Improving Short-term Streamflow Forecasting with Wavelet Transforms: A Large-Sample Evaluation

by

John You

A thesis

presented to the University of Waterloo

in fulfillment of the

thesis requirement for the degree of

Master of Applied Science

in

Civil Engineering

Waterloo, Ontario, Canada, 2024

© John You 2024

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Accurate streamflow forecasting is instrumental to water management, including flood preparation and drought monitoring. The past two decades have seen a steady rise in the application of machine learning (ML) models to streamflow forecasting, given their ability to model highly nonlinear relationships, moderate data requirements, and accuracy. Successful application of ML to streamflow forecasting requires the modeler to select appropriate features (e.g., precipitation and air temperature) to use in an ML model. Since the original features can be insufficient, adding new features derived from existing ones, also known as feature engineering, is often needed to improve the accuracy of streamflow forecasts. Wavelet transforms (WTs) have become promising feature engineering methods for streamflow forecasting since they can decompose time series (e.g., precipitation) into multiple sub-series (coefficients). Each set of coefficients captures changes across different timescales (e.g., monthly, seasonal), allowing for the variance of the original time series to be associated with specific timescales. The different coefficients extracted by WTs are then used as features in an ML model, often improving the accuracy of streamflow forecasts compared to using the original features alone. Furthermore, different wavelet filters can capture different behaviours of a given time series (e.g., trends, short-term transients), making some more suitable for different applications (e.g., streamflow forecasting) than others. This leaves the modeler with the task of finding the right wavelet filter(s) for their application.

Despite many existing studies coupling WTs and ML for streamflow forecasting, none have explored a large hydro-climatically diverse sample of catchments to evaluate the impact of WTs on streamflow forecasting performance. Due to the small number of catchments included in existing streamflow forecasting studies using WT-based ML models, it is not clear how the performance of the adopted models generalizes to catchments with different characteristics. In addition, approximately 90% of studies using WTs for hydrological forecasting misuse WTs. The most common issue is not taking proper precautions to address look-ahead bias (i.e., the use of ‘future data’), invalidating the forecasts for real-world applications.

Thus, this thesis seeks to address the abovementioned gaps in the literature by undertaking a large-sample case study involving 620 catchments across the contiguous United States, using best practices for WT-based streamflow forecasting at the daily timescale. The WT-generated features are used in long short-term memory networks (LSTMs) to produce streamflow forecasts. LSTMs are selected due

to their exceptional streamflow forecasting performance compared to other commonly adopted models, as noted in the literature. In total, three LSTM configurations are considered: baseline LSTM (B-LSTM), wavelet LSTM (W-LSTM), and grid search LSTM (GS-LSTM). In the first configuration, a baseline LSTM model is developed for each catchment. In the second configuration, 33 different wavelet filters are used to engineer features based on several hydro-meteorological features (e.g., precipitation and air temperature), resulting in 33 different W-LSTM models for each catchment. For each catchment, the 33 different W-LSTM models are compared to the B-LSTM model to evaluate the impact of WTs on streamflow forecasting performance. In the third configuration, the B-LSTM models undergo hyper-parameter selection using grid search. This setup is used to test whether grid search has a greater impact on streamflow forecasting performance than WTs. All configurations are applied to one- and three-day-ahead streamflow forecasting.

For the one-day-ahead forecast horizon, W-LSTM improves upon B-LSTM performance in 97% of catchments and improves upon GS-LSTM in 50% of catchments. For a forecast horizon of three days ahead, W-LSTM improves upon B-LSTM performance in 97% of catchments and improves upon GS-LSTM in 41% of catchments. When considering only catchments where the B-LSTM model meets a minimum performance threshold (i.e., out-of-sample Nash-Sutcliffe Efficiency, OOS NSE, greater than 0.4), then for a forecast horizon of one day ahead, W-LSTM improves upon GS-LSTM in 60% of catchments, while for a forecast horizon of three days ahead, W-LSTM improves upon GS-LSTM in 70% of catchments.

Certain wavelet filters perform better than others. For instance, the W-LSTM using the Morris Minimum Bandwidth 4.2 filter outperforms B-LSTM in over 50% of catchments (where B-LSTM has an OOS NSE greater than 0.4) for both forecast horizons.

Overall, WTs provide the greatest improvement to forecasting performance (for both one and three day(s) ahead forecast horizons) in the D (snowy climates) and B (dry climates) Koppen classification regions. This finding presents a clear direction for researchers and practitioners when deciding whether WTs will benefit their streamflow forecasting models in their regions.

This thesis is the first to use a large sample of catchments to demonstrate that WTs are useful for improving ML-based streamflow forecasts. These models can be used for reservoir management, early flood warning systems, irrigation, navigation, and many other water management applications.

Future work can explore the combined optimization of wavelet filters and LSTM hyper-parameters to improve further upon the performance of the models reported in this thesis. Another worthwhile endeavor is to focus on modifications to the LSTM, such as quantile loss functions, Monte Carlo dropout connections, conformal prediction, and/or Bayesian methods to generate probabilistic forecasts enabling risk-based solutions to water management problems.

Acknowledgements

I would like to thank my supervisor, Dr. John Quilty, for providing me with invaluable feedback and support during my time as a graduate student at the University of Waterloo. I am also indebted to Sina and David for their support and am grateful to have worked alongside these friends. To the larger hydrology group here at UW, I had a blast getting to know everyone, and I cherish the many conversations that were had. Finally, I'd like to thank my family, without whom I would be incomplete.

Table of Contents

Author's Declaration ii

Abstract iii

Acknowledgements vi

List of Figures x

List of Tables xiv

Chapter 1 Introduction 1

1.1 Goal and Objectives 5

1.2 Thesis Organization 6

Chapter 2 Background 8

2.1 Brief Overview of Streamflow Forecasting and Typical Model Formulation 8

2.2 Comparison of Process-based and Data-Driven Hydrological Models 10

2.2.1 Process-based Hydrological Models 10

2.2.2 Data-Driven Hydrological Models 11

2.3 Feature Engineering 15

2.3.1 Wavelet Transforms 17

2.4 Feature Selection 24

2.4.1 Filter-based Feature Selection 24

2.5 Long Short-Term Memory Networks 27

2.6 A Gap in the Literature: Large-Sample Evaluation of WT-based Streamflow Forecasting Models 28

Chapter 3 Methods 33

3.1 Data 33

3.2 Maximal Overlap Discrete Wavelet Transform 34

3.2.1 Boundary Coefficients 37
3.3 Long Short-Term Memory Networks 38
3.4 Feature Selection 44
3.4.1 Entropy, Mutual Information, and Conditional Mutual Information 44
3.4.2 Feature Selection Algorithm 46
Chapter 4 Experimental Setup 48
4.1 LSTM Configurations 48
4.1.1 Baseline LSTM 49
4.1.2 Wavelet LSTM 50
4.1.3 Grid Search LSTM 54
4.2 Forecast Pipeline and Settings 54
4.2.1 Dataset Partitioning 55
4.2.2 Handling Boundary Coefficients 56
4.2.3 Feature Normalization 56
4.2.4 Feature Selection 57
4.2.5 Forecast Evaluation Metrics 57
4.2.6 LSTM Training Settings 58
4.2.7 LSTM Configurations 59
4.3 High-Performance Computing and Model Training Times 61
4.4 Filtering Catchments for Detailed Analyses 64
4.4.1 One Day Ahead Forecasting 64
4.4.2 Three Days Ahead Forecasting 65
Chapter 5 Results and Discussions 67
5.1 Impact of Different Wavelet Filters on Forecast Performance 67

5.2 Optimal Wavelet Filters 79
5.3 Comparing Optimal Wavelet Filters Against Hyper-Parameter Selection 85
5.4 Climates Where Wavelet Transforms Improve Performance 89
5.5 Summary of Experiments 96
5.6 Hydrographs for Some Notable Catchments 100
Chapter 6 Summary, Conclusions, and Recommendations 105
6.1 Summary 105
6.1.1 Research Problem and Goal 105
6.1.2 Experiments 106
6.1.3 Main Findings 106
6.2 Conclusions 107
6.3 Recommendations 108
References 109
Appendix A 143
Appendix B 146
Appendix C 155
Appendix D 158

List of Figures

Figure 1. Visualization of decomposing a time series into multiple sub-series using MODWT 20
Figure 2. Some wavelet filters from the Symlet family 36
Figure 3 Example time series with 10 observations. Periodic extension is applied for time indices beyond the observed time index domain (0-9). 37
Figure 4 Pseudocode for the adopted feature selection algorithm 47
Figure 5 Training (navy blue) and validation and testing (green) pipelines common to all three configurations. 55
Figure 6 Histogram of training times for B-LSTM (one and three day(s) ahead forecast horizons/lead time (LT)) 62
Figure 7 Histogram of training times for W-LSTM (one and three day(s) ahead forecast horizons/lead time (LT)) 62
Figure 8 Histogram of training times for GS-LSTM (one and three day(s) ahead forecast horizons/lead time (LT)) 63
Figure 9. Training times for feature selection and MODWT for each forecast horizon/lead time (LT) 63
Figure 10. Catchments where B-LSTM achieved a test set NSE > 0.4 for a forecast horizon of one day 65
Figure 11. Catchments where B-LSTM achieved a test set NSE > 0.4 for a forecast horizon of three days 66
Figure 12. Distribution of test set NSE across all 620 CAMELS catchments for B-LSTM (red coloured line) and the W-LSTM using different wavelet filters 69
Figure 13. Distribution of test set NSE across 90 and 60 catchments (for forecast horizons 1 and 3 days, respectively) where the B-LSTM (red coloured line) achieved a test set NSE > 0.4; the different coloured lines represent different wavelet filters used in the W-LSTM. 70
Figure 14. Median NSE across all 620 catchments for the various models considered in this thesis (see Section 4.1). 75
Figure 15. Similar to Figure 14, but only considering the filtered catchments (90 and 60 catchments, respectively, for one and three day(s) ahead forecast horizons) where the B-LSTM had an NSE > 0.4. 76

