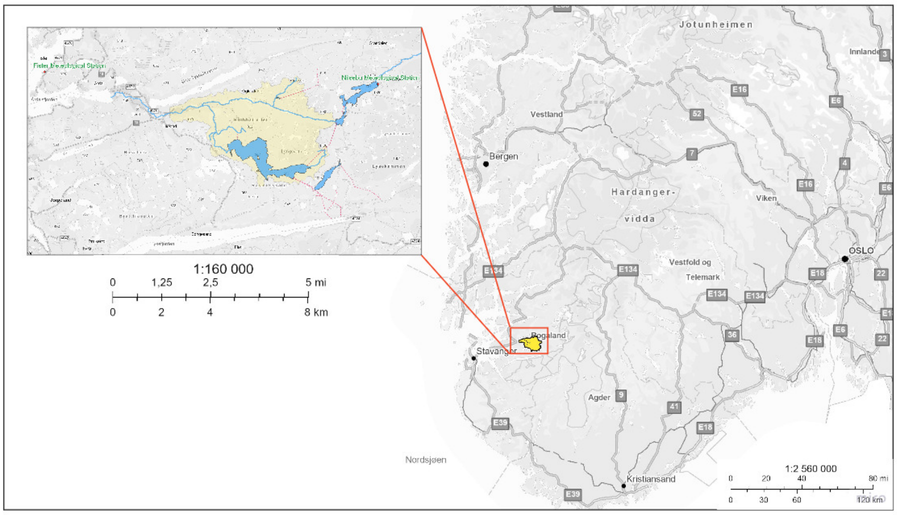

Figure 4.1: Study area map showing the catchment area in the Stavanger region, southwestern coast of Norway.

The dataset initially obtained for this research spans from January 2015 to March 2022, and contains 63288 data points. This raw dataset includes a variety of hydrological and meteorological variables, with data collected at hourly intervals. Alongside this, the dataset was included with simulated data from the HBV hydrological model. However, due to inconsistencies in sensor operation and the simulated data, not all variables cover the entire time range.

The focus of the case study, however, is a filtered subset of the original dataset. This subset, marked by the red area in Figure 4.2, consists of 16500 data points, ranging from November 2018, to September 2020. The filtered dataset was selected based on its comprehensive coverage of spatial features essential for capturing the intricate spatial correlations within the catchment area.

4.2.2 Data Cleaning

The data cleaning process involved several pre-processing steps, including addressing missing values, outliers, and inconsistencies present in the hydrological and meteorological data. Data imputation techniques, such as linear interpolation, were used to fill gaps in the time series data. This technique was validated by comparing interpolated values with actual measurements whenever feasible.

Outliers in the data were identified by examining the overall descriptive statistics (see Appendix B.1), which provided insights into the structure of the variables. The identified outliers were replaced using linear interpolation to minimize their potential impact on the model's performance. Please refer to paper 2 (see Chapter 7) for a more in-depth analysis.

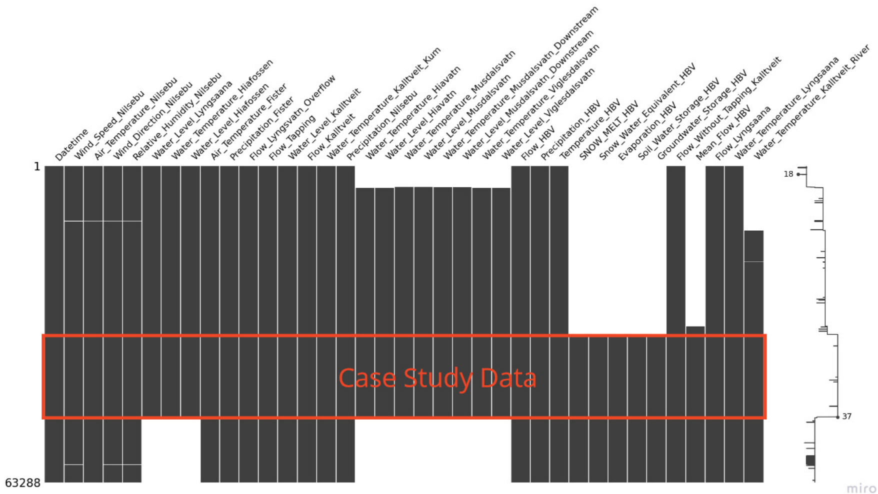

Figure 4.2: Visualization of missing values in the dataset, with the marked area indicating the filtered dataset used in the case study.

To address disparities in the quantity of data points and the number of features present in each group, the dataset was categorized into four distinct groups based on data availability. This resulted in the formation of datasets with fewer data points but a greater number of variables. Despite having fewer time series points, the filtered dataset with the most spatial features was selected for the case study due to its comprehensive set of spatial variables. Descriptive statistics of this cleaned dataset can be found in Appendix B.2.

Additionally, data normalization and transformation procedures were implemented to ensure compatibility and to standardize the scale of all variables before inputting them into the deep learning model. Min-max normalization was used to transform all variables to a common scale, ranging from 0 to 1, thus mitigating potential biases or inaccuracies that might arise from different scales and units.

4.2.3 Model Development

The development of the model involved a number of key steps, which included data pre-processing, feature selection, model architecture design, model training, and model evaluation.

Data Pre-Processing

Normalizing data is important for ML models, including LSTMs, because they are sensitive to the scale of input data [8]. Following normalization, the time series data were converted into a format suitable for training the LSTM model. LSTM models and other neural networks require input data in a specific three-dimensional format, consisting of the sample size, time steps, and number of features. Hence, the data was reshaped accordingly, and a sliding window approach was adopted to create sequences of data for training the model. This method involves creating *windows* of consecutive data points to predict the next point in the sequence, allowing the model to learn temporal dependencies in the data.

Feature Selection

Feature selection was used to categorize the data based on domain knowledge and data structure. The data were first separated based on their type and origin: location-based, meteorological, hydrological, discharge, and simulated hydrological data generated by the HBV model [24]. This separation was done based on the understanding of how different types of variables can influence the hydrological dynamics in the catchment area, and to make sure that the input to the model contained spatial variables.

Model Architecture Design

The design of the model architecture for this study was an essential part, where the LSTM acts as the core model, combined with spatial and temporal attention mechanisms. The architecture design aimed to efficiently capture the spatial and temporal dependencies in hydrological and meteorological time series data to improve the accuracy of inflow forecasting.

The Spatial-Temporal Attention-Based LSTM model consists of several components (see Paper 2 in Chapter 7 for more details):

- **Batch Normalization**: The model begins with a batch normalization layer, which standardizes the input features to have a mean of zero and a variance of one. This process accelerates the training process and helps in regularizing the model, reducing the risk of overfitting.
- **Spatial Attention Module**: The next layer in the model is the spatial attention module, which assigns different importance weights to each spatial feature in the input data. By emphasizing relevant spatial features and capturing the spatial dependencies within the catchment area, the model enhances its capacity to understand spatial dynamics for accurate inflow forecasting.
- **Input Linear Layer**: Following spatial attention, the data passes through an input linear layer. This layer is a fully connected neural network that adjusts the dimensionality of the input data to match the dimensionality of the LSTM's hidden states.
- **LSTM Layer**: The core of the architecture is the LSTM layer, designed to capture the temporal dependencies in the sequence data. It is capable of retaining and updating relevant information over extended periods, essential for understanding patterns and relationships in hydro-meteorological time series data.
- **Temporal Attention Module**: After the LSTM layer is the temporal attention module. It creates a context vector from the hidden states of the LSTM layer, which highlights the most relevant temporal information for inflow forecasting. By assigning different weights to each time step in the

model, it makes the model selectively concentrate on the most significant periods in the sequence.

- **Output Linear Layer**: Lastly, the output from the temporal attention module is passed through an output linear layer, a fully connected layer that transforms it to the final inflow prediction.

Model Training

The model training process focuses on optimizing the model's weights, which are the parameters that the model uses to make its predictions. The process is iterative, usually conducted over multiple epochs, with the aim of minimizing the loss.

First, the dataset is split into training, validation and test sets. The training set is used to update the models parameters, while the validation set is used to check the model's performance and control overfitting. A 70-20-10 split was adopted for this purpose, where 70% is used for training, 20% is used for validating, and 10% is used for testing.

The training process starts with initializing the model weights randomly. For each epoch, the model used the training set to generate predictions and calculate the loss, which is a measure of the discrepancy between the mode's prediction and the actual inflow values, where the loss function is the Mean Squared Error (MSE).

Hyperparameter Tuning

To control the training process hyperparameters were used to impact the model performance. Unlike model parameters that are learned during training, hyperparameters are preset. Therefore, selecting suitable hyperparameters is vital for the success of the model.

The PBT approach was employed in combination with the Ray Tune library [5] for hyperparameter tuning. PBT, as explained above, is a dynamic approach to hyperparameter optimization, where a population of models is trained concurrently. During the training process, underperforming models are periodically replaced with mutated versions of top-performing models, thereby combining the benefits of both genetic algorithms and hand-tuning.

It is important to note that a hyperparameter search was conducted for each variable set to account for their unique characteristics and impact on the inflow prediction. This ensured that the hyperparameters were optimally tuned for each specific variable set, leading to more accurate and reliable forecasting models.

The key hyperparameters tuned in the model search include:

- **Sequence Length**: This parameter defines the length of the past sequence data that the model takes as input to make the predictions. A sequence length of 25 was explored.

- **Batch Size**: This parameter denotes the number of training examples utilized in one iteration. A batch size of 256 was experimented with.

- **Hidden Size**: The number of LSTM units in each LSTM layer is set by the hidden size. Two different sizes were tried: 32 and 64.
- **Number of Layers**: This parameter indicates the depth of the LSTM network. 1, 2, and 3 layer models were examined.
- **Learning Rate**: This is one of the most crucial hyperparameters, controlling how much to adjust the model in response to the estimated error each time the model weights are updated. A log-uniform distribution between $1e-5$ and $1e-1$ was used for the learning rate.
- **Weight Decay**: This parameter is a regularization technique, and it adds a small penalty, to the loss function to prevent overfitting. A log-uniform distribution between $1e-5$ and $1e-1$ was also used for the weight decay.

The hyperparameter tuning process was conducted for a maximum of 100 epochs, with a perturbation interval of 25 epochs. This means that every 25 epochs, the PBT scheduler would examine all the models in the population, and replace the bottom-performing models with the top-performing ones, while also applying some mutations to their hyperparameters. The specific code used for this process is included in Appendix A.1.

Model Evaluation

The model evaluation stage primarily assesses the performance of the trained model in terms of its predictive accuracy and generalizability. The model was evaluated based on its performance on the test set, which is unseen data that was not used during training or validation. This helps to ensure an unbiased evaluation of the model's forecasting performance and its ability to generalize to new data. The specific code used for this evaluation process is detailed in Appendix A.2.

The primary metrics used for model evaluation in this study are the Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), and Coefficient of Determination (R^2). For a more in-depth discussion about these evaluation metrics, please refer to Paper 2 in Chapter 7.

Beyond these metrics, another important aspect of model evaluation was the assessment of computational efficiency. This was accomplished by monitoring the space and time consumption of each model during the training and evaluation phases. Computational efficiency is a critical factor in real-world applications, especially in scenarios where rapid forecasting is required. Therefore, a model that achieves a balance between predictive accuracy and computational efficiency is considered ideal.

Chapter 5

Overview of Part II

The main part of this thesis are two distinct papers that explore the domain of hydrological forecasting by employing data-driven methods and spatial-temporal attention-based LSTM. The purpose of this chapter is to place these papers within the context of the current understanding. The section titles correspond directly to the titles of the included papers.

5.1 Literature Review of Data-Driven Methods Used for Forecasting Hydrological Events

5.1.1 Context

In recent years, the field of hydrological forecasting has seen remarkable growth, triggered by the advancements in data-driven models and an increase in computational resources. This literature review chronologically analyzes the development of data-driven models, from traditional to ML and hybrid approaches. Each of these categories carries its unique advantages and limitations, and these are systematically explored in the context of hydrological forecasting. Furthermore, this review goes into the trends that enhance prediction accuracy, considering univariate, multivariate, and spatial data components. By highlighting the challenges and advancements related to integrating temporal and spatial data, extracting valuable features, and employing attention mechanisms, this review sets the stage for future research directions.

5.1.2 Results

The literature review reveals that conventional data-driven models such as ARMA, ARIMA, and MLR, while reliable, struggle with capturing nonlinear relationships and forecasting non-stationary time series. This is where ML models show superiority due to their flexible handling of such relationships. However, even ML methods face challenges, particularly when integrating spatial data. It is also found that hybrid models, which incorporate both temporal and spatial data, can potentially enhance forecasting accuracy.

The review also highlights the significance of incorporating additional data sources to improve a model's accuracy. Despite the substantial progress made in hydrological forecasting, there are room for further research, particularly in areas like spatio-temporal data integration, feature extraction, and attention mechanism applications. These insights and identified trends aim to drive the hydrological forecasting field towards developing models that are more accurate and robust.

5.2 A Hydrological Case Study of Short-Term Inflow Forecasting with Spatial-Temporal Attention-Based LSTM

5.2.1 Context

The complexity of spatial and temporal interactions within catchment and meteorological data presents a significant challenge in hydrological forecasting. In an effort to address this, a spatial-temporal attention-based LSTM model was developed for hydrological inflow forecasting. The model, leveraging both spatial and temporal information, aimed to capture dependencies within the data. In this case study, the model's performance was evaluated against various baseline models encompassing traditional statistical techniques and deep learning methods. Two distinct forecast horizons were considered: one-hour ahead and recursive 12-hour ahead forecasts.

5.2.2 Results

The results revealed that the spatial-temporal attention-based LSTM model, while slightly under-performing against LSTM with temporal attention in one-hour ahead forecasting, outperformed all baseline models for the 12-hour ahead forecasting task. This outcome emphasizes that there might be an importance of spatial correlations for longer forecasting horizons.

Despite the increased complexity and longer training times of the proposed model, its size was smaller than the LSTM and LSTM with temporal attention models. This suggests that the additional complexity of integrating spatial-temporal attention mechanisms does not substantially increase memory consumption. The spatial-temporal attention-based LSTM model's efficacy in short-term inflow forecasting is promising, but it also opens up several avenues for future research, including evaluating performance across different catchment areas, extending the forecasting horizon, and exploring other attention mechanisms.

Part II

Papers

Chapter 6

Literature Review of Data-Driven Methods Used for Forecasting Hydrological Events

Literature Review of Data-Driven Methods Used for Forecasting Hydrological Events

Magnus Børseth

Abstract— **The field of hydrological forecasting has experienced rapid development in recent years, driven by advancements in data-driven models combined with increased computable resources. This literature review presents a chronological analysis of popular conventional, machine learning, and hybrid data-driven models, showcasing the current state of the art. The paper explores the advantages and disadvantages of each category, highlighting trends that contribute to improved prediction accuracy, including the use of univariate, multivariate, and spatial data components. By examining the challenges and improvements associated with integrating temporal and spatial data, effective feature extraction methods, and utilizing attention mechanisms, this review provides insights for future research directions. These directions seek to enhance the accuracy of hydrological forecasting models, enabling better-informed decision-making and improved management of water resources.**

I. INTRODUCTION

Forecasting hydrological events is a crucial research area in the field of hydrology, and an important tool used in decision making in areas such as flood management, climate change mitigation, and hydropower [1], [2]. Accurate and reliable inflow forecasting is essential, and improvements to predictions will contribute to analysis, water resource management and hazard assessment [3]. This has led to much attention on the field of hydrological forecasting and research to find effective strategies for improving the utilization of water resources [4].

When developing hydrological forecasting models, it is crucial to consider the forecasting time scale. Forecasts are generally categorized based on their forecasting horizon into long-term, mid-term, and short-term forecasts [2]. The forecasting horizons for these categories are typically 1-5 years, 1-52 weeks, and 1-7 days, respectively. Each category poses its own challenges. Short-term forecasting is computationally demanding, as the forecast is needed within a short period to be useful [5]. On the other hand, long-term forecasting is challenging due to the long-term dependencies involved [6]. Therefore, when developing hydrological forecasting methods, it is important to strike a balance between model complexity and accuracy for the selected time horizon.

Another important consideration is the data utilized to train the model. With the availability of a large amount of data and increased computational capabilities [4], there are new opportunities to improve and optimize current hydrological forecasting methods. These methods involve univariate, multivariate, and spatial components to better capture the complex hydrological processes. However, selecting relevant information from the available data is challenging, as the main sources of inflow come from various sources, such as precipitation, streams, and snow melt [6]. Hence, careful selection and extraction of useful data is critical to ensure accurate and reliable hydrological forecasting.