Figure 16. Fraction of catchments (out of 620) where the LSTM model specified in the x-axis outperformed the B-LSTM (in terms of test set NSE). Most W-LSTM models outperformed the B-LSTM for the majority of the 620 catchments. The red horizontal line indicates the 50 percent mark. 77
Figure 17. Similar to Figure 16, but now considering the fraction of filtered catchments (catchments where the B-LSTM achieved a test NSE > 0.4) where the LSTM model specified in the x-axis outperformed the B-LSTM (in terms of the test set NSE); 90 and 60 catchments were considered for forecast horizons one and three day(s) ahead, respectively. 78
Figure 18. Comparing the distribution of test set NSE values across all 620 catchments for the optimal W-LSTM and GS-LSTM models against the B-LSTM. 80
Figure 19. Similar to Figure 18, except the distribution of test set NSE values of the optimal W-LSTM and GS-LSTM models are compared against B-LSTM for the filtered catchments (catchments where the B-LSTM achieves a test NSE > 0.4). There were 90 and 60 filtered catchments for the one and three day(s) ahead forecast horizons, respectively. 81
Figure 20. Distribution of the GS-LSTM model's NSE for catchments where optimal GS-LSTM outperformed optimal W-LSTM. The red region indicates the area under the probability density function to the left of the 0.4 NSE threshold. 89
Figure 21. Geographic clustering appears for catchments where optimal W-LSTM showed the greatest absolute increase in NSE relative to B-LSTM (e.g., see Utah and Nevada). 90
Figure 22. Mean absolute increase in NSE of optimal W-LSTM over B-LSTM across catchments within a particular Koppen climate region. The two bar charts presented here are based on filtered catchments where B-LSTM achieved a test set NSE greater than 0.4. 91
Figure 23. Boxplots of absolute increase in NSE of optimal W-LSTM over B-LSTM across catchments within a particular Koppen climate region. The two charts presented here are based on filtered catchments where B-LSTM achieved a test set NSE greater than 0.4. 92
Figure 24. Correlation between catchment attributes and NSE improvement for the optimal W-LSTM compared to the B-LSTM for forecast horizon of one day ahead. 94
Figure 25. Top row: distribution of percent increase in NSE due to using optimal W-LSTM instead of B-LSTM across filtered catchments (i.e., where B-LSTM achieved an NSE greater than 0.4). Bottom row: Median percent improvement in NSE (across filtered catchments) when using different model configurations instead of B-LSTM. 97

Figure 26. Empirical cumulative distribution function (CDF) plots for one and three day(s) ahead forecast horizons considering all catchments. 98
Figure 27 - Empirical cumulative distribution function (CDF) plots for one- and three day(s) ahead forecast horizons considering filtered catchments (90 and 60 catchments for one and three day(s) ahead forecast horizons). 99
Figure 28. Comparison of optimal W-LSTM (top) to B-LSTM (bottom) for one day ahead streamflow forecasting of Noyo River near Fort Bragg, California. The blue line shows observed streamflow, while the orange lines show the forecasts. 101
Figure 29. Comparison of optimal W-LSTM (top) to B-LSTM (bottom) for one day ahead streamflow forecasting of Andrews Creek near Mazama, Washington. The blue line shows observed streamflow, while the orange lines show the forecasts. 102
Figure 30. Comparison of optimal W-LSTM (top) to B-LSTM (bottom) for one day ahead streamflow forecasting of South Willow Creek near Grantsville, Utah. The blue line shows observed streamflow, while the orange lines show the forecasts. 103
Figure 31. Comparison of optimal W-LSTM (top) to B-LSTM (bottom) for one day ahead streamflow forecasting (mm/day) of Big Sur River near Big Sur, California (United States). The blue line shows observed streamflow, while the orange line shows the forecast. 158
Figure 32. Comparison of optimal W-LSTM (top) to B-LSTM (bottom) for one day ahead streamflow forecasting (mm/day) of Cedar River near Cedar Falls, Washington (United States). The blue line shows observed streamflow, while the orange line shows the forecast. 159
Figure 33. Comparison of optimal W-LSTM (top) to B-LSTM (bottom) for one day ahead streamflow forecasting (mm/day) of General Creek near Meeks Bay, California (United States). The blue line shows observed streamflow, while the orange line shows the forecast. 160
Figure 34. Comparison of optimal W-LSTM (top) to B-LSTM (bottom) for one day ahead streamflow forecasting (mm/day) of Assabet River near Maynard, Massachusetts (United States). The blue line shows observed streamflow, while the orange line shows the forecast. 161
Figure 35. Comparison of optimal W-LSTM (top) to B-LSTM (bottom) for three days ahead streamflow forecasting (mm/day) of Steptoe Creek near Ely, Nevada (United States). The blue line shows observed streamflow, while the orange line shows the forecast. 162

Figure 36. Comparison of optimal W-LSTM (top) to B-LSTM (bottom) for three days ahead streamflow forecasting (mm/day) of Beaver River near Beaver, Utah (United States). The blue line shows observed streamflow, while the orange line shows the forecast. 163
Figure 37. Comparison of optimal W-LSTM (top) to B-LSTM (bottom) for three days ahead streamflow forecasting (mm/day) of Salmon River near Shoup, Idaho (United States). The blue line shows observed streamflow, while the orange line shows the forecast. 164
Figure 38. Comparison of optimal W-LSTM (top) to B-LSTM (bottom) for three days ahead streamflow forecasting (mm/day) of Merced River at Happy Isles Bridge in Yosemite National Park, California (United States). The blue line shows observed streamflow, while the orange line shows the forecast. 165
Figure 39. Comparison of optimal W-LSTM (top) to B-LSTM (bottom) for three days ahead streamflow forecasting (mm/day) of Encampment River, upstream of Hog Park Creek, near the town of Encampment in Carbon County, Wyoming (United States). The blue line shows observed streamflow, while the orange line shows the forecast. 166

List of Tables

Table 1. Candidate wavelet filters used in experiments 52
Table 2. Training times for each LSTM configuration 61
Table 3. Median percent improvement (PI) in test set NSE over B-LSTM and median NSE for all wavelet filters and forecast horizons; all 620 catchments considered. 71
Table 4. Median percent improvement (PI) in test set NSE over B-LSTM, as well as median NSE for all wavelet filters and forecast horizons; 90 and 60 catchments considered (for forecast horizons one and three day(s) ahead, respectively), where the B-LSTM achieved a test set NSE > 0.4. 73
Table 5. Median pairwise percent increase (PI) in NSE (over B-LSTM) and median NSE for optimal W-LSTM and optimal GS-LSTM for one and three days(s) ahead forecast horizons (LT1 and LT3, respectively) 83

Chapter 1 Introduction

Accurate streamflow forecasts are essential for effective water management and flood preparation (Chang & Guo, 2020). For example, hydrological forecasters produce streamflow forecasts to optimize hydropower generation, while government agencies, such as the British Columbia River Forecast Centre, use streamflow forecasts for flood warning and advisory notifications (Luo, 2021). Thus, streamflow forecast accuracy directly impacts the economy and the safeguarding of lives and property.

Streamflow forecasting models can be divided into three broad classes (Troin et al., 2021): statistics-based models, climatology-based models, and numerical weather prediction (NWP)- based models. These categories are not mutually exclusive, and considerable overlap can exist.

The first category of streamflow forecasting models (statistics-based models) typically relies on only historical streamflow measurements and encompasses stochastic models, such as Autoregressive Integrated Moving Average (ARIMA) and linear regression (Box & Jenkins, 1976). With time, these early statistical models evolved to account for nonlinear and nonstationary relationships between streamflow and explanatory variables (e.g., precipitation; see Anning & Parker, 2009), leading to the first use of machine learning (ML) models for streamflow forecasting (Karunanithi et al., 1994). Models in this category may use historical meteorological data up to the forecast "trigger" date when generating streamflow forecasts (Jahangir et al., 2023) but do not consider plausible future meteorological forecasts based on historical records or via NWP, such as the streamflow forecasting models in categories two (climatology-based models) and three (NWP-based models).

The second category of models (climatology-based models) includes historical meteorological observations. Streamflow forecasts are generated for a specific date by considering meteorological data such as the same day from a previous year as a plausible meteorological forecast. This allows for an ensemble of streamflow forecasts to be produced for any given day of the year (Day, 1985). This category is dominated by lumped-conceptual and physically-based distributed hydrological models (Troin et al., 2021). However, multivariate statistical models, such as Seasonal Autoregressive Integrated Moving Average with Exogenous Regressors (SARIMAX), and ML models (e.g., random forest [RF], support vector machine [SVM], extreme gradient boosting [XGBoost], long short-term memory networks [LSTM]) (Breiman, 2001; Chen & Guestrin, 2016; Cristianini & Ricci, 2008; Hochreiter & Schmidhuber, 1997) have also been used extensively.

The third category of streamflow forecasting models (NWP-based models) includes those that use NWP instead of historical meteorological data (as in the second category). These models are often preferable for long-range forecasts but depend on NWP systems, which are computationally expensive (Clark & Hay, 2004; Robertson et al., 2013; Shrestha et al., 2013).

Over the last 30 years, ML techniques have demonstrated improved performance over traditional process-based (physically-based and conceptual) hydrological models for both streamflow simulation and forecasting (Zounemat-Kermani et al., 2021), particularly the LSTM (Arsenault et al., 2023).

One of the defining characteristics that sets ML-based models apart from climatological and NWP-based streamflow forecasting methods is the flexibility in the features used in the ML model. Climatological and NWP-based methods require specific features to forecast streamflow. Climatological-based streamflow forecasting methods require historical and present observations of meteorological features such as precipitation, humidity, and air temperature. NWP-based methods require forecasts of these same features to produce streamflow forecasts. In contrast, ML-based streamflow forecasting models can use each of these previously mentioned features as well as a host of other features, such as satellite data on land use, vegetation, soil moisture, and evaporation (Alquraish & Khadr, 2021) to name a few. However, this thesis considers streamflow and meteorological data observed at a daily scale with increments of one day (i.e., daily time series).

Although there is flexibility in deciding on the features to use in an ML model, it is not known a priori which features (or combination of features) are the "best" or "optimal" for a given forecasting problem. Thus, great care is needed when deciding which features to use in an ML model. In general, determining which features to use in an ML model can be broken down into two tasks: feature engineering and feature selection (Kuhn & Johnson, 2019). The former is focused on improving the quality of features used in an ML model, while the latter is used to select only relevant features to use in an ML model, discarding redundant and irrelevant features (Guyon & Elisseeff, 2003).

Given the success of ML-based models for streamflow forecasting and the importance of carefully determining which features to use in ML models, this thesis couples ML with a promising feature engineering method and assesses its ability to increase streamflow forecasting accuracy via a large-sample study (a current gap in the literature). The adopted feature engineering method is based on the wavelet transform (WT), which has a reputation for being a useful feature engineering method

for improving the accuracy of ML-based streamflow forecasting models (Azarpira & Shahabi, 2021; Drisya et al., 2021; Kisi et al., 2021; Xie et al., 2022). However, despite their widespread success in the streamflow forecasting literature, WTs are often misused for forecasting applications, as described in Sections 2.3 and 3.2. For the remainder of this section, a brief introduction to WTs is provided, detailing how they have been (mis)used in the streamflow forecasting literature and the impact this has had on evaluating their ability to improve streamflow forecasting models. Afterwards, a series of goals and objectives to address this major gap in the literature are provided. The introduction concludes with an outline for the remainder of the thesis.

The WT is a mathematical tool that can be used to decompose a single time series into multiple time series by convolving the original time series with different convolution windows (Percival & Walden, 2000). The multiple time series generated by the WT reveal how frequencies present in the original time series change with time.

Two categories of WTs exist: continuous wavelet transforms (CWTs) and discrete wavelet transforms (DWTs). The latter is more commonly used for streamflow forecasting since the relevant data (e.g., streamflow, precipitation, air temperature) are typically measured at regularly spaced discrete time increments (e.g., every day). Numerous algorithms exist for computing CWTs and DWTs. This thesis uses the Maximal Overlap Discrete Wavelet Transform (MODWT), an algorithm that belongs to the DWT category. The MODWT is adopted since it has properties that make it suitable for forecasting problems (as described below and in Section 3.2). Given their suitability for time series data, only DWT and MODWT will be discussed hereafter.

MODWT takes a single time series as input and outputs multiple time series of the same length as the original time series. The multiple time series represent how the original time series varies through time for different time scales (see Section 3.2), which is useful for capturing seasonality, transients or abrupt changes in the original time series that may otherwise remain unknown if WTs were not used to reveal this information. This extracted time-frequency information can make it easier for the ML model to learn relationships between these new features and the target variable (e.g., future streamflow); see Sharma & Machiwal (2021). This extraction of time-frequency information from the original time series into multiple time series is an example of feature engineering.

When using WTs to generate new features from a given time series, two important parameters must be specified: the decomposition level and the wavelet filter type. The decomposition level

determines the number of new features that will be generated. In contrast, the wavelet filter type determines the shape of the convolution windows used to extract the new WT-generated features.

Unfortunately, most studies that employ WTs for streamflow forecasting and other hydrological applications fail to avoid look-ahead bias—a topic that was carefully studied in Quilty & Adamowski (2018). Look-ahead bias refers to the incorporation of future information into a forecasting model that results in a useless forecasting model since future information is not available in real-world forecasting scenarios, where only information up to the present can be known. Since look-ahead bias enables a forecasting model to access future information (known as data leakage in the ML literature), look-ahead bias results in unrealistically optimistic forecasting performance (Quilty & Adamowski, 2018). Quilty & Adamowski (2018) estimated that about 90% of papers that used WTs for hydrological forecasting up to the article's publishing date included models that suffered from look-ahead bias. The authors proposed a set of correct practices for real-world WT-based forecasting to address this issue. Based on these correct practices, they developed a forecasting framework named the Wavelet Data-Driven Forecasting Framework (WDDFF). The correct practices from this earlier study are adopted in the WT-based models developed herein.

Even for studies that have correctly applied WTs to streamflow forecasting (Fang et al., 2019; Jahangir et al., 2023; Ni et al., 2020), no study, to the best of the author's knowledge, has used a large sample of catchments to assess the performance gains due to coupling WTs with ML models. Previous WT-based streamflow forecasting studies have only used a limited number of catchments (e.g., between one and ten) when assessing improvements in model performance due to WTs (see Section 2.6 for representative studies from the literature).

Thus, the lack of a large-sample study for assessing forecasting performance gains to ML models due to the use of WTs for feature engineering is a noteworthy gap in the hydrological forecasting literature, especially given their growing popularity for streamflow forecasting (Kambalimath S & Deka, 2021). This is because improvements in forecasting performance due to WTs on a small number of catchments may not be generalizable to other catchments (e.g., see Papacharalampous, 2019). In other words, when the number of catchments used to assess improvements in model performance due to WTs is small (as in the literature) there is a higher chance to claim that WTs improve streamflow forecasting performance when they *do not*. Hence, increasing the sample size increases the ability to detect a true effect (i.e., correctly concluding WTs improve streamflow forecasting performance).

A large sample evaluation of WT-based streamflow forecasting models can also help answer the following questions:

1. Does the distribution of streamflow forecasting performance across a large sample of catchments meaningfully improve when WTs are incorporated?
2. Under what hydroclimatic conditions do WTs tend to improve streamflow forecasting performance?

To pursue answers to these questions, the CAMELS (Catchment Attributes and MEteorology for Large-sample Studies; Newman et al., 2015) dataset is adopted. This dataset includes hydrometeorological time series and catchment attributes for 671 catchments across the continental United States. Thus, this thesis attempts to build upon the movement of "large-sample hydrology" that has been influential in recent years (Addor et al., 2020; Gupta et al., 2014; Klingler et al., 2021; Kratzert et al., 2023; Newman et al., 2015; Papacharalampous et al., 2022).

The goal and objectives of this research are given in the next sub-section before an outline of this thesis is provided.

1.1 Goal and Objectives

Given that the literature is missing a large-sample study that assesses whether WTs improve the performance of ML-based streamflow forecasting models, this thesis aims to address this gap. To this end, a case study including a large sample of catchments covering diverse hydroclimatic conditions is used to evaluate improvements in streamflow forecasting performance due to the use of WT-generated features in a baseline model. The case study uses the CAMELS dataset and adopts LSTM as the baseline model due to its popularity and strong performance for streamflow simulation and forecasting (Arsenault et al., 2023; Li & Yuan, 2023). Since the literature lacks a large-sample study that quantifies any improvements in streamflow forecasting performance due to WTs, it is difficult for researchers and practitioners to decide whether the added effort of including WT-generated features will result in strong enough improvements in forecast accuracy to warrant their use. The difficulty arises because it is unclear whether WT-based streamflow forecasting models developed for catchments in a given geographical region will provide similar improvements in accuracy when developed for another geographical region. This research aims to contribute some initial findings on this important issue.

Moreover, most studies applying WTs for hydrological forecasting misapply WTs, leading to look-ahead bias. This thesis adheres to best practices in developing WT-based forecasting models (see Quilty & Adamowski, 2018), eliminating look-ahead bias. Hence, the results herein present a realistic assessment of the impact of WTs on streamflow forecasting performance across many catchments (Arsenault et al., 2023; Li & Yuan, 2023). To reach the intended goal of this thesis, four objectives are considered:

1. Assess how different wavelet filters impact streamflow forecast performance across the CAMELS dataset, using LSTM as the baseline model.
2. Explore the impact of selecting the optimal wavelet filter through exhaustive search on forecast performance.
3. Contrast the improvements from selecting an "optimal" wavelet filter with those from fine-tuning the LSTM model's architecture via a hyper-parameter selection (in particular, grid search).
4. Investigate any regional climate-based groupings of catchments where WTs substantially improve forecast performance compared to the baseline LSTM.

The outcomes of this thesis are intended to help clarify the utility of WT-generated features in streamflow forecasting models, with a focus on evaluating improvements to forecasting performance. While this research adopts LSTM as the baseline model, it is anticipated that the results will still be of general interest to those familiar with different ML models. As this thesis will demonstrate, catchments with certain climatological features have more to gain from coupling WTs with LSTM than others.

1.2 Thesis Organization

The remainder of the thesis proceeds as follows. Chapter 2 (Background) provides essential background information required to understand the experiments and results of this thesis. It includes a brief overview of streamflow forecasting, describes the main types of hydrological models, and provides key background on ML, feature engineering, and WTs.

Chapter 3 (Methods) describes the CAMELS dataset used in the experiments and provides necessary details about WTs, LSTM, and feature selection—the main methods used in the experiments.

Chapter 4 (Experimental Setup) details the configuration of the experiments performed in this thesis. It includes details on the LSTM configurations, the streamflow forecasting "pipeline," model training, and how the experiment results were filtered for further analysis in Chapter 5. Chapter 5 (Results and Discussion) presents key results from the experiments outlined in Chapter 4.

Chapter 6 (Summary, Conclusions, and Recommendations) summarizes the thesis, provides its main conclusions, and provides recommendations anticipated to stimulate future research in WT-based hydrological forecasting.

Chapter 2 Background

This section compares new developments from this thesis with the existing literature to identify this thesis's contribution to the streamflow forecasting domain. This chapter begins by briefly discussing the importance of streamflow forecasting, including general details on the formulation of streamflow forecasting models. Afterwards, process-based hydrological models are contrasted against data-driven models. Next, a brief discussion of ML in hydrology is given, followed by a discussion on feature engineering. The discussion progresses to address WTs, their role in feature engineering, and the architecture and development of LSTM networks. The chapter concludes by identifying a gap in the literature, particularly concerning the extensive evaluation of WT-based streamflow forecasting models.

2.1 Brief Overview of Streamflow Forecasting and Typical Model Formulation

Streamflow forecasting involves predicting the flow rate of a river or stream for a specific future period based on current and historical data. It is a critical component of hydrological applications and has many uses, some highlighted below. While the streamflow forecasts developed in this thesis are made at the catchment outlet, these forecasts can be converted to stages and then to inundation extents for flood inundation forecasting (Aristizabal et al., 2023).

Flood Prediction and Mitigation: Streamflow forecasts can provide early warnings for potential flooding events (Keating et al., 2021). This allows for the evacuation of people, the protection of property, and the implementation of strategies to mitigate flood damage (Kreibich et al., 2017).

Water Resource Management: Accurate streamflow forecasts enable the management of water resources (Gibbs et al., 2018). They help in planning for water supply and demand, managing irrigation needs, and operating hydroelectric power plants (Booras et al., 2018; Cassagnole et al., 2021; Castelletti et al., 2020; Edossa & Babel, 2011; Ramos et al., 2021).

Climate Change Studies: Understanding and forecasting streamflow contributes to the study of climate change (Oo et al., 2020; Wang et al., 2020). Changes in streamflow patterns can indicate broader climatic shifts, such as increased evaporation or changes in precipitation (Sharma & Wasko, 2019; Somorowska, 2017; Vicente-Serrano et al., 2019).

In this thesis, the focus is on daily data and short-range forecasts (Troin et al., 2021), where the forecast horizons are one and three day(s) ahead. Thus, the forecasting models developed herein are intended for applications such as flood prediction and mitigation and reservoir operation.

Streamflow forecasting employs various techniques ranging from statistical methods, such as time series analysis (Adnan et al., 2017; Tyralis et al., 2021) and ML (Kabir et al., 2020; Adnan et al., 2020; Sharma et al., 2022), to more complex distributed physically-based models (Madsen, 2017) that aim to simulate the entire hydrological cycle (e.g., HydroGeoSphere; see Davison et al., 2018). The choice of method depends on various factors, including the available data, the forecast's timeframe, and the specific application.