Hydrological forecasting methods can be broadly classified into two categories: physical-driven models and data-driven models [11], [12]. Physical-driven models, such as Hydrologiska Byråns Vattenbalansavdelning (HBV), have traditionally been used in hydrological forecasting [13]. However, these models require a significant amount of hydrological expertise, curated datasets, and high computational power to generate accurate predictions [3]. In contrast, data-driven models have gained more popularity due to their flexibility and ability to work with less data [14]. However, conventional data-driven models do have limitations in complex calculations and processing time, and they often only capture linear dependencies. To address these limitations, machine learning (ML) models have become increasingly popular in hydrological forecasting [3], [2]. ML models can incorporate non-linear dependencies based solely on historical data, allowing accurate predictions with minimal inputs [15]. Hybridization, which combines different techniques, has also shown significant promise in improving the performance of ML models [16].

Several studies have reviewed the use of data-driven models in hydrological forecasting, with a focus on specific events or techniques. For example, Bordin et al. [2] reviewed the use of ML in hydropower scheduling with a focus on short-term forecasting and cyber-physical systems. Mosavi et al. [3] compared flood prediction models using both short-term and long-term predictions with a focus on ML techniques, including hybrid models. Alawsi et al. [17] evaluated different pre-processing methods for drought forecasting based on appropriate time scales and data types. Zhang et al. [1] reviewed commonly used data-driven models for univariate streamflow forecasting in terms of forecasting performance and accuracy. However, there are currently limited research shedding light on general data-driven applications for all hydrological events with a focus on spatial data

This literature review explores different data-driven methods for hydrological forecasting, focusing on data usage and time horizons. The selected methods are classified into three categories based on their data-driven techniques. Section 2 introduces the state-of-the-art in hydrological event forecasting, ranging from conventional data-driven methods to ML methods and hybrid methods. Section 3 discusses potential areas for further research based on the reviewed literature. Finally, Section 4 presents the conclusions of the literature review.

TABLE I

AN OVERVIEW OF CONVENTIONAL DATA-DRIVEN MODELS FOR FORECASTING HYDROLOGICAL EVENTS. SHORT-TERM: 1-7 DAYS, MID-TERM: 1-52 WEEKS, LONG-TERM: 1-5 YEARS.

Method	Forecasting objective	Data Type	Time Scale	Article Year	Ref
ARMA, ARIMA	Inflow	Univariate	Mid-term	2012	[7]
ARMA, ARIMA	Inflow	Univariate	Mid-term	2013	[8]
MLR	Inflow	Multivariate	Short-term	2011	[9]
MLR	Water demand	Multivariate	Short-term	2012	[10]

II. STATE-OF-THE-ART IN FORECASTING HYDROLOGICAL EVENTS

This section provides an overview of the state-of-the-art in hydrological forecasting by presenting data-driven models in chronological order to demonstrate the evolution of forecasting in the hydro sector.

A. Conventional Data-Driven Models

The Auto Regressive Moving Average (ARMA) model is commonly used for hydrological event forecasting [18]. It combines autoregressive and moving average factors by incorporating residuals from previous lags to make future forecasts. However, to apply the model efficiently, the time series must be stationary [8]. The Auto Regressive Integrated Moving Average (ARIMA) model was introduced to address this by adding an integrated factor to handle non-stationary time series.

ARIMA has been widely used in hydrological forecasting, as seen in studies such as [7] and [8]. In [7], monthly discharge data from a 42-year period was used to compare the accuracy of ARMA and ARIMA for mid-term inflow forecasting. The study found that the accuracy of both models improved as the number of parameters increased, and that ARIMA was superior due to its ability to make the time series data stationary. Similarly, [8] compared ARMA and ARIMA against an autoregressive artificial neural network using monthly discharge data from Taleh Zang hydrometric station. The study concluded that both models' accuracy improved with an increase in the number of parameters.

Another conventional data-driven model is the Multiple Linear Regression (MLR) model, which is a simpler methodology for time series forecasting. It uses a statistical technique that models the relationship between two or more independent and dependent variables through the utilization of a linear regression equation that is applied to the gathered data [9].

The use of multivariate data has been explored in [9], where historical rainfall and inflow data were used to predict inflow using MLR. The study developed twenty models based on different input structure combinations, and it was concluded that the models with daily aggregated data performed better. In [10], multivariate MLR was compared with different ML methods for urban water demand forecasting in Canada. The study indicated that the primary factors influencing the water demand process during summer months are the maximum air temperature, while precipitation plays a lesser role in Montreal, Canada. However, it was also shown that MLR struggled to capture relationships of a pre-specified functional form and was not able to accurately predict nonlinear water demands.

Overall, conventional data-driven models such as ARMA, ARIMA, and MLR have been widely used in hydrological forecasting, and their accuracy have been improved with an increase in the number of parameters. However, these models have their limitations, especially in capturing nonlinear relationships and predicting non-stationary time series. Table I provides an overview of the studies discussed in this section.

B. Machine Learning Data-Driven Models

ML is a field of study focused on developing algorithms that can learn from previous experience to perform new tasks [3]. Unlike conventional models, ML algorithms are data-driven and do not require a prior understanding of the relationship between data and outcomes [19]. The ML modeling process typically involves data collection from reliable sources, followed by preprocessing and partitioning the data into training, validation, and testing sets. The model's architecture is then built and fine-tuned using the training and validation data. Finally, the model's accuracy is evaluated on unseen testing data.

In hydrological forecasting, ML models such as Support Vector Machines (SVMs), Artificial Neural Networks (ANNs), and Recurrent Neural Networks (RNNs) have shown promising results, as discussed in [20] and summarized in Table II.

1) Support Vector Machine (SVM): The support vector machine (SVM) is a robust approach utilized for both classification and regression tasks. SVM is commonly used for classification problems, where it optimizes the range of differences between groups, but it can also be adapted for regression problems by transforming the data. SVM has some limitations, including difficulty handling big data, complex nonlinear problems, and sensitivity to missing data. Further details on the SVM model can be found in the literature [30].

Studies have shown the potential of SVM for forecasting tasks in the field of hydrology. For example, in [23], SVM was used to predict short-term flood forecasts in the Kelantan River, Malaysia, and achieved good performance. In [21], a modified SVM model was developed to predict mid-term inflow at the Shihmen Reservoir in Taiwan, using a combination of inflow and climate data from prior periods. The study demonstrated that SVM, combined with genetic

TABLE II
AN OVERVIEW OF MACHINE LEARNING MODELS FOR FORECASTING HYDROLOGICAL EVENTS. SHORT-TERM: 1-7 DAYS, MID-TERM: 1-52 WEEKS, LONG-TERM: 1-5 YEARS.

Method	Forecasting objective	Data Type	Time Scale	Article Year	Ref
SVM	Inflow	Multivariate	Mid-term	2010	[21]
SVM	Inflow	Univariate	Mid-term	2014	[22]
SVM	Flood	Univariate	Short-term	2021	[23]
ANN	Inflow	Multivariate	Short-term	2022	[24]
ANN	Inflow	Multivariate	Mid-term	2021	[25]
ANN	Inflow	Multivariate	Mid-term	2016	[26]
RNN	Inflow	Univariate	Short-term	2020	[27]
LSTM	Flood	Multivariate	Short-term	2019	[28]
LSTM,CNN	Inflow	Multivariate	Long-term	2021	[6]
SVM, RF, MLP, DNN, RNN, LSTM, GRU	Inflow	Multivariate	Short-term	2022	[29]
SVM, ANN, LSTM	Inflow	Univariate	Short-term	2021	[20]

algorithms, outperformed other models in predicting mid-term inflow, while also highlighting the importance of incorporating climate information into the prediction model. Another modification to the SVM model was presented in [22], which combined empirical mode decomposition with SVM to predict monthly streamflow. The study showed that this approach produced better performance than a neural network model. By incorporating additional features of different resolutions related to non-linear and non-stationary monthly streamflow, the modified SVM model was able to improve overall performance.

2) Artificial Neural Network (ANN): ANN was initially designed to simulate the biological neural system in the human brain to make decisions in a human-like manner. ANN is also known as a feed forward neural network because the inputs are processed in a forward direction through the network. It is made up of many interconnected computing nodes, which are separated into three types of layers: input, hidden and output. The input layer receives the input data and then sends it to the hidden layers. It is then processed in the hidden layer, and there is flexibility in the number of hidden layers that can be employed. Lastly the model gathers the computations performed on the output layer, which gives the result.

These computations are enabled by activation functions associated with each neuron, which introduce non-linearity into the system. This feature equips the ANN with the ability to make human-like decisions. However, designing the architecture can be challenging, as a small architecture may be insufficient to capture the complexity of the data, leading to underfitting, while a large one may overfit the data, resulting in poor generalization to new or unseen data. For further insight into ANN structures, see [31].

ANN has been used in various hydrological applications, such as inflow forecasting in Bang Lang Dam [24]. In this study, daily water inflow, rainfall, atmospheric pressure, humidity, and temperature were collected to analyze the factors affecting the inflow of water in the dam. The ANN deep learning model was applied for water inflow forecasting, and the results were found to be satisfactory. In [25], ANN was developed to forecast weekly reservoir inflows, exploring the impact of input and parameter uncertainty in the forecasted inflow values. A simulation-optimization framework was proposed, and it was found that the operational framework outperformed the planning framework it was compared with. In [26], the researchers employed a dynamic neural network methodology to predict monthly reservoir inflow. The study concluded that the approach was suitable for monthly inflow forecasting but suggested that the accuracy could be improved by fitting the model for de-seasonalized data.

3) Recurrent Neural Network (RNN): The Recurrent Neural Network (RNN) is a commonly used model in deep learning for handling sequential data. RNNs modify the traditional neural network to incorporate dependencies between data points through memory, which stores information from the previous input to make the next prediction. However, RNNs struggle with vanishing and exploding gradients, making it difficult to capture long-term dependencies in data. Other recurrent models like Gated Recurrent Unit (GRU) and Long Short Term Memory (LSTM) have been developed to address these issues.

Studies have been conducted to compare RNNs against traditional ANNs in reservoir inflow forecasting. The comparative analysis in [27] shows that RNNs outperform ANNs, with the LSTM model having the best overall performance. LSTM has also been used in flood forecasting, as seen in [28], where it performed reliably in capturing long-term dependencies. In [6], LSTM was used with historical snow water equivalent and inflow as inputs for long-term inflow forecasting, and the results were promising. Further advancements have been made in integrating multiple ML techniques for real-time inflow forecasting, such as in [29], where ensemble means and switched prediction methods were used to ensure stability. GRU was found to have the best performance among the seven models tested. Finally, a comparison of deep learning algorithms in [20] showed that LSTM outperformed ANN and SVM in inflow forecasting.

C. Hybrid Data-Driven Models

While many ML methods perform well with one task or dataset, hybrid models offer new ways for simple algorithms to work together to solve more complex problems. However, the selection of techniques can be challenging, as different models handle data differently. Data-driven modeling often includes univariate and multivariate models that consider only temporal information from the data. Nevertheless, recent studies have shown that incorporating spatial information can greatly improve prediction accuracy [32]. In Table III, various hybrid methods are presented, ordered by data type, for solving hydrological event forecasting problems.

Univariate models are useful for forecasting the future values of a single scalar based on a sequence of historical data. However, the use of hybrid models can significantly improve the accuracy of hydrological forecasting by reducing uncertainty. One example of a hybrid univariate model can be seen in Fig. 1, which is based on the model used in [33]. In this model, the forecasting ANN receives a sub-series obtained by applying wavelet transformation instead of the observed streamflow as input. The signal is then decomposed into different resolution levels to filter the information into low-frequency content (approximations) and high-frequency content (details), which are then utilized for forecasting.

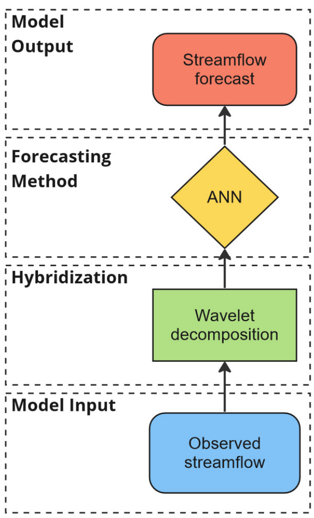

Fig. 1. Hybrid univariate model for streamflow forecasting using wavelet decomposition and ANN.

Other hybrid univariate methods used for improving hydrological forecasting can be seen in [34]. They utilize decomposition-ensemble learning and incorporate it with LSTM for forecasting, along with a logarithmic transformation as a pre-processing method to handle non-stationary inflow data. The proposed method outperformed other compared models for reservoir inflow forecasting. In [35], a Multiscale Deep Feature Learning (MDFL) method with hybrid models is proposed for daily reservoir inflow forecasting. Here, multiscale features are extracted using EEMD and Fourier spectrum, and then sent to three Deep Belief Networks (DBN), and the outputs are reconstructed using a sum-up strategy to get the forecasting result. Another study in [36] used univariate data for mid-term inflow forecasting and compared an ANN genetic algorithm called ANN-GA with a seasonal ARIMA model (SARIMA). It was found that the SARIMA model was more precise in predicting peak values and more suitable for drought years and low flow prediction, while the ANN-GA model was more suitable for wet years and flood forecasting.

Multivariate models aim to improve the forecasting capabilities by utilizing information from multiple variables and selecting the relevant ones. One approach is shown in Fig. 2, adapted from [37], where inflow, rainfall, and ERA-Interim reanalysis data are used with Gradient Boosting Regression Tree (GBRT). Here, the feature selection of the reanalysis dataset is done using Maximal Information Coefficient (MIC), while Partial Autocorrelation Function (PACF) and Cross-correlation Function (CCF) are used for selecting inflow and rainfall. The study concluded that GBRT-MIC can result in enhanced precision and dependability for inflow forecasting, while the reanalysis data chosen by the MIC significantly enhances the performance of the forecasts.

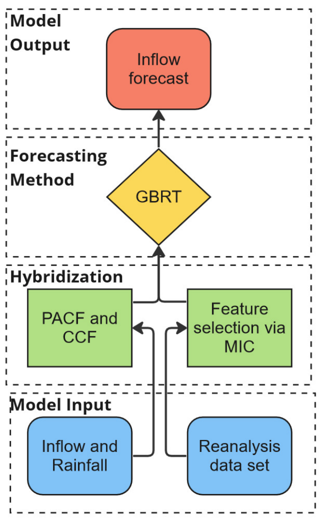

Fig. 2. Illustration of a hybrid multivariate model utilizing GBRT-MIC for inflow forecasting.

Other multivariate hydrological forecasting methods have also been developed, such as [38], which proposed a hybrid model that integrates an enhanced version of Ensemble Empirical Mode Decomposition with Additive Noise (ICEEMDAN), Sample Entropy (SE), Gini Index (GI), and sequence-to-sequence (Seq2Seq) methodologies. The ICEEMDAN technique is employed to decompose the streamflow time series into distinct subcomponents to manage trends and noise, while SE and GI are employed to reduce computational cost. Seq2Seq is a LSTM-based model that handles temporal dependencies. The study showed that this hybrid approach is viable for streamflow forecasting.

Another approach is the deep generative neural network proposed in [39], consisting of a stochastic RNN, a latent variable inference network with normalizing flows, and an attention-based learning network called F2F. The stochastic RNN is utilized to capture temporal dependencies while considering the uncertainty of the hidden state. In order to achieve a higher level of precision and capture intricate distributions, a residual flow network is applied to the output of the stochastic RNN. Additionally, a self-attention network is employed to identify and filter out less relevant historical observations. The proposed F2F model achieved the best performance among the probabilistic models, but with a higher computational cost. In [40], a wavelet-ANN hybrid model was proposed, in which the Tropical Rainfall Measuring Mission (TRMM) data were integrated with inflow data, employing wavelet transformations to enhance performance. This decomposition technique was shown to be powerful in improving the model's performance.

Building on the multivariate forecasting approaches, [41] used Principal Component Analysis (PCA) and Long Short-Term Memory (LSTM) to increase the accuracy of inflow predictions in hydropower generation. The PCA was used to reduce data dimensionality, thereby speeding up the training process. This method improved prediction accuracy when hydro-meteorological data were included in the model.

The Causal Empirical Decomposition (CED) method used by [42] significantly enhanced the accuracy of inflow forecasts. CED combines physics-based causal inference with signal processing-based decomposition to select the most relevant features for forecasting. CED's integration with various forecasting methods indicated its potential as an adaptable, effective preprocessing step.

Following a similar approach, [43] proposed the Causal Variational Mode Decomposition (CVD), another feature selection framework built on multiresolution analysis and causal inference. Applied as a preprocessing step to any machine learning-based forecasting method, CVD was able to reduce computation time while boosting the forecasting accuracy. The technique showed promise when validated on a river system downstream of a Norwegian hydropower reservoir, particularly when combined with LSTM, indicating its potential for application to other cascaded water systems.

The application of classical data mining techniques for univariate and multivariate data has been widely studied. However, when it comes to spatio-temporal data, these methods usually perform poorly due to the high complexity and correlations among the data variables [32]. To address this issue, researchers have developed novel approaches that can handle spatio-temporal data more effectively. Fig. 3, adapted from [44], shows how a Dynamic Spatio-temporal Attention method (DSTA) can be used to improve streamflow forecasting. This method uses four features of meteorological data from different stations and hydrological streamflow data from one station, and consists of three modules: a spatial attention module, a temporal attention module, and a trend module. The spatial attention module captures spatial correlation in

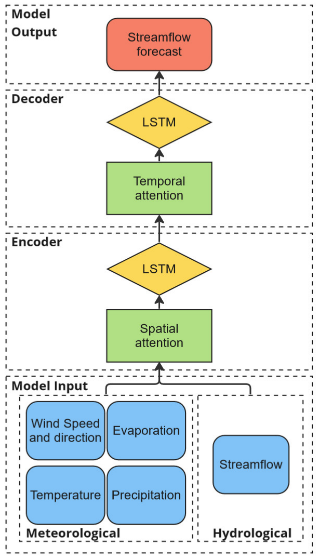

Fig. 3. Spatio-temporal model incorporating DSTA for improved streamflow forecasting.

the meteorological data, while the temporal attention module selects the corresponding time step for streamflow prediction, and the trend module takes into account the influence of future factors in order to enhance the conventional LSTM model.

Similarly, in [45], the authors proposed a spatio-temporal attention LSTM model (STA-LSTM) for flood forecasting using multiple rainfall and streamflow stations. The attention-based models were found to be more accurate than the original LSTM model and outperformed novel models such as convolutional neural networks and Graph Convolutional Network (GCN) models, which also have some benefits when working with spatial-temporal data. In another study, [46], the authors combined GCN and LSTM with attention to create the hybrid ST-GCN model. The GCN module captures the spatial correlation of each hydrological feature, while the LSTM module captures the temporal information. The utilization of the attention method enhances the choice of the ST-GCN network when incorporating historical information. These methods have shown promising results for spatio-temporal data analysis, demonstrating their potential for practical applications.

III. FURTHER WORK

The advancements in hydrological forecasting using hybrid models have demonstrated promising results. However, there are still areas in which further work can be conducted to improve the forecasting capabilities, particularly in the context of deep learning architectures for spatio-temporal forecasting, feature extraction methods for studying spatial and temporal dependencies within hydro-power data, and combining spatio-temporal meteorological and hydrological data to enhance inflow forecasting.

TABLE III

AN OVERVIEW OF HYBRID MODELS FOR FORECASTING HYDROLOGICAL EVENTS. SHORT-TERM: 1-7 DAYS, MID-TERM: 1-52 WEEKS, LONG-TERM: 1-5 YEARS.

Method	Forecasting objective	Data Type	Time Scale	Article Year	Ref
DEL-LSTM	Inflow	Univariate	Short-term	2019	[34]
MDFL	Inflow	Univariate	Short-term	2016	[35]
ANN-GA, SARIMA	Inflow	Univariate	Mid-term	2017	[36]
GBRT-MIC	Inflow	Multivariate	Short-term	2020	[37]
ICEEMDAN-SE-GI-Seq2Seq	Streamflow	Multivariate	Short-term	2021	[38]
F2F	Inflow	Multivariate	Short-term	2022	[39]
WA-ANN	Inflow	Multivariate	Short-term	2019	[40]
PCA-LSTM	Inflow	Multivariate	Short-term	2021	[41]
CED-LSTM	Inflow	Multivariate	Short-term	2022	[42]
CVD-LSTM	Inflow	Multivariate	Short-term	2023	[43]
DSTA	Streamflow	Spatio-temporal	Short-term	2019	[44]
STA-LSTM	Flood forecasting	Spatio-temporal	Short-term	2020	[45]
ST-GCN	Flood forecasting	Spatio-temporal	Short-term	2021	[46]

Despite the significant improvements in hydrological forecasting achieved through the use of additional data, there is a scarcity of studies focused on using spatio-temporal data and extracting relevant information from it. With the increased availability of data from sensors in catchment areas [2], exploring methods for extracting spatial and temporal information becomes crucial for advancing hydrological event forecasting.

One area of further work involves investigating deep learning architectures specifically designed for spatio-temporal forecasting. Developing deep learning architectures that can effectively process spatio-temporal data could yield better forecasting performance in hydrological applications.

Another aspect worth investigating involves identifying feature extraction methods that are better suited for analyzing spatial and temporal relationships in various hydro-power datasets. Attention mechanisms have been applied in various domains, including image classification [47] and natural language processing tasks [48], as well as in some hydrological forecasting literature, as reviewed above. However, there is still potential for further exploration and adaptation of attention mechanisms to better address the challenges specific to hydro-power data. By learning the inner dependencies and highlighting the most important information, attention mechanisms can enhance the sensitivity to features containing relevant information for hydrological forecasting. Further research could investigate new variants of attention mechanisms or their integration with other methods to improve the forecasting performance in this context.

Lastly, further research can focus on combining spatio-temporal meteorological and hydrological data to improve inflow forecasting. By effectively leveraging both types of data, researchers can develop more accurate and robust models for predicting hydrological events. This approach can also aid in better understanding the underlying relationships between meteorological and hydrological variables, leading to more informed decision-making in water resource management.

IV. CONCLUSION

This paper provides an overview of the hydrological forecasting field, which has been undergoing rapid development. A chronological analysis of various data-driven models has been presented, showcasing the current state-of-the art. These models were categorized into three groups: conventional, machine learning (ML), and hybrid data-driven models. The advantages and disadvantages of each category were discussed in detail, and summarized in table I, II, and III.

Several trends have been identified in literature that contribute to improved prediction accuracy. First, conventional data-driven models such as ARMA, ARIMA, and MLR have limitations, especially in capturing nonlinear relationships and predicting non-stationary time series, whereas ML models can handle these relationships more flexibly. Second, while ML methods can effectively manage temporal data, they still face challenges when dealing with spatial data. Lastly, the use of hybrid methods has been shown to enhance forecasting accuracy by incorporating both temporal and spatial data.

Moreover, incorporating additional data sources can significantly improve a model's accuracy. Despite the advances in hydrological forecasting, there is still room for further research, particularly in the areas of spatio-temporal data integration, feature extraction, and the application of attention mechanisms. By continuing to explore and refine these methods, the hydrological forecasting field can progress toward more accurate and robust models.

REFERENCES

[1] Z. Zhang, Q. Zhang, and V. P. Singh, "Univariate streamflow forecasting using commonly used data-driven models: Literature review and case study," *Hydrological Sciences Journal*, vol. 63, no. 7, pp. 1091–1111, May 2018.

[2] C. Bordin, H. I. Skjelbred, J. Kong, and Z. Yang, "Machine Learning for Hydropower Scheduling: State of the Art and Future Research Directions," *Procedia Computer Science*, vol. 176, pp. 1659–1668, Jan. 2020.

[3] A. Mosavi, P. Ozturk, and K.-w. Chau, "Flood Prediction Using Machine Learning Models: Literature Review," *Water*, vol. 10, no. 11, p. 1536, Nov. 2018.

[4] W.-j. Niu and Z.-k. Feng, "Evaluating the performances of several artificial intelligence methods in forecasting daily streamflow time series for sustainable water resources management," *Sustainable Cities and Society*, vol. 64, p. 102562, Jan. 2021.
[5] P. C. Nayak, K. P. Sudheer, D. M. Rangan, and K. S. Ramasastri, "Short-term flood forecasting with a neurofuzzy model," *Water Resources Research*, vol. 41, no. 4, 2005.
[6] Z. C. Herbert, Z. Asghar, and C. A. Oroza, "Long-term Reservoir Inflow Forecasts: Enhanced Water Supply and Inflow Volume Accuracy Using Deep Learning," *Journal of Hydrology*, vol. 601, p. 126676, Oct. 2021.
[7] Samir, "Parameters Estimate of Autoregressive Moving Average and Autoregressive Integrated Moving Average Models and Compare Their Ability for Inflow Forecasting," *Journal of Mathematics and Statistics*, vol. 8, no. 3, pp. 330–338, Mar. 2012.
[8] M. Valipour, M. E. Banihabib, and S. M. R. Behbahani, "Comparison of the ARMA, ARIMA, and the autoregressive artificial neural network models in forecasting the monthly inflow of Dez dam reservoir," *Journal of Hydrology*, vol. 476, pp. 433–441, Jan. 2013.
[9] R. B. MAGAR and V. JOTHIPRAKASH, "Intermittent reservoir daily-inflow prediction using lumped and distributed data multi-linear regression models," *Journal of Earth System Science*, vol. 120, no. 6, pp. 1067–1084, Dec. 2011.
[10] J. Adamowski, H. Fung Chan, S. O. Prasher, B. Ozga-Zielinski, and A. Sliusarieva, "Comparison of multiple linear and nonlinear regression, autoregressive integrated moving average, artificial neural network, and wavelet artificial neural network methods for urban water demand forecasting in Montreal, Canada," *Water Resources Research*, vol. 48, no. 1, 2012.
[11] H. Sanner, J. Harlin, and M. Persson, *Application of the HBV Model to the Upper Indus River for Inflow Forecasting to the Tarbela Dam*. SMHI, 1994.
[12] M. Zounemat-Kermani, A. Mahdavi-Meymand, and R. Hinkelmann, "A comprehensive survey on conventional and modern neural networks: Application to river flow forecasting," *Earth Science Informatics*, vol. 14, no. 2, pp. 893–911, Jun. 2021.
[13] J. Seibert and S. Bergström, "A retrospective on hydrological catchment modelling based on half a century with the HBV model," *Hydrology and Earth System Sciences*, vol. 26, no. 5, pp. 1371–1388, Mar. 2022.
[14] N. M. Khairudin, N. Mustapha, T. N. M. Aris, and M. Zolkepli, "IN-DEPTH REVIEW ON MACHINE LEARNING MODELS FOR LONG-TERM FLOOD FORECASTING," no. 10, p. 19, 2022.
[15] K. S. M. H. Ibrahim, Y. F. Huang, A. N. Ahmed, C. H. Koo, and A. El-Shafie, "A review of the hybrid artificial intelligence and optimization modelling of hydrological streamflow forecasting," *Alexandria Engineering Journal*, vol. 61, no. 1, pp. 279–303, Jan. 2022.
[16] F. Fotovatikhah, M. Herrera, S. Shamshirband, K.-w. Chau, S. Faizollahzadeh Ardabili, and M. J. Piran, "Survey of computational intelligence as basis to big flood management: Challenges, research directions and future work," *Engineering Applications of Computational Fluid Mechanics*, vol. 12, no. 1, pp. 411–437, Jan. 2018.
[17] M. A. Alawsi, S. L. Zubaidi, N. S. S. Al-Bdairi, N. Al-Ansari, and K. Hashim, "Drought Forecasting: A Review and Assessment of the Hybrid Techniques and Data Pre-Processing," *Hydrology*, vol. 9, no. 7, p. 115, Jul. 2022.
[18] S. Mehdizadeh, "Using AR, MA, and ARMA Time Series Models to Improve the Performance of MARS and KNN Approaches in Monthly Precipitation Modeling under Limited Climatic Data," *Water Resources Management*, vol. 34, no. 1, pp. 263–282, Jan. 2020.
[19] C. Ley, R. K. Martin, A. Pareek, A. Groll, R. Seil, and T. Tischer, "Machine learning and conventional statistics: Making sense of the differences," *Knee Surgery, Sports Traumatology, Arthroscopy*, vol. 30, no. 3, pp. 753–757, Mar. 2022.
[20] S. D. Latif, A. N. Ahmed, E. Sathiamurthy, Y. F. Huang, and A. El-Shafie, "Evaluation of deep learning algorithm for inflow forecasting: A case study of Durian Tunggal Reservoir, Peninsular Malaysia," *Natural Hazards*, vol. 109, no. 1, pp. 351–369, Oct. 2021.
[21] P.-H. Li, H.-H. Kwon, L. Sun, U. Lall, and J.-J. Kao, "A modified support vector machine based prediction model on streamflow at the Shihmen Reservoir, Taiwan," *International Journal of Climatology*, vol. 30, no. 8, pp. 1256–1268, 2010.
[22] S. Huang, J. Chang, Q. Huang, and Y. Chen, "Monthly streamflow prediction using modified EMD-based support vector machine," *Journal of Hydrology*, vol. 511, pp. 764–775, Apr. 2014.
[23] A. Faruq, A. Marto, and S. Abdullah, "Flood Forecasting of Malaysia Kelantan River using Support Vector Regression Technique," *Computer Systems Science and Engineering*, vol. 39, no. 3, pp. 297–306, 2021.
[24] A. Benhawan, S. Matayong, and C. Choksuchat, "Forecasting Water Inflow for Bang Lang Dam," in *2022 IEEE 12th Annual Computing and Communication Workshop and Conference (CCWC)*, Jan. 2022, pp. 0229–0233.
[25] K. S. Kasiviswanathan, K. P. Sudheer, B.-S. Soundharajan, and A. J. Adeloye, "Implications of uncertainty in inflow forecasting on reservoir operation for irrigation," *Paddy and Water Environment*, vol. 19, no. 1, pp. 99–111, Jan. 2021.
[26] N. Basnayake, D. Attygalle, L. Liyanage, and K. D. W. Nandalal, "Ensemble Forecast for monthly Reservoir Inflow; A Dynamic Neural Network Approach," Jan. 2016.
[27] H. Apaydin, H. Feizi, M. T. Sattari, M. S. Colak, S. Shamshirband, and K.-W. Chau, "Comparative Analysis of Recurrent Neural Network Architectures for Reservoir Inflow Forecasting," *Water*, vol. 12, no. 5, p. 1500, May 2020.
[28] X.-H. Le, H. V. Ho, G. Lee, and S. Jung, "Application of Long Short-Term Memory (LSTM) Neural Network for Flood Forecasting," *Water*, vol. 11, no. 7, p. 1387, Jul. 2019.
[29] I.-H. Huang, M.-J. Chang, and G.-F. Lin, "An optimal integration of multiple machine learning techniques to real-time reservoir inflow forecasting," *Stochastic Environmental Research and Risk Assessment*, vol. 36, no. 6, pp. 1541–1561, Jun. 2022.
[30] Y. Gong, Y. Zhang, S. Lan, and H. Wang, "A Comparative Study of Artificial Neural Networks, Support Vector Machines and Adaptive Neuro Fuzzy Inference System for Forecasting Groundwater Levels near Lake Okeechobee, Florida," *Water Resources Management*, vol. 30, no. 1, pp. 375–391, Jan. 2016.
[31] M. Kale, M. Nagdeve, and S. Wadatkar, "Reservoir Inflow Forecasting using Artificial Neural Network," *Hydrology Journal*, vol. 35, no. 1and2, p. 52, 2012.
[32] S. Wang, J. Cao, and P. S. Yu, "Deep Learning for Spatio-Temporal Data Mining: A Survey," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 8, pp. 3681–3700, Aug. 2022.
[33] C. A. Guimarães Santos and G. B. L. da Silva, "Daily streamflow forecasting using a wavelet transform and artificial neural network hybrid models," *Hydrological Sciences Journal*, vol. 59, no. 2, pp. 312–324, Feb. 2014.
[34] Y. Qi, Z. Zhou, L. Yang, Y. Quan, and Q. Miao, "A Decomposition-Ensemble Learning Model Based on LSTM Neural Network for Daily Reservoir Inflow Forecasting," *Water Resources Management*, vol. 33, no. 12, pp. 4123–4139, Sep. 2019.
[35] Y. Bai, Z. Chen, J. Xie, and C. Li, "Daily reservoir inflow forecasting using multiscale deep feature learning with hybrid models," *Journal of Hydrology*, vol. 532, pp. 193–206, Jan. 2016.
[36] H. Moeeni, H. Bonakdari, S. E. Fatemi, and A. H. Zaji, "Assessment of Stochastic Models and a Hybrid Artificial Neural Network-Genetic Algorithm Method in Forecasting Monthly Reservoir Inflow," *INAE Letters*, vol. 2, no. 1, pp. 13–23, Apr. 2017.
[37] S. Liao, Z. Liu, B. Liu, C. Cheng, X. Jin, and Z. Zhao, "Multistep-ahead daily inflow forecasting using the ERA-Interim reanalysis data set based on gradient-boosting regression trees," *Hydrology and Earth System Sciences*, vol. 24, no. 5, pp. 2343–2363, May 2020.
[38] H. Apaydin and M. Sibtain, "A multivariate streamflow forecasting model by integrating improved complete ensemble empirical mode decomposition with additive noise, sample entropy, Gini index and sequence-to-sequence approaches," *Journal of Hydrology*, vol. 603, p. 126831, Dec. 2021.
[39] F. Zhou, Z. Wang, D. Chen, and K. Zhang, "Reservoir Inflow Forecasting in Hydropower Industry: A Generative Flow-based Approach," *IEEE Transactions on Industrial Informatics*, pp. 1–1, 2022.
[40] C. A. G. Santos, P. K. M. M. Freire, R. M. da Silva, and S. A. Akrami, "Hybrid Wavelet Neural Network Approach for Daily Inflow Forecasting Using Tropical Rainfall Measuring Mission Data," *Journal of Hydrologic Engineering*, vol. 24, no. 2, p. 04018062, Feb. 2019.