ML is now gaining acceptance in operational settings, traditionally dominated by physically-based models (Teng et al., 2017), moving beyond streamflow forecasting models to spatially distributed flood inundation models (Nevo et al., 2022). While this thesis uses WTs and ML to forecast streamflow at the catchment outlet, the same tools can be used for spatially distributed flood forecasting.

In applying ML to streamflow forecasting, meteorological features averaged across a catchment's area are commonly used to forecast streamflow. These features include daily minimum and maximum air temperature, average humidity, precipitation, soil moisture content, etc. In this thesis, the data for each catchment are in tabular format (rows and columns represent measurements and features, respectively).

The structure of the data used in this thesis is as follows. The first column includes the dates (measured on a daily scale) associated with the target variable Y, which is in the second column. The target variable represents daily streamflow at a given forecast horizon (i.e., lagged forward in time from the date vector by the given forecast horizon; see below). The remaining columns of the table include the features matrix $\boldsymbol{X}$, containing the meteorological features. The rows of the table are arranged such that the first row corresponds to the oldest observation, and the last row corresponds to the newest observation.

Consider an example where the entry in the date column for a given row is 04/22/2010, and the forecast horizon is three days ahead. In this case, the target is associated with streamflow on 04/25/2010, while the meteorological features are measured for 04/22/2010. This thesis uses data in this format to formulate the forecasting problem as a regression problem.

2.2 Comparison of Process-based and Data-Driven Hydrological Models

Hydrological models are pivotal in understanding the hydrological cycle, simulating future scenarios, and supporting water management decisions. These models can be broadly classified into process-based (physically-based and conceptual) and data-driven hydrological models. Data-driven models include classical regression (e.g., multiple linear regression), time series, and ML models. While both classes of hydrological models may serve different needs, both are often used to simulate and forecast streamflow. Further details on process-based and data-driven models are given in the following sub-sections.

2.2.1 Process-based Hydrological Models

Process-based models simulate the water cycle based on the underlying physical principles governing the flow of water in the environment (Li et al., 2023). Process-based models are taken here to include both distributed physically-based models and lumped conceptual models.

Physically-based models often include equations derived from the laws of physics to describe processes such as precipitation, evaporation, infiltration, and runoff. Examples of such models include the Soil Water Assessment Tool (SWAT) (USDA ARS Grassland Soil and Water Research Laboratory & Texas A&M AgriLife Research, 2018), the Precipitation-Runoff Modeling System (PRMS) (Regan et al., 2018), and the Hydrologic Engineering Center Hydrologic Modeling System (HEC-HMS) (Scharffenberg & Harris, 2012).

Physically-based models are based on a theoretical understanding of hydrological processes. This underlying knowledge enables them to provide insights into the mechanics of a hydrological system and often allows them to make informed predictions under various conditions or regimes. However, these models require specific input data, such as soil properties, land use attributes, and topography. Additionally, configuring and calibrating these models is intricate and time-consuming. Moreover, the models depend on a wide range of parameters. These parameters are often difficult to assign directly, introducing uncertainty into the model's predictions (Brigode et al., 2013; Fan, 2020; Herrera et al., 2022).

Lumped conceptual models are widely used in hydrology for streamflow forecasting and are characterized by their simplified representation of catchment processes (Sezen et al., 2019). Some different models include Génie Rural à 4 paramètres Journalier (GR4J; (Perrin et al., 2003),

Hydrologiska Byråns Vattenbalansavdelning (HBV; (Seibert & Bergström, 2022), and Hydrological Model of École de technologie supérieure (HMETS; (Martel et al., 2017). These models are known for their parametric parsimony and are widely used due to their ability to yield good model performance after calibration using historical watershed input-output data (Randrianasolo et al., 2010). Lumped conceptual models treat a catchment as a single homogeneous unit, making them relatively easy to implement and computationally efficient. They are particularly useful in ungauged or data-limited regions, where they can provide valuable insights into hydrological processes (Chiew, 2010).

However, lumped conceptual hydrological models, while efficient, face several limitations. They struggle to accurately simulate (and, by extension, forecast) different climate conditions, particularly during dry periods (Liu et al., 2023). The usefulness of these models depends on the accuracy of their calibration, which can be challenging due to the difficulty in directly measuring many of the model parameters (Lan Anh et al., 2010). Furthermore, these models often oversimplify the complex evapotranspiration process, leading to inaccuracies in predictions (Liu et al., 2023).

Compared to physically-based models, lumped conceptual models are computationally efficient and require fewer input data, making them suitable for data-limited regions. Physically-based models, on the other hand, aim to represent the physical processes of the hydrological cycle in detail, considering spatial variability and physical characteristics of the catchment – a characteristic lacking in lumped conceptual models. While physically-based models may provide a more detailed representation of hydrological processes, they often require extensive data and computational resources (Yang et al., 2020) beyond the requirements of lumped conceptual models, which can be challenging in ungauged or data-limited regions.

2.2.2 Data-Driven Hydrological Models

Data-driven hydrological models, often called “black box” models, use statistical relationships between input and output data to generate streamflow forecasts. These models contrast process-based hydrological models in that they lack a physical understanding of hydrological systems (hence the term “black box”) and model hydrological variables (such as streamflow) as random variables (Sharma & Machiwal, 2021).

Streamflow forecasting using data-driven models began with classical regression and time series models, such as multiple linear regression (Carlson et al., 1970) and ARIMA (Box & Jenkins, 1976), respectively. Numerous studies involving data-driven models have since been applied

successfully for streamflow forecasting. Many data-driven models used for streamflow forecasting are time series models derived from ARIMA. Some examples include seasonal autoregressive integrated moving average (SARIMA) (Alonso Brito et al., 2021), fractionally differenced autoregressive integrated moving average (FARIMA) (Mehdizadeh et al., 2019), autoregressive fractionally integrated moving average (ARFIMA) (Tyralis et al., 2021), ARIMA and SARIMA with exogenous regressors (ARIMAX and SARIMAX, respectively) (Abrahart & See, 2000; Castellano-Méndez et al., 2004; Fernando De Toledo et al., 2023; Papacharalampous & Tyralis, 2020; Valipour, 2015; Valipour et al., 2013).

Since the late 1960s, more complex and nonlinear data-driven models have been applied to streamflow forecasting (Amiri, 2015). The growing interest in ML in hydrology has correspondingly led to an increasing number of streamflow forecasting models based on state-of-the-art ML methods derived from other domains (Ng et al., 2023; Papacharalampous & Tyralis, 2022).

The introduction of artificial neural networks (ANNs) (Haykin, 1999) to streamflow forecasting (Karunanithi et al., 1994) led to its widespread application in hydrology due to its ability to learn highly nonlinear relationships between streamflow and meteorological features (e.g., rainfall), alongside its high forecasting accuracy (ASCE Task Committee on Application of Artificial Neural Networks in Hydrology, 2000). To date, ML continues to grow in popularity within hydrology, with many successful streamflow forecasting models developed using RFs (Abbasi et al., 2021), XGBoost (Ni et al., 2020), K-Nearest-Neighbors (KNN) (Khazaee Poul et al., 2019), SVM (Kambalimath & Deka, 2021), and LSTM (Modaresi et al., 2018; Ni et al., 2020; Pham et al., 2020).

Despite the usefulness of ML, an enduring problem when forecasting streamflow is the multiscale change present in streamflow data (Sivakumar, 2017; Sivakumar & Berndtsson, 2010). To address the issue of multiscale change, WTs have been applied successfully in numerous studies, providing improved performance when coupled with data-driven models compared to using data-driven models alone (Santos et al., 2023; Syed et al., 2023; Wang et al., 2022; Yilmaz et al., 2022).

The coupling of two or more data-driven techniques, such as WT-based ML models for streamflow forecasting, has been a rapidly evolving trend in the literature (Ng et al., 2023). Such coupling of data-driven techniques, known as hybridization, has been the focus of recent publications due to improved streamflow forecasting accuracy provided by such hybrid methods (Sharma & Machiwal, 2021).

Since this thesis adopts a WT-based LSTM model for undertaking the first large-sample streamflow forecasting case study, it contributes to the growing literature on hybrid methods for improving streamflow forecasting accuracy. A brief review of ML-based hydrological models is provided in what follows before discussing WTs as a popular feature engineering approach for streamflow forecasting.

2.2.2.1 Machine Learning-based Hydrological Models

ML uses various algorithms to establish a relationship between input and output data. The result of training an ML model is a prediction rule that can be used to predict a target (e.g., streamflow) given input data (e.g., average air temperature, precipitation).

ML models, as a subset of data-driven models, have become increasingly popular in streamflow forecasting, as highlighted in review studies such as (Ng et al., 2023; Tripathy & Mishra, 2024; Yaseen et al., 2015). In part, ML models have grown in popularity as they often produce streamflow simulations and forecasts with superior accuracy compared to process-based hydrological models (Arsenault et al., 2023; Feng et al., 2022; Kratzer et al., 2019; Sabzipour et al., 2023). Furthermore, ML models require less detailed input data compared to process-based models, which can be an advantage in areas where such data are not available (Mount et al., 2016). For instance, the Streamflow Hydrology Estimate using Machine Learning (SHEM) is an ML model that provides accurate and timely proxy streamflow data for inoperative stream gauges (Petty & Dhingra, 2017). Another example is the Neural Runoff Model (NRM) developed using deep learning, which has shown success in making long-term hourly streamflow forecasts (Xiang & Demir, 2020).

ML models vary widely in how they map features to forecasts. For instance, ensemble decision tree methods based on bagging and boosting (e.g., XGBoost) and kernel-based methods, such as SVM and Gaussian process regression, have been applied successfully for streamflow forecasting (Adnan et al., 2021; Pham et al., 2021; Sun et al., 2022; Szczepanek, 2022; Yu et al., 2020).

Another popular ML model applied to streamflow forecasting is the extreme learning machine (ELM), a computationally efficient version of backpropagation-based neural networks (Yaseen et al., 2019). ELM does not use backpropagation (Rumelhart et al., 1986) for training and instead leverages the randomization of certain parameters and the Moore-Penrose pseudoinverse (Golan, 1995) for estimating the remaining parameters. As a result, ELM has been shown to substantially reduce training time compared to a neural network of similar size trained by backpropagation (Huang et al., 2006).

Many studies have successfully applied ELM to streamflow forecasting (Adnan et al., 2023; Adnan, et al., 2021; Difi et al., 2023; Ikram et al., 2023; Nguyen, 2022; Quilty et al., 2023). For instance, Yaseen et al. (2016) demonstrated that ELM produces better forecasting performance than other popular ML models, such as support vector regression (SVR) and the generalized regression neural network, for streamflow forecasting in a semi-arid region in Iraq.

In the last 5 years, there has been a greater focus on using deep learning for streamflow forecasting (Ng et al., 2023). Deep learning architectures based on recurrent neural network (RNNs) are especially suitable for streamflow forecasting and often outperform traditional neural networks, such as multi-layer perceptron (Apaydin et al., 2020). LSTM (Hochreiter & Schmidhuber, 1997), a type of RNN that is easier to train has experienced great popularity in the streamflow forecasting and simulation literature due to its ability to outperform process-based hydrological models across a variety of hydroclimatic scenarios (Arsenault et al., 2023; Dehghani et al., 2023; Man et al., 2023; Nifa et al., 2023; Wi, 2022).

A more recent deep learning architecture, the Transformer (Vaswani et al., 2017), has also shown promise for streamflow forecasting (Castangia et al., 2023). Initially developed for natural language processing (NLP), the Transformer can extract short- and long-memory information from time series without processing the data sequentially as in LSTM, making it quicker to train (Vaswani et al., 2017). In (Liu et al., 2022), a novel double-encoder Transformer model was introduced for streamflow forecasting of the Yangtze River, incorporating a "cross-attention" mechanism to analyze the relation between streamflow and El Niño-Southern Oscillation (ENSO) data. Monthly forecasts for 1998 to 2018 showed that the proposed model outperformed common benchmark models, such as ARIMA, Temporal Convolutional Network (TCN), LSTM, and classic Transformer.

Alongside the development of ML models for streamflow forecasting has been the evolution of two related areas of research: feature engineering and feature selection (Kuhn & Johnson, 2019). Both feature engineering and selection are preliminary stages before training an ML model. Since the features used in an ML model are one of the primary determinants of the model's performance (Chandrashekar & Sahin, 2014; Gada et al., 2021), feature engineering and selection should be carefully considered. Sections 2.3 and 2.4 provide background on both topics.

A relatively recent advance in streamflow forecasting has been the inclusion of WT-based features in ML models, which has been shown to improve forecasting performance for various ML

models (Quilty & Adamowski, 2018). The central aim of this thesis is to demonstrate the impact on performance by including WT-based features in ML-based streamflow forecasting models. Many recent studies have developed WT-based ML models for streamflow forecasting (Kambalimath & Deka, 2021; Khazaee Poul et al., 2019; Santos et al., 2023; Syed et al., 2023; Yilmaz et al., 2022). However, these studies have 1) misapplied WTs and/or 2) did not evaluate the impact of WTs on streamflow forecasting performance using a large, hydroclimatically diverse set of catchments. This thesis aims to address this gap by providing a large-sample benchmark study to demonstrate the impact of WT-based features on streamflow forecasting performance.

The following sub-section provides a brief overview of feature engineering, before describing WTs and their application to streamflow forecasting.

2.3 Feature Engineering

Feature engineering is the process of adding new features (to the original set of features) to improve the performance of an ML model (Kuhn & Johnson, 2019). There are many ways to construct new features. The two most basic methods are: 1) collecting data for a new feature that the modeler believes can help predict the target and 2) deriving a new feature based on existing features (e.g., a 7-day moving average of precipitation). This thesis adopts the second feature engineering method, whereby new features are constructed by applying WTs to existing features. Within the context of the current thesis, where the forecasting problem is formulated as a regression problem, the addition of a new feature amounts to adding a new column to the feature matrix $\boldsymbol{X}$.

An example of a widely employed feature engineering technique is “one-hot encoding” (Rodríguez et al., 2018). This technique converts a categorical feature (e.g., the month of the year) into numerical features consisting of ones and zeros. For instance, to convert a feature indicating the month of the year into a set of numerical features, 12 new features are created, with each new feature corresponding to a particular month. For each dataset row, the feature corresponding to the present month would contain a one, while the remaining 11 features would contain a zero. After constructing the 12 new features, the original categorical feature (i.e., the month of the year) would be removed from the data set.

Another popular feature engineering method is Principal Component Analysis (PCA) (VanDerwarker & Marcoux, 2018). PCA is a statistical technique commonly employed in feature engineering to reduce the number of features in large datasets. PCA helps reduce the risk of overfitting

ML models while attempting to preserve the most relevant information in the features (Jolliffe, 2002). It transforms the original features into a new set of uncorrelated variables known as principal components. PCA has been shown to improve the performance of streamflow forecasting models (Fan et al., 2016). However, one of the drawbacks of PCA is that it is a linear method. Thus, PCA is not ideal for hydrometeorological data since it cannot model nonlinear relationships. Instead, feature engineering methods that can capture complex details embedded within hydrometeorological data should be preferred when forecasting streamflow.

With streamflow data, complex underlying dynamics exist that drive streamflow behaviour. Streamflow data typically appear highly volatile, contain complex transients, and multiple periodicities (Sikorska-Senoner & Quilty, 2021). WTs have been used successfully to analyze streamflow data by separating streamflow data into multiple time series named wavelet and scaling coefficients. The wavelet and scaling coefficients include high- and low-frequency information over varying time scales, respectively (Chong et al., 2019). Furthermore, the wavelet and scaling coefficients extracted from a streamflow time series can be used as new features for ML-based streamflow forecasting. Adopting WTs in this way (i.e., as a feature engineering method) has been shown to improve forecasting streamflow performance (Jahangir et al., 2023).

Using WTs to engineer new features for improving ML streamflow forecasting models is a relatively recent development. Thus, it has been misapplied in numerous studies (Barzegar et al., 2018; Samadianfard et al., 2018; Shi et al., 2018; Yu et al., 2018) due to a lack of best and correct practices in its application. Quilty & Adamowski (2018) estimated that approximately 90% of all papers published on the application of WTs to hydrological and water resources forecasting (at the time of publication) misapplied WTs, resulting in forecasting models that cannot be applied in real-world scenarios since they depend on inaccessible future data at the time of issuing forecasts.

Moreover, among the studies that have properly implemented WTs for streamflow forecasting, none of these studies considered a large sample (on the order of hundreds) of catchments when evaluating improvements in forecast performance due to WTs (e.g., Agarwal et al., 2022; Azarpira & Shahabi, 2021; Dariane & M. Behbahani, 2024; Criswell, 2022; Karsavran et al., 2023; Khatun et al., 2023; Lü et al., 2022; Mazarei Behbahani & Mazarei, 2023; Momeneh & Nourani, 2023; Nourani et al., 2021; Olfatmiri et al., 2022; Rolim et al., 2023; Shaikh et al., 2022; Syed et al., 2023; Vogeti et al., 2022; Wei et al., 2023, 2024; Xiao & Wang, 2021; Yuan & Forshay, 2022; Zakhrouf et al., 2020). This

lack of comprehensive assessment raises questions about the generalizability of the improvements provided by WTs in these models.

This thesis seeks to overcome these gaps in the existing literature by conducting a large-sample benchmark study. This benchmark study focuses on evaluating how features engineered using WTs impact the performance of LSTM when applied to short-term streamflow forecasting. This study encompasses over 600 catchments within the contiguous United States (CONUS). Additionally, this research aims to clarify the specific hydroclimatic conditions under which WTs substantially enhance LSTM-based streamflow forecasting accuracy.

The following sub-section explores WTs in detail. This discussion introduces the MODWT as a feature engineering method and describes how it can (and has been) used for hydrological forecasting (and streamflow forecasting, in particular).

2.3.1 Wavelet Transforms

WTs are mathematical methods for decomposing a time series into multiple time series, i.e., wavelet and scaling coefficients (Percival & Walden, 2000). These coefficients capture time-frequency information about the original time series and are typically ordered from higher to lower frequencies. WTs can be considered microscopes for time series, able to extract rapidly changing features of a time series localized in time (wavelet coefficients) and slowly changing trends (scaling coefficients).

A wavelet, or wavelet filter, is a waveform with a limited duration/length and an average value of zero. This means it oscillates above and below zero. The name ‘wavelet’ means small wave, and it captures the idea that a wavelet function is a short-lived wave that spends most of its time near zero, unlike, for example, a sine wave that continues indefinitely. Section 4.1 shows some examples of common wavelet filters.

In the case of the DWT, the canonical WT variant used in time series applications, the wavelet coefficients are obtained by convolving a different “stretched” version of a chosen wavelet filter with progressively down-sampled (shortened) versions of the original time series. The scaling filter is formed by modifying the wavelet filter (see Section 3.2). This scaling filter is then used to construct the scaling coefficients. Two main parameters need to be specified to estimate wavelet and scaling coefficients using DWTs: the type of wavelet filter and the decomposition level.

Intuitively, DWT computes similarity (localized in time) between a time series and a wavelet (or scaling) filter. The type of wavelet filter should be selected such that its waveform is similar in shape to features of the time series, as it will enable improved feature extraction (Percival & Walden, 2000). Higher decomposition levels result in using more stretched versions of a wavelet or scaling filter, which translates to an increase in the number of time series observations included when computing local similarity. As the decomposition level increases, the wavelet coefficients extract additional information on higher periodicities. The remaining frequency content of the time series is then stored in the scaling coefficients. Thus, high decomposition levels lead to scaling coefficients that capture broader trends in a time series than lower decomposition levels.

The number of wavelets is usually denoted by J. The user must set this value. The selection of J is often made based on the periodicities the user is interested in extracting from a given time series.

However, a key drawback of the DWT is that the wavelet and scaling coefficients are shorter than the original time series. As the decomposition level increases, the length of the wavelet and scaling coefficients decreases. Due to the length mismatch between the original time series and the DWT-generated wavelet and scaling coefficients, the wavelet and scaling coefficients cannot directly be used as new features in forecasting models.

To address this issue, the DWT multiresolution analysis (DWT-MRA) is commonly used, which results in detail and approximation coefficients that match the length of the original time series (Percival & Walden, 2000). However, the DWT-MRA suffers from look-ahead bias and cannot be used properly for forecasting (Quilty & Adamowski, 2018).

The drawbacks of the DWT and DWT-MRA can be resolved by using the MODWT instead, a variant of the DWT. The MODWT can obtain wavelet and scaling coefficients that retain the same length as the original time series while avoiding look-ahead bias – making this method useful for engineering new features for forecasting (Quilty & Adamowski, 2018). Thus, the MODWT is used in this research for feature engineering by obtaining wavelet and scaling coefficients from the features for forecasting streamflow. Another option is using the à trous algorithm (AT), which avoids look-ahead bias (Quilty et al., 2018). However, the AT is not explored here for two reasons: 1) since it uses an ad-hoc approach to calculate the wavelet coefficients (Aussem et al., 1998) and 2) to allow the focus to be on the MODWT. However, future research could compare the MODWT and the AT in a large-sample streamflow forecasting case study to better evaluate the differences between the two WTs.