[41] X. Cheng, H. Farahmand, M. Yousefi, R. Arghandeh, and H. Wang, "Inflow Forecasting Based On Principal Component Analysis and Long Short Term Memory," in *2021 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech)*, pp. 589–596.

[42] M. Yousefi, X. Cheng, M. Gazzea, A. H. Wierling, J. Rajasekharan, A. Helseth, H. Farahmand, and R. Arghandeh, "Day-ahead inflow forecasting using causal empirical decomposition," vol. 613, p. 128265. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S002216942200837X

[43] M. Yousefi, J. Wang, Fandrem Høivik, J. Rajasekharan, A. Hubert Wierling, H. Farahmand, and R. Arghandeh, "Short-term inflow forecasting in a dam-regulated river in Southwest Norway using causal variational mode decomposition," vol. 13, no. 1, p. 7016. [Online]. Available: https://www.nature.com/articles/s41598-023-34133-8

[44] J. Feng, L. Yan, and T. Hang, "Stream-Flow Forecasting Based on Dynamic Spatio-Temporal Attention," *IEEE Access*, vol. 7, pp. 134 754–134 762, 2019.

[45] Y. Ding, Y. Zhu, J. Feng, P. Zhang, and Z. Cheng, "Interpretable spatio-temporal attention LSTM model for flood forecasting," *Neurocomputing*, vol. 403, pp. 348–359, Aug. 2020.

[46] J. Feng, Z. Wang, Y. Wu, and Y. Xi, "Spatial and Temporal Aware Graph Convolutional Network for Flood Forecasting," in *2021 International Joint Conference on Neural Networks (IJCNN)*, Jul. 2021, pp. 1–8.

[47] X. Mei, E. Pan, Y. Ma, X. Dai, J. Huang, F. Fan, Q. Du, H. Zheng, and J. Ma, "Spectral-Spatial Attention Networks for Hyperspectral Image Classification," *Remote Sensing*, vol. 11, no. 8, p. 963, Jan. 2019.

[48] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is All You Need," 2017.

Chapter 7

A Hydrological Case Study of Short-Term Inflow Forecasting with Spatial-Temporal Attention-Based LSTM

A Hydrological Case Study of Short-Term Inflow Forecasting with Spatial-Temporal Attention-Based LSTM

Magnus Børseth

***Abstract*— In hydrological forecasting, understanding the complex spatial and temporal interactions within catchment and meteorological data is crucial. This paper presents a spatial-temporal attention-based LSTM model for hydrological inflow forecasting. Taking advantage of both spatial and temporal information, the model captures complex dependencies within catchment and meteorological data. The performance of the proposed model is compared against several baseline models, including traditional statistical techniques and state-of-the-art machine learning methods, for one-hour ahead and recursive 12-hour ahead forecasts. The results demonstrate that while the proposed model slightly under-performs against LSTM with temporal attention in the one-hour ahead forecasting, it outperforms all the baseline models in the 12-hour ahead forecasting task. This emphasizes the significance of spatial correlations in longer forecasting horizons. This study contributes to the advancement of hydrological forecasting models by demonstrating the effectiveness of integrating spatial-temporal attention mechanisms with the LSTM networks.**

I. Introduction

Hydrological forecasting is a critical component of water resource management, providing essential information for flood control [1], drought management [2], and hydropower generation [3]. Accurate inflow forecasting is an important part of reservoir operations, as it enables optimization of water storage and release decisions in response to changing weather patterns and water demands [4]. In practice, making accurate predictions is difficult, because it requires a thorough understanding of the complex interaction between meteorological and hydrological factors, as well as the impact of human-controlled variables such as discharge from upstream reservoirs, which may experience time delays.

In recent years, data-driven models have emerged as a promising approach for inflow forecasting, taking advantage of machine learning and statistical techniques to extract patterns and relationships from historical data [4], [5]. Models such as Long Short-Term Memory (LSTM) have shown great potential for modeling temporal dependencies in hydrological time series, with applications ranging from rainfall-runoff modeling to flood forecasting [6], [7].

Through the examination of existing literature [8], it was found that several data-driven methods have been used for forecasting hydrological events. However, while LSTM models have shown great promise for temporal modeling, they struggle with spatial correlations and fail to capture the complex interactions between different locations within a catchment area [9]. To address this challenge, recent studies have proposed attention-based models that incorporate both spatial and temporal information to capture the complex spatial correlations [10], [11]. These models have shown promising results in improving inflow forecasting accuracy, showing the importance of using spatial and temporal information in hydrological modeling.

This paper presents a case study of inflow forecasting using a spatial-temporal attention-based LSTM model, using data from sensors in a catchment and meteorological area to improve inflow forecasting accuracy. The performance of the proposed model is evaluated by comparing it against traditional data-driven approaches and a generic LSTM-based model. The study contributes to the growing literature on the application of attention-based LSTM models for hydrological forecasting and highlights the importance of incorporating both spatial and temporal information in hydrological forecasting.

The rest of the paper is organized as follows. Section II provides an overview of the study area and the data used for the research. Section III discusses the data cleaning process to prepare the data for modeling. Section IV presents the proposed spatial-temporal attention-based LSTM model and its methodology. In Section V, , the experimental results are reported on and analyzed to assess the model's performance. Section VI draws conclusions from the findings. Lastly, Section VII suggests future research directions and potential improvements to the model.

II. Study Area and Data

The study area chosen for this research is located on the southwestern coast of Norway, in the Stavanger region. The catchment (Figure 1) encompasses a diverse topography and a temperate oceanic climate characterized by mild summers and winters, along with substantial rainfall throughout the year. The hydrological system within the catchment area comprises a network of rivers, streams, and reservoirs, which are influenced by various meteorological factors, such as precipitation, temperature, and evaporation.

The initial dataset used in this study consists of measurements collected at hourly intervals over various time periods from January 2015 to March 2022. However, due to sensor operation inconsistencies, not all variables cover the entire time range. This dataset contained a diverse set of hydrological and meteorological variables. After a process of data filtering, a more focused subset of data was selected, ranging from November 4, 2018, to September 22, 2020, with a count of 16500 data points.

Hydrological data includes discharge measurements and logger values obtained from multiple sensor locations. Meteorological data, encompassing air temperature, precipitation, humidity, and wind speed measurements, were collected

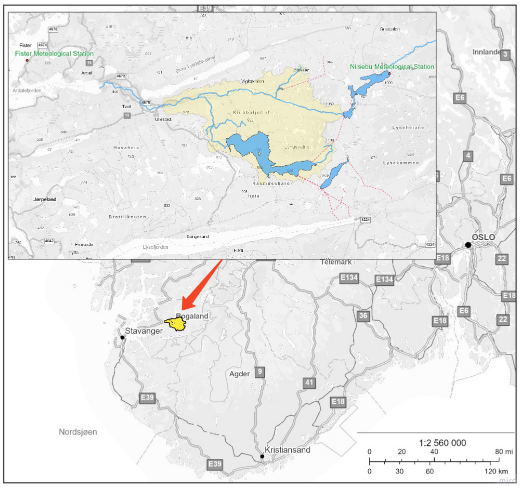

Fig. 1. Map of the study area in the Stavanger region of southwestern Norway.

from two weather stations (Figure 1) within the catchment area. Additionally, the dataset includes simulated hydrological data generated by the HBV (Hydrologiska Byråns Vattenbalansavdelning) model [12], a physical-based model that simulates various hydrological processes.

TABLE I
OVERVIEW OF THE FILTERED DATASET FROM NOVEMBER 4, 2018, TO SEPTEMBER 22, 2020

Variables	Count	Descriptions
Discharge	5	Flow rate data of water at specific locations and system values
HBV Sim. Data	9	Simulated hydrological data by HBV model, including precipitation, snow melt, and temperature
Logger Values	15	Data from sensors at multiple locations, e.g., water level, and temperature
Meteorological Data	7	Meteorological conditions data, e.g., air temperature, precipitation, humidity, and wind speed from various locations

By focusing on this filtered dataset (as summarized in Table I), we aim to gain a more accurate representation of the spatial-temporal interactions within the catchment area for our deep learning model.

III. DATA CLEANING

Before implementing the spatial-temporal attention-based LSTM model (explained in Section IV-D), the hydrological and meteorological dataset were cleaned through several pre-processing steps. These steps addressed missing values, outliers, and inconsistencies to ensure the suitability of the data for model training and evaluation.

To handle missing values, data imputation techniques were utlized, such as linear interpolation. This allowed to fill gaps in the time series data without causing significant distortions. The effectiveness of this approach was verified by comparing interpolated values with actual measurements whenever possible. The outliers within the dataset were identified through a detailed examination of the overall descriptive statistics. Similar to missing data, linear interpolation was used to replace these outliers, reducing their potential negative impact on the performance of the model.

Finally, data normalization and transformation procedures were conducted. This ensured that all variables were compatible and on a similar scale, which is a prerequisite for the input to the deep learning model. Specifically, min-max normalization was applied to transform all variables to a common scale, ranging from 0 to 1. This crucial step prevents potential biases or inaccuracies when working with data of different scales and units.

IV. METHODOLOGY

The methodology employed in this research centers on the development and implementation of a spatial-temporal attention-based LSTM model for inflow forecasting. The model incorporates both temporal and spatial attention mechanisms to capture complex interactions among different locations within the catchment area and temporal dependencies within time series data. The subsections below provide a detailed overview of the LSTM model and our proposed spatial and temporal attention mechanisms.

A. Long Short-Term Memory (LSTM)

Long Short-Term Memory (LSTM) is a type of Recurrent Neural Network (RNN) that is specifically designed to capture long-term dependencies in sequence data [3]. Unlike traditional RNNs, which suffer from vanishing and exploding gradient problems when dealing with long sequences, LSTMs incorporate gating mechanisms that enable the network to retain and update relevant information over long periods. These mechanisms include input, forget, and output gates, which together allow the LSTM to learn complex patterns and relationships in time series data [3].

B. Temporal attention

The Temporal Attention mechanism employed in this study is a variant of self-attention, which enables the model to selectively focus on relevant time steps within a sequence when making predictions by assigning different levels of importance, or attention weights, to each time step. These attention weights are computed from the hidden states of the LSTM itself, which captures the temporal dependencies of the sequence. The mechanism computes a context vector as a weighted sum of the LSTM hidden states, where the weights represent the attention scores. This allows the model to emphasize significant time steps, enhancing its ability to predict inflow based on historical data.

C. Spatial Attention

Similarly, the Spatial Attention mechanism implemented in this study is another form of self-attention. It assigns importance weights to each spatial feature in the input data, thereby allowing the model to capture complex spatial interactions among different locations within the catchment area. These attention scores are computed from the input features themselves, representing their relative importance for the current prediction task. The mechanism then multiplies the input features by their respective attention scores, enabling the model to emphasize relevant spatial features and thus enhance its forecasting performance.

D. Proposed Model: Spatial-Temporal Attention-Based LSTM

In this study, a Spatial-Temporal Attention-Based LSTM model is proposed, which combines the strengths of both spatial and temporal attention mechanisms within an LSTM architecture to effectively capture the spatial and temporal dependencies in hydrological and meteorological time series data. The proposed model comprises several components, including a batch normalization layer, spatial attention module, input linear layer, LSTM layer, temporal attention module, and output linear layer. The architecture of the proposed model is illustrated in Figure 2.

The input data is first pre-processed by applying batch normalization to improve the model's convergence rate and performance. This technique ensures that the features have a consistent scale and distribution, which can facilitate learning and reduce training time.

The spatial attention layer is then applied to the input data, allowing the model to selectively focus on different locations within the catchment area by assigning importance weights to each feature. By emphasizing the most relevant spatial features, the model is better equipped to capture the complex interactions between various locations and their respective hydrological properties.

The input data, after applying the spatial attention, is passed through a linear transformation layer. This layer adjusts the dimensions of the input data to match the LSTM's hidden state dimension. The transformed data is then fed into the LSTM layer, which models the temporal dependencies in the data by retaining and updating relevant information over long periods.

The temporal attention layer computes a context vector from the hidden states of the LSTM layer, capturing the most relevant temporal information for inflow forecasting. By assigning importance weights to different time steps in the sequence, the model can selectively focus on the most critical periods that contribute to accurate predictions. This context vector is used to update the LSTM hidden states and is then passed through the output layer, which generates the final inflow prediction.

V. Experiment Results and Analysis

This section presents the results of the experiments to evaluate the performance of the proposed spatial-temporal attention-based LSTM model in comparison with the baseline models. To showcase the model's ability to handle different forecasting horizons, two sets of experiments were performed. First, the model's performance is looked at in one-hour ahead inflow forecasting, which represents a simpler forecasting task. Next, the model is evaluated in a more challenging scenario by performing recursive multi-step forecasting with a horizon of 12 hours. A longer forecasting horizon allows to investigate the model's ability to capture and leverage spatial and temporal dependencies more effectively, as differences in performance between the models might become more visible at this scale.

A. Baseline models

To evaluate the performance of the proposed spatial-temporal attention-based LSTM model, it is compared against several baseline models. These models represent a range of approaches to inflow forecasting, from traditional statistical techniques to state-of-the-art machine learning methods. By comparing the model's performance with these baselines, the aim is to demonstrate the benefits of incorporating both spatial and temporal information in hydrological forecasting. The baseline models include:

- **HA:** Historical Average (HA) is a straightforward baseline model that uses historical data to calculate the average inflow for each time step. This calculated average value is then used as the forecasted value for future time steps.
- **ARIMA:** Autoregressive Integrated Moving Average (ARIMA) is a widely-used statistical model for time series forecasting [13]. ARIMA captures the temporal dependencies in the data by modeling the linear relationships between past and future observations.
- **FCN:** Fully Connected Networks (FCN) is a type of deep learning model that utilize multiple fully connected layers to learn and model the relationships between input features [11]. While FCN models can capture some temporal dependencies through their architecture, they are not specifically designed for time series data and may not account for spatial information, which may limit their performance in hydrological forecasting tasks.
- **LSTM:** The standard LSTM model, as previously introduced, is recognized for its capability to handle time series data, and is widely adopted in hydrological forecasting [6]. It provides a crucial comparison for our proposed model. However, it is worth noting that this version does not incorporate any attention mechanisms, meaning it may not fully capture the intricate spatial correlations that the proposed spatial-temporal attention-based LSTM model is designed to handle.
- **Temporal Attention LSTM:** This model represents an intermediate step between the generic LSTM model and the proposed spatial-temporal attention-based LSTM model. It incorporates a temporal attention mechanism within the LSTM architecture, allowing the model to selectively focus on different parts of the sequence [10]. However, it does not include the spatial attention component, and thus, does not account for the complex interactions between different spatial locations within the catchment and meteorological area.