The next sub-section provides a brief overview of the MODWT within the context of feature engineering, including a discussion of relevant studies using MODWT in hydrology. Mathematical details on the MODWT can be found in Section 3.2.

2.3.1.1 Maximal Overlap Discrete Wavelet Transform (MODWT)

The MODWT (Percival & Walden, 2000), an extension of the DWT, is designed to overcome some of the limitations associated with the DWT. Two characteristics of the DWT make the method unsuitable for engineering features from existing time series (e.g., meteorological features, such as daily maximum and minimum air temperature). These two characteristics are down-sampling and look-ahead bias (Quilty & Adamowski, 2018).

Down-sampling refers to the shortening of the wavelet and scaling coefficients produced by the DWT when applied to a time series of length N. Each consecutive set of wavelet and scaling coefficients (produced during the decomposition of a time series at level $j = 1, \dots, J$) contain fewer entries (less than N) compared to the original time series. In a regression context, all features need to have the same length, N. Hence, the wavelet and scaling coefficients produced by the DWT cannot be used directly as new features. The MODWT solves the problem of down-sampling by producing wavelet and scaling coefficients that match the length of the original time series. This enables MODWT wavelet and scaling coefficients to be used directly as features.

Look-ahead bias refers to using time series (e.g., precipitation) observations from the future in a forecast or prediction. In practice, observations of the target time series would only be known up to (and possibly including) the present time t. Future observations (i.e., observations of the time series at time $t + 1$ and beyond) would not be known at time t. The DWT (and DWT-MRA) requires knowledge of a time series at time $t + 1$ and beyond. This makes the DWT infeasible for real-world forecasting scenarios. In contrast, the MODWT only requires historical observations of a time series up to and including time t. Hence, future observations of a given time series are not required. This makes MODWT feasible to implement into existing ML-based forecasting workflows.

2.3.1.1.1 MODWT for Feature Engineering

When building forecasting models, this research considers tabular data (i.e., features and the target as columns, observations as rows). Thus, using the MODWT for feature engineering amounts to

adding new columns to the original feature matrix $\boldsymbol{X} \in \mathbb{R}^{N \times d}$. Here, N is the number of observations, and d represents the number of features.

As an example, Figure 1 shows a precipitation time series P decomposed into MODWT wavelet and scaling coefficients using a decomposition level of six ($J = 6$) and the Haar wavelet filter. The first six sub-series on the right (starting from the bottom) correspond to the wavelet coefficients for $j = 1, \ldots, 6$. The seventh (uppermost) sub-series on the right are the scaling coefficients for $j = 6$. The MODWT typically omits the scaling coefficients for levels $j = 1, \ldots, J-1$ since this information is jointly captured by the scaling coefficients at $j = J$ and the wavelet coefficients for levels $j = 1, \ldots, J-1$.

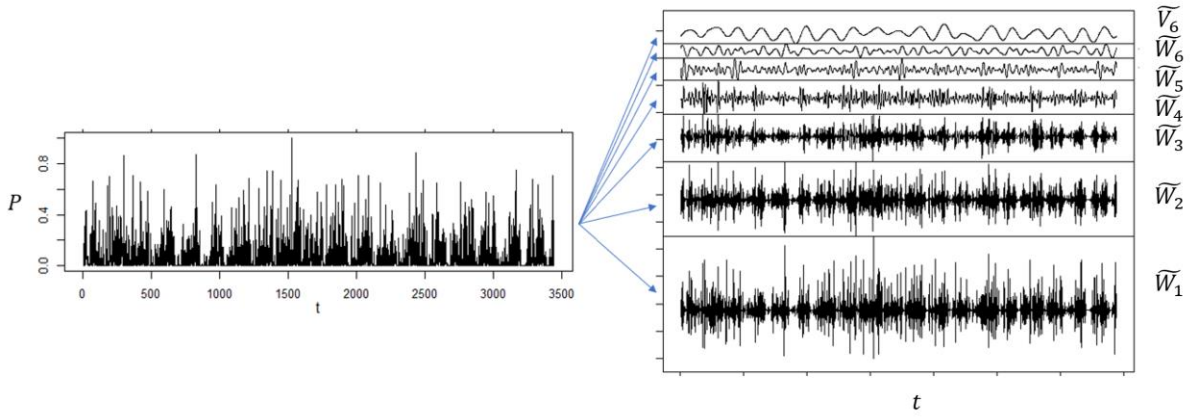


Figure 1. Visualization of decomposing a time series into multiple sub-series using MODWT

To generate Figure 1, two parameters must be specified for the MODWT. The first is the maximum decomposition level ($J = 6$ in the above example). The maximum decomposition level dictates the number of new features generated by the MODWT, equal to $J + 1$. If the MODWT is used to decompose each feature in $\boldsymbol{X}$, then a total of $d \times (J + 1)$ new features will be generated. The second parameter is the type of wavelet filter. The wavelet filter should be selected to maximize the amount of relevant time-frequency information extracted from each feature. This thesis adopts an exhaustive search through pre-selected wavelet filters to identify the best one for each catchment (according to the validation set Nash-Sutcliffe Efficiency or NSE – see Section 4.2.5). However, a recent study by Mazarei Behbahani & Mazarei (2023) indicated that wavelet entropy may be useful for identifying an optimal wavelet filter (and decomposition level) for a given time series. Although it is outside the scope

of this thesis, the use of wavelet entropy for wavelet filter and decomposition level selection is an interesting topic for a future large-sample study on WT-based streamflow forecasting.

The final step in preparing MODWT wavelet and scaling coefficients for use in an ML model is to remove the boundary coefficients to avoid look-ahead bias. This issue is described in detail in Section 4.2.2.

Given the above description of how MODWT is used to engineer features for ML models, the next sub-section reviews recent literature on the use of MODWT in hydrological forecasting. Of note, studies using MODWT for streamflow forecasting are rare, given that the use of MODWT for hydrological forecasting is still a recent practice (Quilty & Adamowski, 2018).

2.3.1.1.2 MODWT in Hydrological Forecasting

Several hydrology studies have assessed the use of MODWT or MODWT-MRA (which cannot be properly used for forecasting) to enhance the performance of hydrological simulations and forecasts. Below is a brief review of select publications to offer context as to how MODWT and the MODWT-MRA have been applied in the hydrological forecasting literature.

Prasad et al. (2017) evaluated a wavelet-hybrid ANN model with an iterative input selection (IIS) algorithm (IIS-W-ANN) for forecasting monthly water levels and compared it against the M5 Tree model. The models were developed for three sites within Australia's Murray-Darling Basin, using a global predictor matrix that includes water levels and meteorological data. The IIS algorithm was employed to select the best features, which the MODWT-MRA then decomposed. Model performance was assessed using several statistical measures, including NSE, correlation coefficient, and Willmott's Index. The IIS-W-ANN model outperformed the standalone ANN and M5 Tree models and their IIS-enhanced versions, demonstrating improved accuracy in water level forecasting. The main issue with this study is that the authors adopted the MODWT-MRA instead of the MODWT. Therefore, the forecasting models based on the MODWT-MRA suffer from look-ahead bias, making it difficult to assess the usefulness of WTs.

Snow-depth forecasting is an important area of research in hydrological forecasting. Accurate snow depth forecasts are essential for predicting water availability, especially in regions where snowmelt contributes substantially to water supply. Adib et al. (2021) introduced a novel forecasting algorithm that integrates various WTs, specifically DWT, MODWT, MODWT–MRA with ARIMA,

and ML models (e.g., artificial neuro-fuzzy inference system, ANFIS) to predict one day ahead snow depth at the North Fork Jocko snow telemetry (SNOTEL) station in Missoula, Montana, USA. The WT-based models showed superior accuracy compared to standalone ARIMA and ML models. The MODWT-MRA-ARIMA-ANFIS model, in particular, demonstrated exceptional forecasting accuracy with a correlation coefficient (R2) of 0.9998, RMSE of 1.66 cm, MAE of 1.04 cm, and NSE of 0.9998, although performance was marginally better than the remaining ARIMA and ML models. While the MODWT-MRA is inappropriate for this forecasting application, the performance of the standalone ARIMA and ML models demonstrated potential for further research in snow-covered region analysis.

Groundwater level (GWL) simulation and forecasting are other important domains in hydrological research. Chidepudi et al. (2023) address the critical need for accurate groundwater level simulations crucial for effective water resource management, particularly under the challenges posed by climate change. Traditional physically-based models like MODFLOW (Langevin et al., 2017) are often limited by high computational demands and complex calibration processes. The authors utilize three deep learning models: LSTM, Gated Recurrent Unit (GRU), and Bidirectional LSTM (BiLSTM), comparing their performance in simulating GWLs with varying low-frequency characteristics—inertial, annual, and mixed. They introduce a preprocessing step using the MODWT to better capture the relevant frequency content from input variables like precipitation and air temperature. The findings indicate that MODWT preprocessing substantially improves the accuracy of GWL simulations, particularly for inertial-type GWLs where low-frequency variations dominate. The MODWT-enhanced models (MODWT-DL) performed better than standalone models, especially when using raw data inputs. The improvement was less pronounced but still substantial for annual and mixed GWL types, suggesting that the frequency-specific information extracted via MODWT is crucial for enhancing model performance across different GWL variability types.

A central area of research in hydrology is flood forecasting, of which rainfall and runoff forecasting are crucial components. Recently, Amini et al. (2024) presented an enhanced approach for forecasting rainfall by integrating the MODWT preprocessing technique with automatically tuned deep neural networks (DNNs), using Daubechies wavelets for improved accuracy in urban catchments. The forecasted rainfall was then used in the stormwater management model (SWMM) (US EPA, 2014) to produce urban runoff forecasts. Applied to Tehran's East Drainage Catchment (EDC), the study used random search for hyper-parameter selection of both univariate and multivariate DNNs. The authors found substantial improvements in prediction performance when using the MODWT technique with

DNNs, raising NSE values from 0.54 to 0.97 compared to standalone DNNs. The results highlight the effectiveness of automatically tuned MODWT-DNN models for accurate rainfall hyetograph and peak flow predictions in urban areas with short concentration times. This study further adds to the existing literature surrounding the usefulness of MODWT in hydrological forecasting.