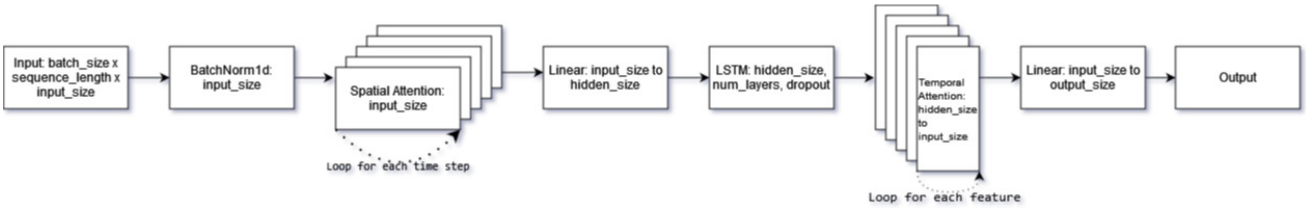

Fig. 2. Architecture of the proposed Spatial-Temporal Attention-Based LSTM model, illustrating the combination of spatial and temporal attention mechanisms within the LSTM framework for effective inflow forecasting.

B. Evaluation Metrics

In this paper, four common evaluation metrics are used to assess the performance of our proposed spatial-temporal attention-based LSTM model in inflow forecasting. These metrics were chosen for their ability to provide a comprehensive overview of the model's prediction accuracy, precision, and consistency.

- **Mean Absolute Error (MAE):** MAE calculates the average absolute difference between predicted and actual values. This metric provides an indication of the magnitude of errors made by the model irrespective of their direction. In the context of inflow forecasting, a lower MAE indicates more precise predictions. It is defined as:

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

where y_i is the actual value, $\hat{y}_i$ is the predicted value, and n is the number of data points.

- **Root Mean Squared Error (RMSE):** RMSE measures the square root of the average of squared differences between predicted and actual values. The squaring of differences in RMSE gives higher weight to large errors, making it particularly useful in identifying when the model's predictions significantly deviate from the actual values. It is defined as:

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

- **Mean Absolute Percentage Error (MAPE):** MAPE determines the average absolute percentage difference between predicted and actual values. This metric is especially useful for understanding the error rate in percentage terms, which can offer a more intuitive sense of the model's accuracy, especially when comparing across different datasets or models. It is defined as:

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \times 100$$

- **Coefficient of Determination (R^2):** R^2 evaluates the proportion of variance in the dependent variable that can be explained by the independent variables. A high R^2 indicates that the model can explain a large portion of the variability in the inflow data, signifying a strong predictive performance. However, it is also essential to be cautious with R^2 as it can be artificially high if the model is overfitted. In some cases, R^2 can even be negative if the chosen model performs worse than a simple horizontal line [14]. It is defined as:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y}_i)^2}$$

where $\bar{y}_i$ is the mean of the actual values.

Employing these evaluation metrics provide a comprehensive understanding of the model's ability to effectively forecast inflows [11], considering both spatial and temporal dependencies in the hydrological and meteorological data. These metrics makes it possible to assess the model's performance in different ways, ensuring a thorough evaluation of its inflow forecasting capabilities.

C. Training Configuration

To ensure a fair comparison between the proposed spatial-temporal attention-based LSTM model and the baseline models, the neural network models were implemented using the same experimental setup. The models were trained and tested on various combinations of hydrological, meteorological, and HBV simulated data, which were pre-processed and split into training, validation, and testing sets using a 70:20:10 ratio. The testing set consisted of 3,295 data points, representing a time span of approximately four and a half months during the year 2020. The models were trained using the Adam optimizer [15].

A Population-Based Training (PBT) method was used for hyperparameter tuning, which dynamically adjust and optimize hyperparameters during the training process [16]. By combining the advantages of random search and hand-tuning, PBT periodically adjusts hyperparameters of the well-performing models, promoting the exploration of diverse hyperparameter configurations and effectively accelerating the tuning process.

The hyperparameters were tuned using the Ray Tune library [17] to find the optimal combination for each model and each data combination. Training the models on different combinations of hydrological, meteorological, and HBV simulated data allowed for the exploration of the impact of different types of input data on the performance of the models. This approach also ensured that the hyperparameters of each model were tuned to optimize their performance for each specific data combination. The optimal hyperparameters for each model and data combination were then used for evaluation on the testing set.

The models were trained for a maximum of 100 epochs, with PBT making potential alterations to hyperparameters every 25 epochs. The number of epochs was chosen to allow sufficient time for the models to learn while mitigating the risk of overfitting. In this implementation, various hyperparameters including learning rate, weight decay, hidden size, and number of layers were tuned to optimize model performance.

Upon training 25 individual models for each combination of model type and variable set, the average performance of each set of models was computed according to the evaluation metrics. The model type that exhibited the best average performance was deemed the representative for each specific combination of model type and variable set. This can be seen in Table II and III, which is taking into account the unique characteristics of each input data combination and the variability introduced by different hyperparameters.

D. Hour Ahead Forecast

This subsection presents the hour-ahead inflow forecasting results for the proposed spatial-temporal attention-based LSTM model and the baseline models. Each model's performance is evaluated using the four evaluation metrics described earlier, and the results are summarized in Table II.

TABLE II

AVERAGE PERFORMANCE OF EACH MODEL TYPE ON THEIR BEST VARIABLE SET FOR A HOUR-AHEAD FORECAST

Model	Variable Set	MAE	RMSE	MAPE	R2
ARIMA	univariate	5.156	6.244	114.836	-0.043
HA	univariate	1.891	3.413	175.267	0.688
FCN	meteorological + hydrological + hbv	2.260	4.028	29.106	0.716
LSTM	meteorological + hydrological + hbv	3.015	4.322	49.290	0.650
LSTMTemporalAttention	**meteorological + hydrological**	**1.557**	**2.441**	**23.632**	**0.885**
LSTMSpatioTemporalAttention	meteorological + hydrological	1.994	2.960	28.880	0.782

From the results in Table II, the LSTM with Temporal Attention model (LSTMTemporalAttention), using both meteorological and hydrological variables, is the top performer. This model outperforms all others, achieving the lowest MAE, RMSE, and MAPE, and the highest R2 value.

The above suggests that the temporal attention mechanism plays a significant role in enhancing model performance for short-term forecasts like the one-hour ahead prediction. The spatial-temporal attention-based LSTM model also shows strong performance, emphasizing the benefits of incorporating both spatial and temporal information in inflow forecasting. However, it should be noted that the short forecasting horizon might reduce the relative importance of spatial correlations, as reflected by the performance of the LSTM model with only temporal attention.

The traditional ARIMA model, using a univariate input, shows the weakest performance, which could be due to the simplicity of the model. The Historical Average (HA) model, despite using a simple approach, achieves reasonable performance. The Fully Connected Network (FCN) model, leveraging meteorological, hydrological, and HBV variable sets, also provides a decent performance, but it still falls behind the LSTM-based models.

E. 12-Hour Ahead Forecast

This subsection presents the 12-hour-ahead recursive inflow forecasting results for the proposed LSTM model with spatial-temporal attention, along with other baseline models. The models developed for the hour-ahead forecast are utilized, and a recursive multi-step forecasting approach is employed to predict the inflow 12 hours into the future. The recursive multi-step forecasting approach is a method where the one-hour ahead forecast model is used to make multiple predictions into the future. For each forecast, the input sequence is updated with the model's latest prediction, while other features remain unchanged. This process continues recursively until the forecast horizon (12 hours, in this case) is reached.

This forecasting task presents additional challenges due to the extended forecasting horizon and the cumulative nature of prediction errors in the recursive approach, making the task more complex as the forecast horizon extends.

TABLE III

AVERAGE PERFORMANCE OF EACH MODEL TYPE ON THEIR BEST VARIABLE SET FOR A RECURSIVE 12-HOUR AHEAD FORECAST

Model	Variable Set	MAE	RMSE	MAPE	R2
ARIMA	univariate	5.471	6.820	176.511	-0.244
HA	univariate	4.540	7.162	258.459	-0.372
FCN	meteorological + hydrological	4.717	8.340	59.235	-0.133
LSTM	meteorological + hydrological	4.127	6.709	52.717	0.247
LSTMTemporalAttention	meteorological	3.402	6.187	40.662	0.321
LSTMSpatioTemporalAttention	**meteorological**	**3.333**	**5.913**	**39.501**	**0.370**

The results in Table III, shows that the proposed LSTM with Spatial Temporal Attention model surpasses all other models in the 12-hour-ahead forecasting task, recording the lowest MAE, RMSE, and MAPE, and the highest R2 score. This shows the models ability to effectively handle the increasing complexity of predictive challenges presented by longer forecasting horizons.

Interestingly, the optimal variable sets for each model in the 12-hour ahead forecast are not identical to those in the hour-ahead forecast. For both the LSTM with Temporal Attention and the proposed model, the meteorological variable set performs best, without the need for hydrological data. This can indicate that in the 12-hour recursive forecasting scenario, the meteorological information becomes more dominant for predicting inflow rates.

In comparison to the one-hour ahead forecast, the proposed model now outperforms the LSTM with Temporal Attention model. This indicates the increasing importance of spatial information as the forecasting horizon extends, highlighting the advantage of incorporating spatial dependencies, which the proposed model successfully implements.

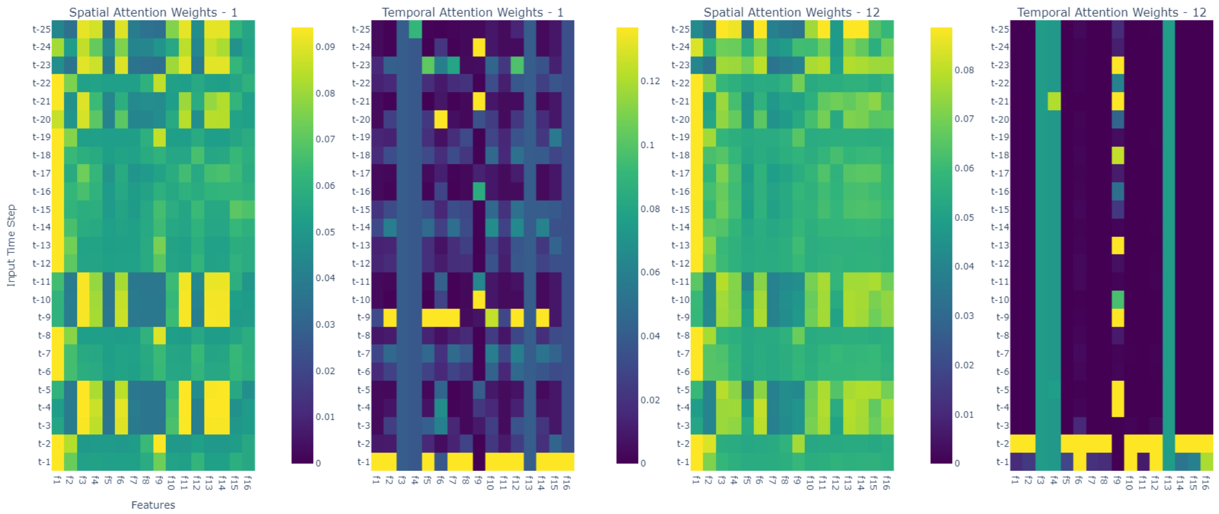

Fig. 3. Four heatmap visualizations of spatial and temporal attention weights for both one-hour-ahead (first and second from left) and recursive 12-hour-ahead (third and fourth) forecasts. Dark purple colors indicate low importance, while yellow colors denote high importance. This visual representation aids in understanding the model's focus while processing spatial-temporal data and how it varies between different forecast periods.

The performances of ARIMA and HA models deteriorate, possibly due to their inability to handle the complex temporal dependencies in the data over a more extended horizon. The FCN model, despite using both meteorological and hydrological variables, also falls short, possibly due to its limitations in modeling temporal dependencies as effectively as LSTM-based models.

F. Time-Space Consumption

In addition to forecasting accuracy, another important consideration is the computational efficiency of the model. This includes not just the time it takes for the model to be trained, but also the time it takes to make predictions (testing time), and the memory usage of the model (model size). These factors have been analyzed for the proposed spatial-temporal attention-based LSTM model and then compared to the baseline models. The results are summarized in Table IV.

TABLE IV
TIME AND SPACE CONSUMPTION OF THE MODELS

Model	Size (KB)	Training Time (s)	Testing Time (s)
HA	14.023	-	0.008
ARIMA	0.046	1.060	0.021
FCN	61.420	34.669	0.080
LSTM	525.320	54.300	0.210
LSTMTemporalAttention	734.693	64.570	0.251
LSTMSpatioTemporalAttention	424.229	106.095	0.229

Table IV demonstrates that the proposed spatial-temporal attention-based LSTM model has the longest training time among all models, excluding HA, which does not require training. This extended training time can be attributed to the additional complexity introduced by incorporating both spatial and temporal attention mechanisms. Despite this, the testing time of the proposed model is comparable to other LSTM-based models, indicating that the increased complexity does not significantly impact the prediction speed.

In terms of space consumption, although the proposed model requires more space than ARIMA and HA models, it consumes less space than LSTM and LSTM with Temporal Attention models. This smaller model size, despite its increased complexity, might be due to the more efficient representation learned by the spatial-temporal attention mechanism or due to the hyperparameter search process, which might affect the model size.

G. Interpretation of the Spatial-Temporal LSTM Model

Interpreting the decision-making process of this spatial-temporal attention-based LSTM model can provide valuable insights into the underlying hydrological processes. The focus is on the interpretation on the model's spatial and temporal attention mechanisms, which allow it to focus on different features and time steps during the forecasting process. By visualizing the spatial and temporal attention weights, one can understand which locations and time steps the model deems most important for making its predictions.

Figure 3 presents a series of four heatmaps, displayed in a horizontal sequence. They represent the spatial and temporal attention weights for the one-hour-ahead forecast and the recursive 12-hour-ahead forecast. The attention weights are color-coded, with darker purple colors signifying lower importance, and yellow colors indicating higher importance assigned by the model.

The first heatmap from the left shows the spatial attention for the one-hour-ahead forecast, followed by the temporal attention for the same. The third heatmap illustrates the spatial attention for the recursive 12-hour-ahead forecast, and finally, the last heatmap represents the temporal attention for the same forecast.

By comparing the heatmaps of the one-hour-ahead and 12-hour-ahead forecasts, a noticeable change in the attention weights can be observed. While the spatial attention heatmaps show relatively minor changes between the two forecast timeframes, the temporal attention heatmaps exhibit a more substantial shift. In the 12-hour-ahead temporal attention heatmap, more weights appear to have been assigned lower importance, as indicated by the darker purple colors.

VI. Conclusion

In this study, a spatial-temporal attention-based LSTM model for inflow forecasting in the Stavanger region of southwestern Norway has been developed and evaluated. This region, with its diverse topography and temperate oceanic climate, presented a complex environment for hydrological forecasting. The proposed spatial-temporal attention-based LSTM model incorporates spatial and temporal attention mechanisms, enabling it to capture complex spatial and temporal dependencies within the hydrological and meteorological data. The proposed model's performance was compared against several baseline models, including HA, ARIMA, FCN, a generic LSTM, and an LSTM with only temporal attention. The evaluation of the models was carried out using two forecasting tasks: one-hour ahead forecasting and recursive multi-step forecasting with a horizon of 12 hours.

The experimental results demonstrated the effectiveness of the proposed spatial-temporal attention-based LSTM model. In the one-hour ahead forecasting task, the temporal attention LSTM model performed better than the spatial-temporal attention-based LSTM model, suggesting that temporal dependencies might be more important for short-term inflow forecasts. However, the spatial-temporal attention-based LSTM model still outperformed most of the baseline models, indicating that the inclusion of both spatial and temporal information can contribute to improved accuracy in short-term inflow forecasts. In the more challenging 12-hour ahead forecasting task, the spatial-temporal attention-based LSTM model surpassed all baseline models, indicating that the model's ability to capture spatial dependencies might become increasingly important as the forecasting horizon extends.

VII. Future Research Directions

While the proposed spatial-temporal attention-based LSTM model has demonstrated promising results in short-term inflow forecasting, several directions for future research can be identified to further enhance the model's performance and applicability.

- **Evaluating performance across different catchment areas:** The study was conducted on a single catchment area, which limits the generalizability of the results. Future research could explore the model's performance across different catchment areas with varying hydrological and meteorological characteristics. This would help understand the model's robustness and adaptability to different environments.
- **Extending the forecasting horizon:** The proposed model's performance was tested with one-hour ahead and 12-hour ahead forecasts. It would be interesting to investigate how the model performs with even longer forecasting horizons, such as one-day or one-week ahead forecasts. This could provide further insights into the model's ability to capture long-term spatial-temporal dependencies.
- **Exploring other attention mechanisms:** While this model employs a specific type of self-attention mechanism for modeling spatial and temporal dependencies, there are other types of attention mechanisms that could potentially improve the model's performance. For instance, multi-head attention, a key component of Transformer models, could be explored in future studies to allow the model to focus on different aspects of the input simultaneously.