Lakes play a crucial role in supporting global biodiversity and ecosystems, with water level being a key physical variable that has profound socio-economic and environmental impacts (Molinos et al., 2015). Accurate forecasting of lake water level changes is essential for effective water budget management, infrastructure protection, navigation, water quality, flood control, and sustainable water resources management (Coulibaly, 2010). Barzegar et al. (2021) focused on enhancing the accuracy of water level forecasts in Lake Michigan and Lake Ontario up to three months in advance by integrating the MODWT preprocessing with a hybrid Convolutional Neural Network (CNN) and LSTM model, or CNN-LSTM model. The performance of CNN-LSTM was compared with that of ML models, such as RF and SVR, which were also preprocessed with BC-MODWT. Data from January 1918 to February 1988 was used for model calibration, and data from March 1988 to December 2018 for validation. Features for ML models were selected through a correlation-based feature selection and particle swarm optimization method, while for CNN-LSTM models, the selection was automatic. The study utilized various wavelet types for data decomposition. The MODWT-based CNN-LSTM model outperformed both standalone CNN-LSTM, RF, and SVR and MODWT-based RF and SVR models, demonstrating superior accuracy in water level forecasting for both lakes, with the best results achieved using the Haar mother wavelet. The authors' findings indicate that the MODWT-based CNN-LSTM model is a promising approach for improving the accuracy of lake water level forecasts, highlighting the usefulness of MODWT.

The studies highlighted in this sub-section, which cover a broad range of hydrological forecasting domains (streamflow, rainfall, groundwater level, flood, and lake water levels), provide positive evidence regarding the use of MODWT to enhance ML-based hydrological forecasting. However, between generating new features using WTs, such as the MODWT, and training an ML model, the user must determine the final set of features used in the ML model. The following sub-section addresses the important topic of feature selection.

2.4 Feature Selection

Feature selection, also known as input variable or attribute selection, is a critical step in developing ML models. It refers to selecting a subset of relevant features for a model (Kuhn & Johnson, 2019). Feature selection is useful for three reasons: simplifying models to make them easier to interpret, shortening training times, and mitigating model overfitting (Chandrashekar & Sahin, 2014).

Feature selection techniques are generally divided into three categories: filter, wrapper, and embedded methods (Chandrashekar & Sahin, 2014; Guyon & Elisseeff, 2003; Motoda, 2007). This research focuses only on filter methods due to their computational efficiency. Both wrapper and embedded methods are computationally demanding, making their use for large-sample studies less appealing.

2.4.1 Filter-based Feature Selection

Filter-based feature selection methods evaluate the relevance of features by ranking them according to some metric that quantifies how strongly a feature is related to a target. These methods are generally faster and less computationally expensive than other methods as they do not require the involvement of the model to be used for predictive purposes post-feature selection. Examples of filter methods include Pearson correlation, Chi-square test, and information gain (Chandrashekar & Sahin, 2014).

Partial correlation input selection (PCIS) (May et al., 2008) is a linear filter method that has been used in recent hydrological simulation and forecasting publications (Quilty et al., 2023; Ren et al., 2021; Sikorska-Senoner & Quilty, 2021; Snieder et al., 2020). PCIS uses partial correlation (a statistical technique used to quantify the linear relationship between two features while controlling for the effect of one or more other features). In particular, PCIS is used to identify which features are most relevant to a target variable by isolating the direct relationships between each feature and the target, independent of other features.

Another popular filter method in hydrology is partial mutual information (PMI) selection (PMIS) (May et al., 2008; Sharma, 2000b, 2000a; Sharma et al., 2000). PMI is, in practice, computed by estimating conditional mutual information (CMI). CMI is a concept in information theory that measures the amount of information shared between two random variables given the knowledge of a third random variable (or set of random variables). CMI quantifies how much knowing one of the

variables reduces uncertainty about another when a third random variable (or set of random variables) is known. CMI depends on the information-theoretic notion of differential entropy, which measures a random variable's uncertainty. Within the context of feature selection, each feature in a feature set corresponds to a random variable.

CMI must be estimated using parametric or non-parametric methods, such as the KNN algorithm. Estimation is needed since the true probability distribution of a feature can never be known in practice. The estimation of CMI required to perform PMIS introduces high computational demands, particularly for datasets with many features and many observations (Molavipour et al., 2021). Once CMI is computed for each feature in a feature set, features are selected based on a minimum CMI threshold.

Comparing PCIS and PMIS, PCIS is computationally efficient but can only measure linear correlation between features. Using concepts from information theory, PMIS can measure general nonlinear relationships between features but can be much more computationally demanding.

To overcome the drawback of PMIS being computationally demanding, the use of Edgeworth approximations (EA) has been proposed in the literature (Quilty et al., 2016). There are two benefits of the EA method of estimating CMI. First, the EA method is parameterless, which reduces the subjective aspect of parameter selection that plagues other CMI estimation methods, such as KNN. Second, the EA method has been shown to outperform other mutual information (MI) estimation methods, such as KNN and kernel density estimation (KDE) (Hulle, 2005). Additionally, the EA version of the PMIS method has been shown to perform well in hydrological simulation and forecasting applications (Quilty et al., 2023; Quilty & Adamowski, 2020, 2021; Sikorska-Senoner & Quilty, 2021). This thesis makes use of the EA version of PMIS for feature selection.

To date, only a single study has compared EA against different filter methods for streamflow forecasting. Ren et al. (2020) examined the efficacy of eight filter methods for one month ahead streamflow forecasting, including PCIS, PMIS, and EA. The authors used features selected by these methods as input to three regression models (multiple linear regression (MLR), ensemble extreme learning machine (enELM), and KNN regression). The study used data from three CAMELS stations to compare these filter methods' performance in one-month ahead streamflow forecasting. Notably, PCIS emerged as the most effective filter method when resampling was used to select the final model from the candidate combinations. However, the streamflow time series for the three stations showed

high autocorrelation, which could have led to this result. In cases where streamflow data show low autocorrelation, such as in arid catchments, it is important to consider nonlinear feature selection methods, such as EA, due to the importance of precipitation. A similar result was found in Sikorska-Senoner & Quilty (2021). Given that the CAMELS dataset spans multiple climates, this research adopts the EA method due to its ability to measure general nonlinear relationships amongst variables and its computational efficiency.

2.4.1.1 Edgeworth Approximation-based Conditional Mutual Information

Regardless of the methods used, the goal of feature selection is to find the most relevant features that can help in building a robust and accurate ML model while:

1. Ignoring irrelevant and redundant features to avoid overfitting
2. Reducing computational cost
3. Improving model interpretability

Usually, these desired conditions cannot be met simultaneously. Highly accurate models often come at the cost of high computational demand and low model interpretability. Thus, due to the high computational cost of training LSTMs across hundreds of catchments, the EA method was adopted to reduce the computation time associated with (nonlinear) feature selection, offering similar performance to the popular PMIS method at a reduced computational cost.

The EA method shares the same benefits of PMIS, mainly the ability to measure general nonlinear relationships between variables using information theory, but is considerably more computationally efficient when the number of features in the dataset is smaller than the number of measurements (Quilty et al., 2016). This condition is commonly met for streamflow forecasting studies, and it is the case in this thesis. The EA method has shown very strong performance for a variety of feature selection applications in hydrology, some of which are described below.

In Quilty et al. (2016), the EA method was introduced and shown to provide competitive performance alongside PMIS and two other CMI-based feature selection approaches in terms of selection accuracy on benchmark feature selection datasets. However, the EA method was orders of magnitude more computationally efficient than competing methods.

Quilty & Adamowski (2021) demonstrated further evidence for the effectiveness of EA for monthly rainfall forecasting in the Awash River Basin (Ethiopia). Of the nonlinear feature selection methods considered in the study, EA resulted in the highest number of best-performing rainfall forecasts compared to PMIS and iterative input selection. For instance, the best-performing model for the Bantu Liben station in terms of RMSE was an ELM model with EA-based feature selection. The study also emphasized the computational efficiency of EA compared to PMIS and iterative input selection.

In Sikorska-Senoner & Quilty (2021), ML models were used to correct residuals from an ensemble of hydrological models used for streamflow simulation. The study evaluated different model configurations that couple the ensemble hydrological model with eight ML models for correcting residuals, including MLR, KNN, Second Order Volterra (SOV) series model, ANN, XGBoost, and RF. Feature selection was performed using PCIS and EA; PCIS was coupled with MLR, and EA was coupled with SOV, ANN and KNN. Although the study showed that RF and XGBoost resulted in the best residual corrections, EA was an effective feature selection method across the various nonlinear ML models.

The studies mentioned above provide credence to the utility of EA, particularly for nonlinear ML models. As will be discussed in the following sub-section, the LSTM is a nonlinear ML model and, hence, suitable for coupling with EA based on existing literature (Sikorska-Senoner & Quilty, 2021).

2.5 Long Short-Term Memory Networks

LSTM is a type of RNN introduced in Hochreiter & Schmidhuber (1997). RNNs are known for their ability to process sequential data by maintaining an internal state across time steps. However, conventional RNNs suffer from vanishing and exploding gradients when trained via backpropagation through time (Werbos, 1988; Williams, 1989), making them unable to learn long-term dependencies effectively (Bengio et al., 1994; Pascanu et al., 2013).

LSTMs are designed to solve this problem by introducing mechanisms known as "gates". The LSTM cell contains an input gate, a forget gate, and an output gate, which control the flow of information inside a unit, allowing it to remember or forget information over long sequences (Graves, 2012). This gating mechanism prevents the gradients from “vanishing” during backpropagation, allowing LSTMs to learn dependencies over long sequence lengths.

LSTMs have been applied successfully in various domains, including natural language processing, speech recognition, time series forecasting, and music generation, due to their ability to model temporal data (Graves et al., 2013; Sutskever et al., 2014). More recently, the transformer architecture (Vaswani et al., 2017) has replaced LSTM in many of these areas due to its generally superior performance on sequence-based problems that are less sensitive to the system's state.

One area where the transformer architecture has yet to show superiority over LSTM is in the domain of time series forecasting (Zeng et al., 2022). The LSTM architecture remains a state-of-the-art method for streamflow simulation (Arsenault et al., 2023; Liu et al., 2024) and forecasting (Girihagama et al., 2022). Due to its ability to learn temporal dependencies and easily incorporate previous streamflow observations, LSTM can outperform semi-distributed process-based streamflow forecasting methods that consider data assimilation (Sabzipour et al., 2023).

A streamflow forecasting model can be built with LSTM by feeding it a sequence of past observations of meteorological features (e.g., precipitation, temperature, humidity, wind speed), and training it to predict future streamflow values. The gates within the LSTM cells can learn the most important meteorological features and determine the past observations to retain (or forget) for accurate forecasting.