Pursuing the above research directions can further advance the field of inflow forecasting and improve water resource management and the response to hydrological events.

References

[1] S. D. Latif, A. N. Ahmed, E. Sathiamurthy, Y. F. Huang, and A. El-Shafie, "Evaluation of deep learning algorithm for inflow forecasting: A case study of Durian Tunggal Reservoir, Peninsular Malaysia," vol. 109, no. 1, pp. 351–369. [Online]. Available: https://doi.org/10.1007/s11069-021-04839-x

[2] M. A. Alawsi, S. L. Zubaidi, N. S. S. Al-Bdairi, N. Al-Ansari, and K. Hashim, "Drought Forecasting: A Review and Assessment of the Hybrid Techniques and Data Pre-Processing," vol. 9, no. 7, p. 115. [Online]. Available: https://www.mdpi.com/2306-5338/9/7/115

[3] H. Apaydin, H. Feizi, M. T. Sattari, M. S. Colak, S. Shamshirband, and K.-W. Chau, "Comparative Analysis of Recurrent Neural Network Architectures for Reservoir Inflow Forecasting," vol. 12, no. 5, p. 1500. [Online]. Available: https://www.mdpi.com/2073-4441/12/5/1500

[4] C. Bordin, H. I. Skjelbred, J. Kong, and Z. Yang, "Machine Learning for Hydropower Scheduling: State of the Art and Future Research Directions," vol. 176, pp. 1659–1668. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050920320925

[5] A. Mosavi, P. Ozturk, and K.-w. Chau, "Flood Prediction Using Machine Learning Models: Literature Review," vol. 10, no. 11, p. 1536. [Online]. Available: https://www.mdpi.com/2073-4441/10/11/1536

[6] X.-H. Le, H. V. Ho, G. Lee, and S. Jung, "Application of Long Short-Term Memory (LSTM) Neural Network for Flood Forecasting," vol. 11, no. 7, p. 1387. [Online]. Available: https://www.mdpi.com/2073-4441/11/7/1387

[7] Z. C. Herbert, Z. Asghar, and C. A. Oroza, "Long-term Reservoir Inflow Forecasts: Enhanced Water Supply and Inflow Volume Accuracy Using Deep Learning," vol. 601, p. 126676. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0022169421007241

[8] M. Børseth, "Literature review of data-driven methods used for forecasting hydrological events."

[9] S. Wang, J. Cao, and P. S. Yu, "Deep Learning for Spatio-Temporal Data Mining: A Survey," vol. 34, no. 8, pp. 3681–3700.

[10] J. Feng, L. Yan, and T. Hang, "Stream-Flow Forecasting Based on Dynamic Spatio-Temporal Attention," vol. 7, pp. 134 754–134 762.

[11] Y. Ding, Y. Zhu, J. Feng, P. Zhang, and Z. Cheng, "Interpretable spatio-temporal attention LSTM model for flood forecasting," vol. 403, pp. 348–359. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231220307530

[12] H. Sanner, J. Harlin, and M. Persson, *Application of the HBV Model to the Upper Indus River for Inflow Forecasting to the Tarbela Dam*. SMHI. [Online]. Available: http://urn.kb.se/resolve?urn=urn:nbn:se:smhi:diva-2432

[13] M. Valipour, M. E. Banihabib, and S. M. R. Behbahani, "Comparison of the ARMA, ARIMA, and the autoregressive artificial neural network models in forecasting the monthly inflow of Dez dam reservoir," vol. 476, pp. 433–441. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S002216941200981X

[14] 3.3. Metrics and scoring: Quantifying the quality of predictions. scikit-learn. [Online]. Available: https://scikit-learn/stable/modules/model$_{e}valuation.html$

[15] (09.05.23) Adam — PyTorch 2.0 documentation. [Online]. Available: https://pytorch.org/docs/stable/generated/torch.optim.Adam.html

[16] (18.05.23) A Guide to Population Based Training with Tune — Ray 2.4.0. [Online]. Available: https://docs.ray.io/en/latest/tune/examples/pbt$_{g}uide.html$

[17] (09.05.23) Ray Tune: Hyperparameter Tuning — Ray 2.4.0. [Online]. Available: https://docs.ray.io/en/latest/tune/index.html

Part III

Application

Chapter 8

Application: An Interactive Tool for Hydrological Forecasting

This thesis primarily focuses on the theoretical and practical applications of ML for hydrological forecasting, as detailed in the previous chapters. In addition to this theoretical exploration, a conceptual tool was developed to provide a hands-on, practical application of these theories.

This chapter provides an overview of this tool, an interactive application designed for individuals working at hydrological power plants, allowing them to experiment with hydrological forecasting using various ML models. The application serves as a bridge between theoretical ML methods and their practical application in hydrological forecasting, giving the end users the possibility to apply these techniques without the need to interact directly with the underlying code.

It should be noted that this application serves as a conceptual sketch, rather than a finalized product. As of now, no user-testing has been conducted.

8.1 Development Process

The development of the application involved a strategic approach towards code structuring to allow for code sharing between the application and experimental development in notebooks. This has been achieved through the following components:

- **Notebooks:** Interactive notebooks were used for the early stages of model development and experimental analysis. They are organized in a sequence from data cleaning, to model training and evaluation. The code in these notebooks enables easy transfer and integration into the application component.

- **App:** The app represents the frontend of the application, offering a user-friendly interface for interacting with the ML models and the data. Code from the notebooks were integrated into this part to provide the functionality of the application.
- **Data:** The *data* directory is categorized into *raw_data*, *clean_data*, and *transformation_data*, each self-explanatory of the data they contain. The structured segregation of data aids in maintaining the data pipeline and ensuring data integrity.
- **Source Python Code:** The primary Python project contains the shared code. It contains Python scripts that define core functionalities including data preprocessing, model definition, training procedures, and evaluation metrics. These scripts serve as the backbone of both the notebooks and the application, ensuring code consistency and minimizing redundancy.

Details related to obtaining the source code is available in Appendix A

8.2 Implementation and Technical Details

The implementation of the project was done using version control to manage the development process. Git is used to ensuring that the most recent and stable version of the code is available.

The development was also supported by various open-source Python libraries and frameworks, which greatly simplified the process and enhanced the application's functionality. Below is a brief overview of the main frameworks and libraries used in this project:

- **Streamlit:** Streamlit [4] was used for creating the web-based user interface of the application. It allows for the rapid prototyping of the application and interactive visualization of the model results, allowing for an intuitive and user-friendly application.
- **Poetry:** Poetry [1] was utilized for package management and environment setup. It managed dependencies effectively, ensuring interoperability of different libraries used in the project. Furthermore, it allowed for replication of the development environment across different setups.
- **PyTorch:** PyTorch [2] was chosen as the ML library for this project. It provides a wide range of functionalities for building and training neural networks.
- **Ray Tune:** Ray Tune [3] was used for hyperparameter tuning of the ML models. It supports a variety of search algorithms and scheduling mechanisms, making it an excellent tool for optimizing model performance and achieving the best results.

In addition to these primary tools, several other libraries were also utlized for data manipulation and analysis (numpy, pandas), data visualization (matplotlib, plotly), and ML (scikit-learn). The full list of dependencies is provided in the *pyproject.toml* file, included in the project repository. For the source code URL, please refer to Appendix A.

8.3 Overview of the Application

The conceptual application is an interface to demonstrate the use of ML in hydrological forecasting. While not the primary focus of this thesis, it serves as a practical demonstration of the theories and models explored in the previous chapters.

The application is divided into three main sections:

- **The App Page (C.1):** This page serves as the landing point for users. It provides essential information about the application and includes the functionality for data upload.
- **The Training Page (C.2):** Here, users can select their desired ML models and configure parameters for hyperparameter training. This component allows users to experiment with different forecasting models without directly interacting with complex code.
- **The Visualization Page (C.3):** This section presents the data and results of the forecasting models in an intuitive and interactive format. The interactivity of this page allows users to actively engage with the model's evaluation metric.

Each of these sections are designed with simplicity and ease-of-use in mind. Providing a straightforward method of exploring hydrological forecasting without the necessary understanding of coding. For a visual representation of the application, please refer to Appendix C, which includes a comprehensive screenshot series of the application's interface and functionalities.

8.4 Application Design and Workflow

The application is designed to offer a straightforward and efficient workflow. The process begins with data upload, moves through the model training stage, and concludes with the interactive visualization of results. Each step in this sequence is intuitive, ensuring users can easily navigate through the application. The workflow is outlined below and visually represented in Figure 8.1.

8.4.1 Data Upload and Pre-Processing

The first step within the application takes place on the app page, where users are tasked with uploading their datasets. The application is currently configured to accept data exclusively in CSV format.

It is crucial to note that the application assumes that the uploaded data has already been cleaned and prepared. This is to ensure the accuracy and reliability of the ML models. If the data is not properly prepared, the application will automatically eliminate rows containing missing values. This can lead to unintended gaps in the sequential data, potentially distorting the final analysis or predictions. As such, users are advised to pre-process their data, handling missing values, and conduct any necessary transformation before uploading.

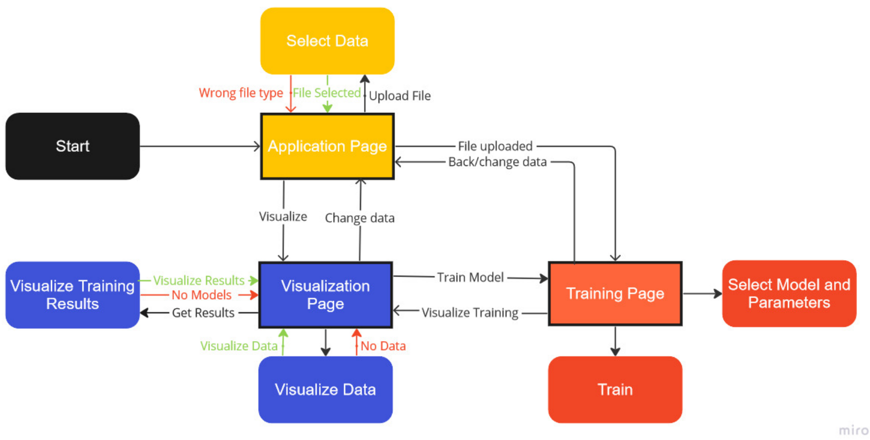

Figure 8.1: Flowchart illustrating the workflow of the interactive hydrological forecasting application, showcasing the process of data upload, model training, and result visualization.

8.4.2 Model Selection and Training

The training page of the application is where the users first select the datetime variable and the target variable. When these parameters have been set, the option for hyperparameter search parameters becomes available. Here the user can select models to use: FCN (Fully Connected Network), LSTM, LSTM with temporal attention, or LSTM with spatio-temporal attention. The user is also given the option for what variables sets it should include in the search, so if the user wants to compare the selected model on univariate data and meteorological data, they can do so. The training process is only shown as a loading circle to show progress. When training is completed, the trained models are displayed along with its results.

8.4.3 Data and Result Visualization

The visualization page is created to represent both the data and the model outputs in a user-friendly and interactive manner. It features dynamic plotly-generated graphs that allow users to zoom, pan, and hover over specific data points for more detailed information.

A comparison graph illustrating predicted versus actual values of the best performing model is shown, offering a clear visual overview of the model's forecasting capabilities.

Furthermore, an interactive table is provided, detailing the key performance metrics of all trained models. This table can be sorted by column values, offering a fast and convenient way to compare and evaluate model performance based on various metrics.

8.5 Further Work: Future Potential and Limitations

This conceptual application serves as a tool for understanding the potential of ML in hydrological forecasting. By allowing users, specifically hydrological power plant staff, to upload their own data, select models, and visualize results, the application offers a hands-on experience with the forecasting building process and the potential benefits of ML.

Despite being a conceptual tool, the application can present valuable utility in the field of hydrological forecasting. While it is not designed to replace comprehensive professional forecasting software, it can serve as an additional resource for end-users.

Moving forward, a possible further development could be automating this process. Currently, the application is mostly manual, requiring users to upload data, select models, select parameters, and initiate the training process. Future improvements of the application could incorporate more automation, such as automated data cleaning and preprocessing, model selection, hyperparameter tuning, and result interpretation. This would streamline the user experience, making the tool more convenient for a broader range of users.

It is also important to note that further work should involve a thorough user-testing process. Feedback from end-users, especially those operating in real-world hydrological power plants, would be invaluable in refining the tool's features and ensuring it meets their specific needs.

Finally, as this application remains a conceptual tool, the code underlying its features and functionalities is not fully optimized. Future work may involve refining and optimizing the code for greater efficiency and improved performance. This includes leveraging more advanced ML algorithms and techniques for superior forecasting accuracy.

Appendix

Appendix A

Source code

The following listings show code used in model training and evaluation. The full source code is available at: `https://github.com/BorMagnus/hydro-ml`.

A.1 Model Training

Listing 1: Setting up the environment and importing the necessary modules.

```python
import os
import sys

import pandas as pd

from ray import tune
from ray.tune.schedulers import PopulationBasedTraining

module_path = os.path.abspath(os.path.join(os.getcwd(), ".."))
if module_path not in sys.path:
    os.environ["PYTHONPATH"] = module_path
    sys.path.append(module_path)

from src.train import train_model
from src.experiment import get_variables_combinations
```

Listing 2: Definition of the main function including configuration for the model, hyperparameters, and PBT scheduler.

```python
def main(
    i,
    model,
    exp_name,
    file_name,
    n_samples,
    max_num_epochs,
    min_num_epochs,
```

```python
    local_dir="../ray_results",
):
    target_variable = "Flow_Kalltveit"
    datetime_variable = "Datetime"

    variables = [get_variables_combinations(file_name, datetime_variable)[i]]

    config = {
        "data_file": file_name,
        "datetime": datetime_variable,
        "data": {
            "target_variable": target_variable,
            "sequence_length": tune.choice([25]),
            "batch_size": tune.choice([256]),
            "variables": tune.grid_search(variables),
            "split_size": {
                "train_size": 0.7, "val_size": 0.2, "test_size": 0.1
            },
        },
        "model": tune.grid_search(model),
        "model_arch": {
            "input_size": tune.sample_from(
                lambda spec: len(spec.config.data["variables"]) + 1
            ),
            "hidden_size": tune.choice([32, 64]),
            "num_layers": tune.choice([1, 2, 3]),
            "output_size": 1,
        },
        "training": {
            "learning_rate": tune.loguniform(1e-5, 1e-1),
            "weight_decay": tune.loguniform(1e-5, 1e-1),
        },
        "num_epochs": max_num_epochs,
    }

    reporter = tune.JupyterNotebookReporter(
        metric_columns=[
            "train_loss", "val_loss", "test_loss", "training_iteration"
        ]
    )

    scheduler_population = PopulationBasedTraining(
        time_attr="training_iteration",
        perturbation_interval=min_num_epochs,
        hyperparam_mutations={
            "weight_decay": tune.uniform(0.0, 0.3),
            "learning_rate": tune.loguniform(1e-5, 1e-1),
            "model_arch.hidden_size": tune.choice([32, 64]),
            "model_arch.num_layers": tune.choice([1, 2, 3]),
        },
    )

```

```python
    stop = {
        "training_iteration": max_num_epochs,
    }

    if not os.path.exists(local_dir):
        os.makedirs(local_dir)

    results = tune.run(
        train_model,
        resources_per_trial={"cpu": 12, "gpu": 1},
        config=config,
        num_samples=n_samples,
        scheduler=scheduler_population,
        progress_reporter=reporter,
        name=exp_name,
        local_dir=local_dir,
        metric="val_loss",
        mode="min",
        stop=stop,
        keep_checkpoints_num=1,
        checkpoint_score_attr="val_loss",
    )

    return results
```

Listing 3: Execution of the main function for different models and dataset configurations.

```python

data_dir = "./data"
clean_data_dir = os.path.abspath(os.path.join(data_dir, "clean_data"))

results = []

model_dict = {
    "test-lstm": "LSTM",
    "test-temp": "LSTMTemporalAttention",
    "test-spa_temp": "LSTMSpatioTemporalAttention",
    "test-fcn": "FCN",
}
for i in range(4):
    for exp_name, model in model_dict.items():
        filename = "cleaned_data_4.csv"
        file_path = os.path.join(clean_data_dir, filename)

        num = filename.split("_")[2].split(".")[0]
        experiment = f"data_{num}-{exp_name}"

        analysis = main(
            i,
            [model],
            exp_name=experiment,
```

```
            file_name=filename,
            n_samples=25,
            max_num_epochs=100,
            min_num_epochs=25,
        )

        results.append(analysis)
```

Listing 4: Compilation of the results and printing of the combined dataframe.

```
dfs = []
for analysis in results:
    df = analysis.dataframe()[
        [
            "train_loss",
            "val_loss",
            "train_loss",
            "config/model",
            "time_total_s",
            "config/data/variables",
        ]
    ]
    dfs.append(df)

combined_df = pd.concat(dfs, ignore_index=True)
print(combined_df)
```

A.2 Model Evaluation

Listing 5: Python code for importing required libraries and modules for model evaluation

```
import json
from pathlib import Path
from operator import itemgetter
import pandas as pd
import numpy as np
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score

import sys
import os

import plotly.graph_objs as go
import plotly.subplots as sp
from plotly.offline import init_notebook_mode, plot, iplot
import plotly.express as px
from plotly.subplots import make_subplots

module_path = os.path.abspath(os.path.join("../"))
if module_path not in sys.path:
```

```
    sys.path.append(module_path)

from src.data import *
from src.train import create_model
from src.evaluate import *
```

Listing 6: This section includes code that plots the actual vs predicted values, calculates the model's metrics, averages them based on variable set, and prints the results.

```
plot_pred_actual(model_dirs, experiment)

model_dfs, parameters = calculate_model_metrics(model_dirs, experiment, best)

df_concat_avg_w_var = average_with_var(model_dfs, experiment)
print(df_concat_avg_w_var)
```

Listing 7: This code examines the best model's attention mechanism, applying the model to data to generate and plot attention weights.

```
for model_dir in model_dirs:
    if experiment not in str(model_dir):
        continue
    rows = []
    best_checkpoints = find_best_checkpoints(model_dir, num_best=1)
    for i, (checkpoint, val_loss, params) in enumerate(best_checkpoints):
        if params["model"] == "LSTM" or params["model"] == "FCN":
            continue
        elif params["model"] == "LSTMTemporalAttention":
            continue

        model = create_model(params)
        model = load_model_from_checkpoint(model, checkpoint)
        data_loader, _ = get_dataloader(params)
        test_dataloader = data_loader["test"]

        inputs, targets = next(iter(test_dataloader))

        output, spatial_attention_weights, temporal_attention_weights = model(
            inputs, True
        )
```

Listing 8: Running multi-step ahead predictions on the models, calculates the model's metrics, averages them based on variable set, and prints the results.

```
steps_ahead = 12
model_dfs = evaluate_multi_step_models(
                                  model_dirs, experiment, steps_ahead, best)
df_concat_avg_w_var = average_with_var(model_dfs, experiment)
print(df_concat_avg_w_var)
```

Appendix B

Data description

Variable	**Count**	**Mean**	**Std**	**Min**	**Max**
Wind_Speed_Nilsebu	62955	-11.36	373.07	-7999.00	22.10
Air_Temp_Nilsebu	63102	3.64	6.92	-20.70	27.20
Wind_Direction_Nilsebu	63079	213.03	113.67	-99.90	360.00
Relative_Humidity_Nilsebu	63079	64.44	43.18	-99.90	99.70
W_Level_Lyngsaana	50380	0.74	0.18	0.33	1.49
W_Temp_Hiafossen	50192	6.24	5.35	-0.20	19.90
W_Level_Hiafossen	50192	1.08	0.16	0.85	2.02
Air_Temp_Fister	63288	8.81	5.91	-11.80	32.40
Precipitation_Fister	63288	0.22	0.75	0.00	21.60
Flow_Lyngsvatn_Overflow	63286	0.04	0.74	0.00	22.40
Flow_Tapping	63286	0.94	1.44	0.00	49.05
W_Level_Kalltveit	63274	-26.23	517.62	-9999.00	2.28
Flow_Kalltveit	63288	6.51	7.57	0.00	110.26
W_Temp_Kalltveit_Kum	63269	-25.73	563.04	-9999.00	15.84
Precipitation_Nilsebu	63288	0.33	1.24	0.00	30.00
W_Temp_Hiavatn	45837	6.71	5.50	0.00	20.00
W_Level_Hiavatn	45836	1.94	0.22	1.70	3.85
W_Temp_Musdalsvatn	45978	5.82	5.44	0.00	19.20
W_Level_Musdalsvatn	45978	1.51	0.29	0.77	3.28
W_Temp_Musdalsvatn_DS	45955	5.85	5.20	-0.10	18.60
W_Level_Musdalsvatn_DS	45955	1.23	0.15	1.02	2.66
W_Temp_Viglesdalsvatn	45858	6.73	5.43	0.00	20.00
W_Level_Viglesdalsvatn	45858	1.79	0.19	1.52	3.22
Flow_HBV	63288	5.35	6.81	0.01	89.83
Precipitation_HBV	63288	0.28	0.82	0.00	26.32
Temp_HBV	63288	4.07	6.92	-20.20	27.72
Snow_Melt_HBV	29595	0.09	0.16	0.00	1.38
Snow_Water_Equivalent_HBV	29595	187.21	248.75	0.00	1006.83
Evaporation_HBV	29595	0.04	0.05	0.00	0.32
Soil_Water_Storage_HBV	29595	71.16	18.56	7.26	84.94
Groundwater_Storage_HBV	29595	15.95	11.31	0.62	102.19
Flow_Without_Kalltveit	63288	5.59	7.76	0.00	110.26
Mean_Flow_HBV	31241	5.18	6.60	0.01	137.05
Flow_Lyngsaana	63288	2.26	3.34	0.00	55.23
W_Temp_Lyngsaana	63283	6.40	5.34	-0.77	19.40
W_Temp_Kalltveit_River	50316	7.31	5.48	0.00	22.85

Table B.1: This table gives a full statistical description of all the variables in the raw dataset. For each variable, it provides the number of non-null values, the mean, the standard deviation, the minimum, and the maximum.

Variable	Count	Mean	Std	Min	Max
Wind_Speed_Nilsebu	16500	6.31	2.19	0.10	19.00
Air_Temp_Nilsebu	16500	4.02	6.52	-19.00	26.20
Wind_Direction_Nilsebu	16500	214.87	112.07	0.00	360.00
Relative_Humidity_Nilsebu	16500	42.86	44.83	0.00	99.70
W_Level_Lyngsaana	16500	0.74	0.17	0.37	1.32
W_Temp_Hiafossen	16500	6.29	5.33	-0.20	19.80
W_Level_Hiafossen	16500	1.06	0.15	0.89	1.99
Air_Temp_Fister	16500	9.26	5.65	-3.70	32.40
Precipitation_Fister	16500	0.23	0.85	0.00	21.60
Flow_Lyngsvatn_Overflow	16500	0.00	0.00	0.00	0.00
Flow_Tapping	16500	0.74	1.76	0.00	24.95
W_Level_Kalltveit	16500	0.62	0.15	0.46	2.09
Flow_Kalltveit	16500	6.16	6.43	1.12	90.86
W_Temp_Kalltveit_Kum	16500	6.07	1.65	2.76	10.80
Precipitation_Nilsebu	16500	0.30	1.03	0.00	30.00
W_Temp_Hiavatn	16500	6.40	5.42	0.00	19.90
W_Level_Hiavatn	16500	1.92	0.20	1.71	3.68
W_Temp_Musdalsvatn	16500	5.46	5.39	0.10	18.20
W_Level_Musdalsvatn	16500	1.45	0.29	0.77	3.28
W_Temp_Musdalsvatn_DS	16500	5.49	5.10	-0.10	17.40
W_Level_Musdalsvatn_DS	16500	1.20	0.14	1.02	2.66
W_Temp_Viglesdalsvatn	16500	6.41	5.35	0.30	20.00
W_Level_Viglesdalsvatn	16500	1.76	0.18	1.55	3.11
Flow_HBV	16500	5.62	6.36	0.06	89.83
Precipitation_HBV	16500	0.29	0.86	0.00	21.00
Temp_HBV	16500	4.36	6.54	-18.50	26.72
Snow_Melt_HBV	16500	0.11	0.18	0.00	1.38
Snow_Water_Equivalent_HBV	16500	222.59	292.03	0.00	1006.83
Evaporation_HBV	16500	0.04	0.06	0.00	0.32
Soil_Water_Storage_HBV	16500	72.43	15.15	12.42	84.94
Groundwater_Storage_HBV	16500	17.56	11.32	2.17	102.19
Flow_Without_Kalltveit	16500	5.43	6.54	0.00	90.86
Mean_Flow_HBV	16500	4.89	5.51	0.03	66.02
Flow_Lyngsaana	16500	2.15	2.92	0.00	34.29
W_Temp_Lyngsaana	16500	6.77	5.45	-0.10	19.40
W_Temp_Kalltveit_River	16500	7.21	5.34	0.00	22.58

Table B.2: This table gives a full statistical description of all the variables in the processed dataset. For each variable, it provides the number of non-null values, the mean, the standard deviation, the minimum, and the maximum.

Appendix C

Application

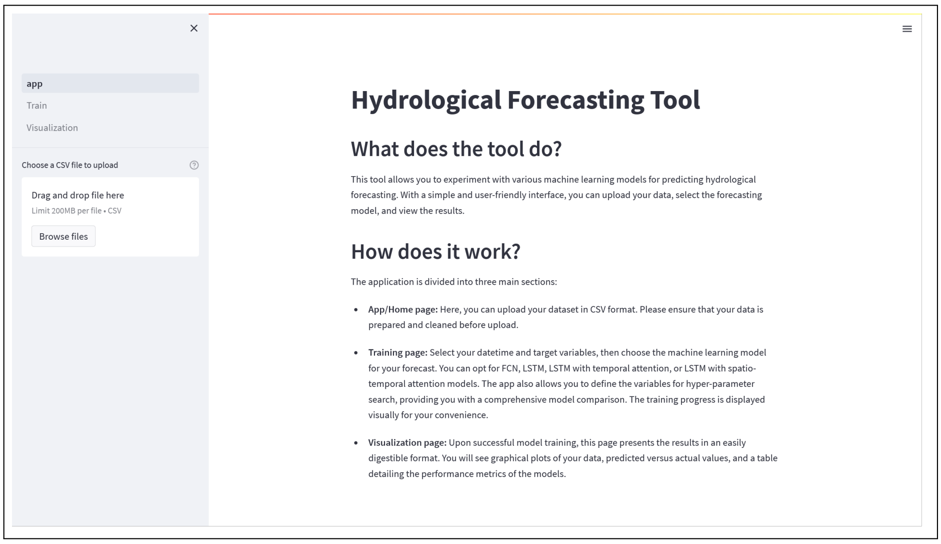

Figure C.1: The home page of the application, illustrating the user interface and file upload functionality in the left sidebar.

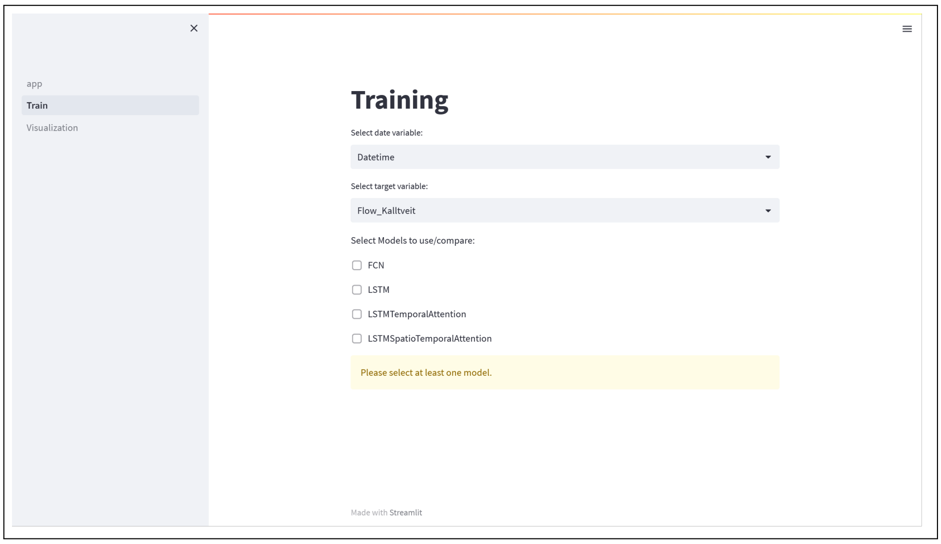

Figure C.2: The training page of the application, presenting the options for selecting the date variable, target variable, and ML models for training.

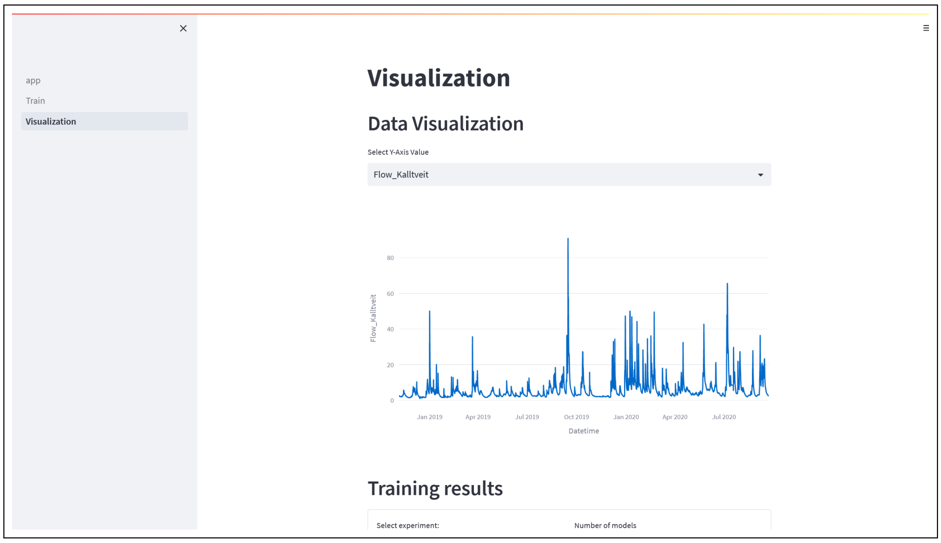

Figure C.3: The visualization page, offering interactive plots for viewing and analyzing the data and the results from the ML models.

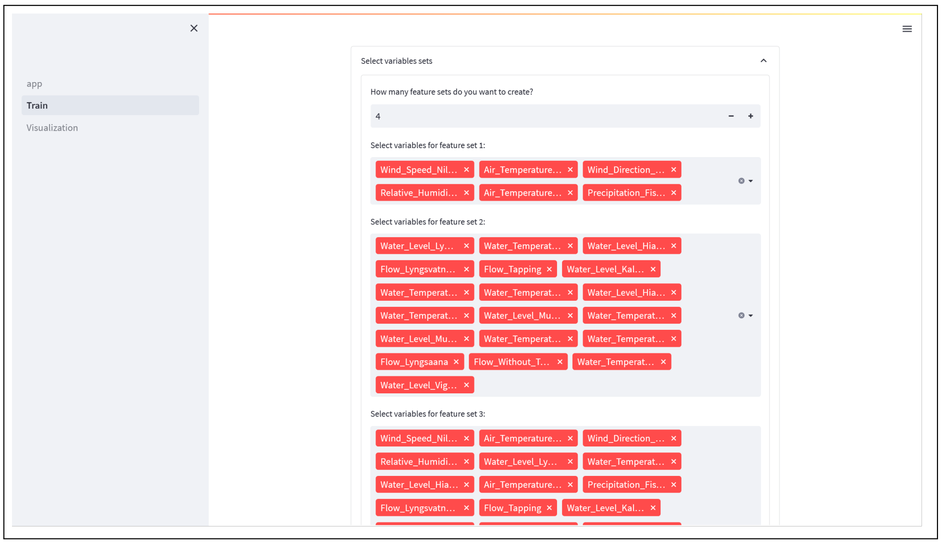

Figure C.4: Illustration of the variable set selection process within the application. If no specific set is selected, the application defaults to using all available variables.

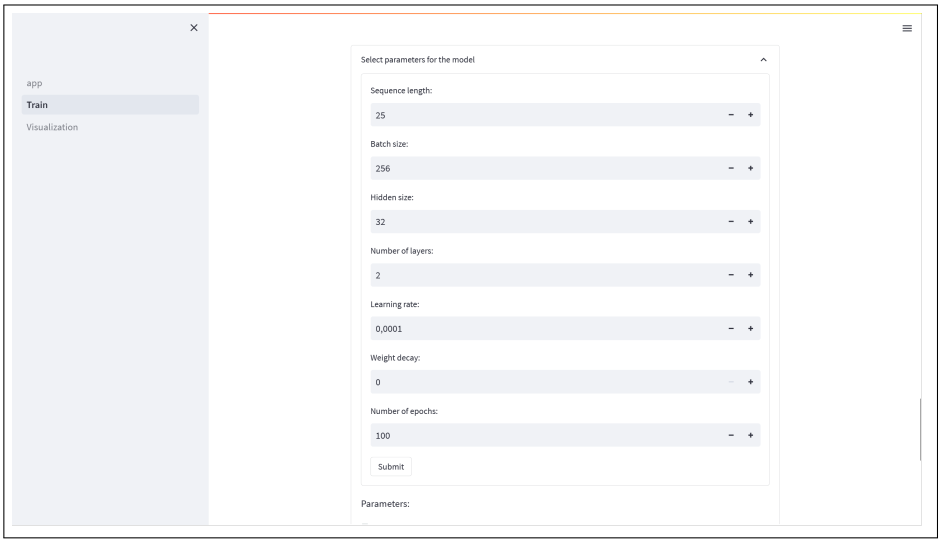

Figure C.5: Selection of hyperparameters for ML model training. In case of no specific selection, a random search is conducted for hyperparameters.

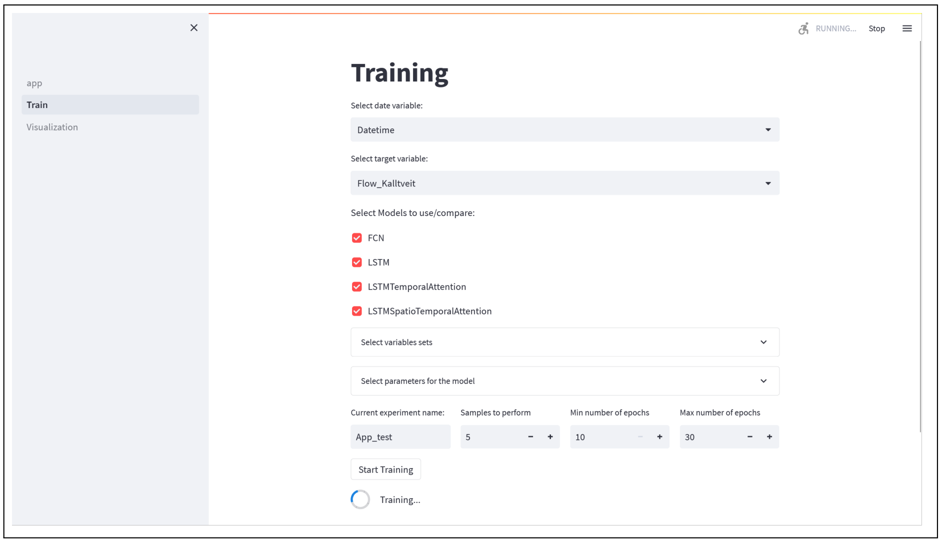

Figure C.6: Finalization of parameter selection, including naming the experiment, determining the sample size for training, and setting the range of epochs for model training. The image also illustrates the appearance of the application during the training process.

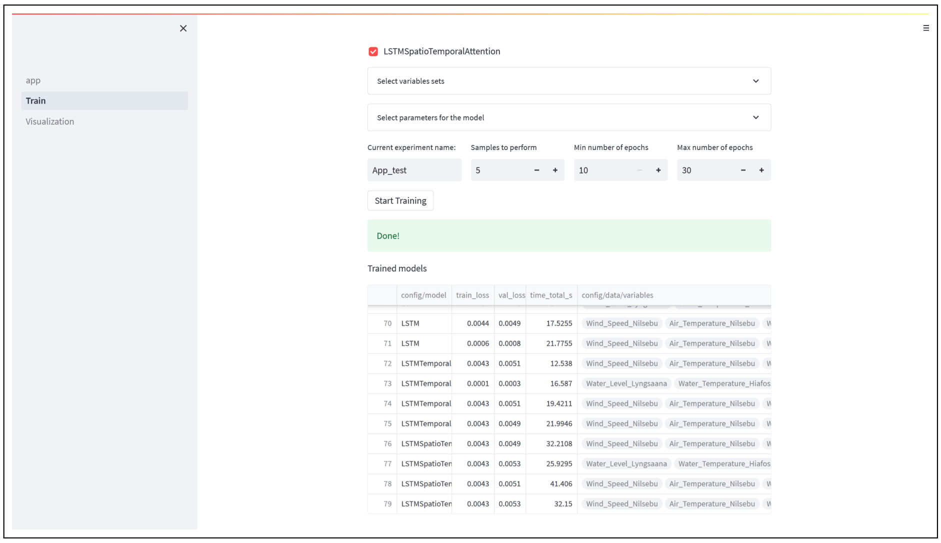

Figure C.7: The application's display after the completion of model training, featuring the results for each model in an interactive dataframe sortable by columns.

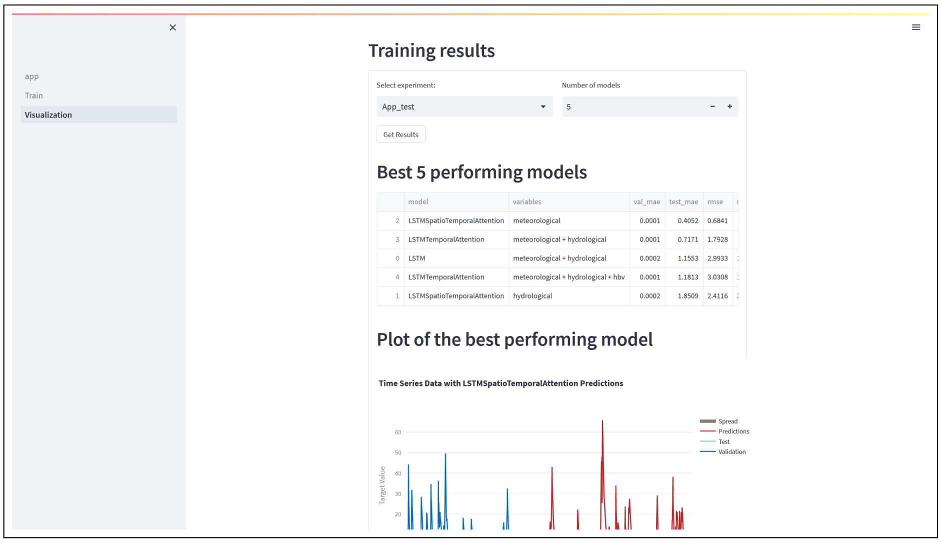

Figure C.8: The visualization page with displayed training results for a selected experiment, offering insights into the model's performance.

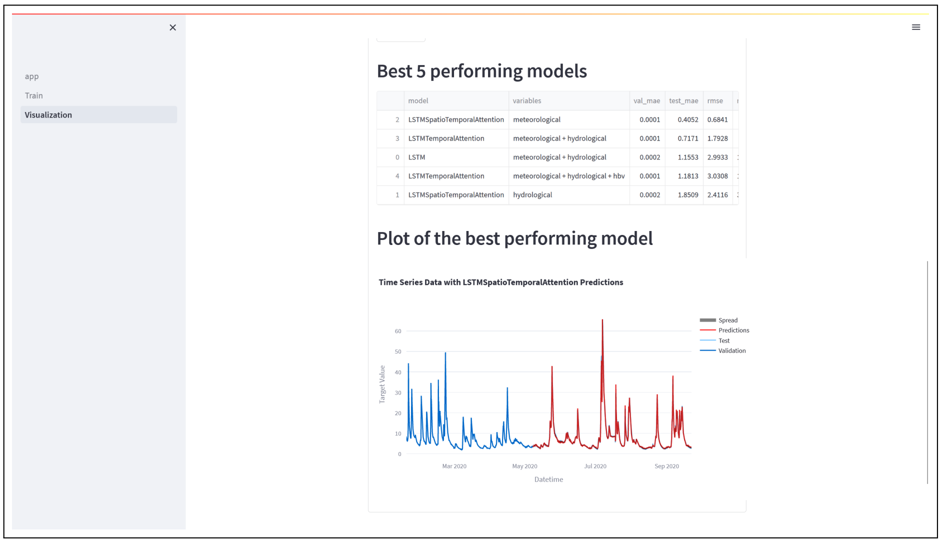

Figure C.9: Plot showcasing the performance of the best-performing model from the experiment, providing a interactive visual representation of the model's predictions.

Bibliography

[1] Poetry - Python dependency management and packaging made easy, https://python-poetry.org/

[2] PyTorch, https://www.pytorch.org

[3] Ray Tune - Fast and easy distributed hyperparameter tuning, https://www.ray.io/ray-tune

[4] Streamlit • A faster way to build and share data apps, https://streamlit.io/

[5] Ray Tune: Hyperparameter Tuning — Ray 2.4.0 (090523), https://docs.ray.io/en/latest/tune/index.html

[6] A Guide to Population Based Training with Tune — Ray 2.4.0 (180523), https://docs.ray.io/en/latest/tune/examples/pbt_guide.html

[7] Adamowski, J., Fung Chan, H., Prasher, S.O., Ozga-Zielinski, B., Sliusarieva, A.: Comparison of multiple linear and nonlinear regression, autoregressive integrated moving average, artificial neural network, and wavelet artificial neural network methods for urban water demand forecasting in Montreal, Canada **48**(1). https://doi.org/10.1029/2010WR009945, https://onlinelibrary.wiley.com/doi/abs/10.1029/2010WR009945

[8] Alawsi, M.A., Zubaidi, S.L., Al-Bdairi, N.S.S., Al-Ansari, N., Hashim, K.: Drought Forecasting: A Review and Assessment of the Hybrid Techniques and Data Pre-Processing **9**(7), 115. https://doi.org/10.3390/hydrology9070115, https://www.mdpi.com/2306-5338/9/7/115

[9] Apaydin, H., Feizi, H., Sattari, M.T., Colak, M.S., Shamshirband, S., Chau, K.W.: Comparative Analysis of Recurrent Neural Network Architectures for Reservoir Inflow Forecasting **12**(5), 1500. https://doi.org/10.3390/w12051500, https://www.mdpi.com/2073-4441/12/5/1500

[10] Bordin, C., Skjelbred, H.I., Kong, J., Yang, Z.: Machine Learning for Hydropower Scheduling: State of the Art and Future Research Directions **176**, 1659–1668. https://doi.org/10.1016/j.procs.2020.09.190, https://www.sciencedirect.com/science/article/pii/S1877050920320925

[11] Burkov, A.: The Hundred-Page Machine Learning Book. Andriy Burkov

[12] Ding, Y., Zhu, Y., Feng, J., Zhang, P., Cheng, Z.: Interpretable spatio-temporal attention LSTM model for flood forecasting **403**, 348–359. https://doi.org/10.1016/j.neucom.2020.04.110, https://www.sciencedirect.com/science/article/pii/S0925231220307530

[13] Feng, J., Wang, Z., Wu, Y., Xi, Y.: Spatial and Temporal Aware Graph Convolutional Network for Flood Forecasting. In: 2021 In-

ternational Joint Conference on Neural Networks (IJCNN). pp. 1–8. https://doi.org/10.1109/IJCNN52387.2021.9533694

[14] Fotovatikhah, F., Herrera, M., Shamshirband, S., Chau, K.w., Faizollahzadeh Ardabili, S., Piran, M.J.: Survey of computational intelligence as basis to big flood management: Challenges, research directions and future work **12**(1), 411–437. https://doi.org/10.1080/19942060.2018.1448896, https://doi.org/10.1080/19942060.2018.1448896

[15] Huang, I.H., Chang, M.J., Lin, G.F.: An optimal integration of multiple machine learning techniques to real-time reservoir inflow forecasting **36**(6), 1541–1561. https://doi.org/10.1007/s00477-021-02085-y, https://doi.org/10.1007/s00477-021-02085-y

[16] Ibrahim, K.S.M.H., Huang, Y.F., Ahmed, A.N., Koo, C.H., El-Shafie, A.: A review of the hybrid artificial intelligence and optimization modelling of hydrological streamflow forecasting **61**(1), 279–303. https://doi.org/10.1016/j.aej.2021.04.100, https://www.sciencedirect.com/science/article/pii/S111001682100346X

[17] Kasiviswanathan, K.S., Sudheer, K.P., Soundharajan, B.S., Adeloye, A.J.: Implications of uncertainty in inflow forecasting on reservoir operation for irrigation **19**(1), 99–111. https://doi.org/10.1007/s10333-020-00822-7, https://doi.org/10.1007/s10333-020-00822-7

[18] Latif, S.D., Ahmed, A.N., Sathiamurthy, E., Huang, Y.F., El-Shafie, A.: Evaluation of deep learning algorithm for inflow forecasting: A case study of Durian Tunggal Reservoir, Peninsular Malaysia **109**(1), 351–369. https://doi.org/10.1007/s11069-021-04839-x, https://doi.org/10.1007/s11069-021-04839-x

[19] Ley, C., Martin, R.K., Pareek, A., Groll, A., Seil, R., Tischer, T.: Machine learning and conventional statistics: Making sense of the differences **30**(3), 753–757. https://doi.org/10.1007/s00167-022-06896-6, https://doi.org/10.1007/s00167-022-06896-6

[20] Li, P.H., Kwon, H.H., Sun, L., Lall, U., Kao, J.J.: A modified support vector machine based prediction model on streamflow at the Shihmen Reservoir, Taiwan **30**(8), 1256–1268. https://doi.org/10.1002/joc.1954, http://onlinelibrary.wiley.com/doi/abs/10.1002/joc.1954

[21] Mosavi, A., Ozturk, P., Chau, K.w.: Flood Prediction Using Machine Learning Models: Literature Review **10**(11), 1536. https://doi.org/10.3390/w10111536, https://www.mdpi.com/2073-4441/10/11/1536

[22] Niu, W.j., Feng, Z.k.: Evaluating the performances of several artificial intelligence methods in forecasting daily streamflow time series for sustainable water resources management **64**, 102562. https://doi.org/10.1016/j.scs.2020.102562, https://linkinghub.elsevier.com/retrieve/pii/S2210670720307800

[23] Samir: Parameters Estimate of Autoregressive Moving Average and Autoregressive Integrated Moving Average Models and Compare Their Ability for Inflow Forecasting **8**(3), 330–338. https://doi.org/10.3844/jmssp.2012.330.338, http://thescipub.com/abstract/10.3844/jmssp.2012.330.338

[24] Sanner, H., Harlin, J., Persson, M.: Application of the HBV Model to the Upper Indus River for Inflow Forecasting to the Tarbela Dam. SMHI, http://urn.kb.se/resolve?urn=urn:nbn:se:smhi:diva-2432

[25] Seibert, J., Bergström, S.: A retrospective on hydrological catchment modelling based on half a century with the HBV model **26**(5), 1371–1388. https://doi.org/10.5194/hess-26-1371-2022, https://hess.copernicus.org/articles/26/1371/2022/
[26] Tekin, S.F., Karaahmetoglu, O., Ilhan, F., Balaban, I., Kozat, S.S.: Spatio-temporal Weather Forecasting and Attention Mechanism on Convolutional LSTMs, http://arxiv.org/abs/2102.00696
[27] Tian, D., He, X., Srivastava, P., Kalin, L.: A hybrid framework for forecasting monthly reservoir inflow based on machine learning techniques with dynamic climate forecasts, satellite-based data, and climate phenomenon information **36**(8), 2353–2375. https://doi.org/10.1007/s00477-021-02023-y, https://doi.org/10.1007/s00477-021-02023-y
[28] Valipour, M., Banihabib, M.E., Behbahani, S.M.R.: Comparison of the ARMA, ARIMA, and the autoregressive artificial neural network models in forecasting the monthly inflow of Dez dam reservoir **476**, 433–441. https://doi.org/10.1016/j.jhydrol.2012.11.017, https://linkinghub.elsevier.com/retrieve/pii/S002216941200981X
[29] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is All You Need. https://arxiv.org/pdf/1706.03762.pdf
[30] Wang, S., Cao, J., Yu, P.S.: Deep Learning for Spatio-Temporal Data Mining: A Survey **34**(8), 3681–3700. https://doi.org/10.1109/TKDE.2020.3025580