It is important to note that the performance of LSTM models can be affected by various factors, including the selection of features, the input sequence length (i.e., the look-back period) and the complexity of the network architecture (the presence of dropout layers, the number of hidden units, the number of LSTM layers, etc.) Therefore, careful training and validation of LSTM models are necessary to achieve optimal streamflow forecasting performance. The next sub-section completes the background chapter by highlighting a major gap in the WT-based hydrological forecasting literature this thesis seeks to address.

2.6 A Gap in the Literature: Large-Sample Evaluation of WT-based Streamflow Forecasting Models

To the best of the author’s knowledge, no study to date has developed and evaluated WT-based hydrological forecasting models using a large sample (on the order of hundreds) of catchments. Thus, the literature lacks information to confirm whether WTs can *generally* improve streamflow forecasts across a wide range of hydroclimatic conditions.

Specifically, two issues are commonly found in the literature on WT-based streamflow forecasting. First, studies often show that a proposed WT-based streamflow forecasting model performs well using a small number of catchments (Dariane & M. Behbahani, 2024; Wei et al., 2024). Second, case studies often show that a WT-based streamflow forecasting model performs well in a given geographical area that is not diverse in its hydroclimatology (Olfatmiri et al., 2022; Shaikh et al., 2022).

In the first case, a general claim about the strength of the proposed model is made using a small sample size. In the second case, it is not clear whether the proposed model is expected to generalize well when used in other hydroclimates. In addition, such studies often use WTs, such as the DWT-MRA or the MODWT-MRA, that suffer from look-ahead bias and cannot be used correctly for streamflow forecasting (Quilty & Adamowski, 2018).

In this sub-section, relevant studies on WT-based streamflow forecasting models from the past two decades are reviewed to emphasize the abovementioned issues related to evaluating model performance: the small number of catchments and the lack of hydroclimatic diversity in the catchments used in their case studies. This literature review is not intended to be exhaustive; the goal is to illustrate an important gap in the WT-based streamflow forecasting literature where there is a lack of emphasis on large-sample evaluation of WT-based forecasting models.

Upon reviewing the most cited papers on WT-based streamflow forecasting (see Appendix A for a listing of select highly cited papers), the number of catchments considered in these studies ranged between one and six, with one catchment being used in most case studies.

In one of the most highly cited WT-based streamflow forecasting studies (Adamowski & Sun, 2010), the efficacy of coupled WT–ANN was explored for forecasting streamflow one and three- day(s) ahead in two semi-arid, non-perennial watersheds in Cyprus. It was found that WT–ANN models based on the DWT-MRA produced more accurate forecasts than traditional ANN models. The enhanced accuracy was attributed to the ability of WTs to decompose time series data, enabling the ANN to utilize information across various time and frequency resolutions effectively. The findings suggest that WT–ANN models are highly accurate for one day ahead forecasting and moderately accurate for three days ahead forecasting in semi-arid regions with intermittent streamflow. The authors indicated that potential improvements in the forecast might be obtained by incorporating meteorological variables, such as air temperature and rainfall. They recommended further research on the models' application in

different environments, extended lead times, and comparison of different types of WT and ANNs. The study has two main drawbacks. First, due to look-ahead bias, the DWT-MRA cannot be used properly for real-world forecasting. Second, only two catchments were considered, making it difficult to generalize the results.

Kişi (2009) proposed a WT-ANN model for forecasting daily intermittent streamflow that used the DWT-MRA. The study used daily intermittent streamflow data from two stations in the Thrace Region of Turkey (Uzunkopru and Babaeski stations). The results showed that their proposed model substantially improved the forecast accuracy compared to the benchmark ANN. The author suggested that this model can be used for streamflow forecasting in other regions with similar water systems and weather patterns. This study, however, lacks diversity in the geographical and hydroclimatic data used to evaluate their proposed models and uses the DWT-MRA, which cannot be used properly for forecasting due to look-ahead bias.

Another influential study (Karthikeyan & Nagesh Kumar, 2013) proposed WT- and empirical mode decomposition (EMD)-based autoregressive moving average (ARMA) models for forecasting monthly streamflow volumes and precipitation. The proposed models decomposed the monthly streamflow volumes and precipitation time series into sub-series using DWT-MRA and EMD. Afterwards, the sub-series generated by DWT-MRA and EMD were forecasted for both monthly streamflow volumes and precipitation using ARMA models. The forecasts across the sub-series were then added together to achieve the overall DWT-MRA- and EMD-based forecasts for streamflow volumes and precipitation. The results showed that the proposed models outperformed traditional ARMA models regarding forecasting accuracy. Overall, the study demonstrated the potential of DWT-MRA- and EMD-based ARMA models for forecasting hydrological time series. However, it is important to note that the DWT-MRA considers future data use. Therefore, the DWT-MRA-based models adopted by Karthikeyan & Nagesh Kumar (2013) cannot be used properly for forecasting in real-world scenarios. Notably, of the reviewed studies in Appendix A, Karthikeyan & Nagesh Kumar (2013) explored the highest number of streamflow gauging stations and had the most diversity among stations. Four streamflow gauging stations in the USA were considered: Mississippi River at Clinton (IA), North Fork River near Tecumseh (MO), War Eagle Creek near Hindsville (AR), and Lopez Creek near Arroyo Grande (CA).

Outside of the most highly cited papers using WTs for streamflow forecasting, more recent studies have combined WTs and deep learning models to improve streamflow forecasting performance. In these studies, LSTMs are typically used.

In Ni et al. (2020), two hybrid LSTM models, based on WTs (WLSTM) and convolutional layers (CLSTM), were proposed for monthly streamflow and rainfall forecasting. The WLSTM model used the AT for feature engineering. The models were developed using monthly streamflow volume data from Cuntan and Hankou stations in the Yangtze River basin, China, and monthly rainfall data from Jinan and Wenjiang (Chengdu) stations (also in China). The results showed that the WLSTM and CLSTM models outperformed the benchmark ANN model on multiple metrics, including NSE. The study demonstrated the potential of using hybrid models for streamflow and rainfall forecasting and provided insights into the performance of different LSTM architectures. While the authors adopted a WT that can be used properly for forecasting (i.e., AT), their study only focused on two streamflow gauging stations.

Zuo et al. (2020) proposed daily streamflow forecasting models that coupled LSTM with variational mode decomposition (VMD), ensemble EMD (EEMD), and WTs. The authors explored different strategies for using the features engineered by VMD, EEMD, and WT in their models, focusing on addressing boundary conditions. While the authors claimed to use DWT, the study did not include sufficient details to ascertain how the wavelet and scaling coefficients were obtained. The authors developed their models using daily streamflow data from two catchments in China, the Han River and the Jing River. The study found that the proposed models outperformed three benchmark models regarding accuracy and robustness, particularly in dealing with boundary conditions. However, due to a lack of details on developing the WT-based models, it is unclear if they properly addressed the boundary conditions outlined in Quilty & Adamowski (2018). Thus, whether the WT-based models developed in this study are suitable for real-world streamflow forecasting is unknown. Furthermore, only two catchments were used in this study, limiting the generalizability of its findings.

A more recent study by Jahangir et al. (2023) developed a quantile-based encoder-decoder framework based on LSTMs for daily streamflow forecasting. The proposed models also considered the case where MODWT was used for feature engineering based on the best practices from Quilty & Adamowski (2018). The MODWT-based models were benchmarked against the same model without

MODWT-engineered features alongside LSTMs coupled to dense networks (with and without MODWT-engineered features). The authors developed models for three catchments from the CAMELS dataset with different hydroclimates. The authors found that the quantile-based encoder-decoder framework combined with MODWT provided the best performance, producing daily streamflow forecasts with high reliability across the three catchments. While this study introduced a promising WT-based model for streamflow forecasting, the study is still limited in the number of catchments used for developing and evaluating improvements in performance due to WTs.

A clear pattern emerges from these studies: WT-based models are generally presented in the literature as promising methods for streamflow forecasting. However, these studies only consider a small number of catchments when evaluating their models and/or did not consider a collection of catchments with diverse hydroclimates. This limits the ability to generalize the results from these studies and makes it difficult to understand the potential improvements in performance due to WTs that can be expected if these models were applied to catchments with hydroclimates different than those considered in the respective studies.

Thus, it is clear from the literature that there is a long-standing interest (over 15 years) in using WT-based streamflow forecasting models. It is also clear that the literature would benefit from further research into applying WT-based streamflow forecasting models to many catchments with diverse hydroclimates. Such research would enable a more comprehensive understanding of these models' performance (and potential transferability) across a wide range of hydroclimatological conditions. The goal of this thesis is to address this gap. In the next chapter, the data and methods used to develop the WT-based streamflow forecasting models in this research are described in detail.

Chapter 3 Methods

In this chapter, the data and methods used to develop the WT-based streamflow forecasting models is described. The methods include the MODWT, LSTM, and the EA feature selection algorithm.

3.1 Data

The CAMELS dataset collects catchment attributes and hydrometeorological time series for 671 catchments across the CONUS. The dataset is composed of two parts. The first part contains a collection of catchment attributes that describe each catchment (Addor et al., 2017; Newman et al., 2022). Catchment attributes are a catchment's physical, climatic, and land use characteristics that influence water movement through the landscape. The interaction between different catchment attributes affects how water is stored and transferred in a catchment. Many catchment attributes have been developed over the years, serving a critical role in many hydrological studies (Tarasova et al., 2024). The second part comprises a hydrometeorological time series for each catchment (Newman et al., 2014; Newman et al., 2015). The data structure is identical across all 671 catchments included in CAMELS, which includes the date, streamflow (mm/day), daylight (s), precipitation (mm/day), solar radiation (W/m2), snow water equivalent (mm/day), maximum and minimum air temperature (°C), and vapour pressure (Pa). All time series are measured on the daily scale from 01/01/1980 to 12/31/2014.

The streamflow data were obtained from a database maintained by the United States Geological Survey (USGS), known as Geospatial Attributes of Gages for Evaluating Streamflow (GAGES-II). This database contains information for over 9000 stream gauges maintained by the USGS. Streamflow data in CAMELS is a subset of the streamflow records in GAGES-II. Specifically, the streamflow records in CAMELS correspond to stream gauges in GAGES-II that the USGS marks as “reference” gauges, which indicate gauges in catchments that have had minimal impact by humans (e.g., intervention due to engineering projects). To arrive at the final 671 catchments included in CAMELS, further filtering was done on the GAGES-II reference gauges using information from the Hydro-Climatic Data Network (HCDN) dataset (Slack & Landwehr, 1992). A gauging station was included in the CAMELS dataset if it met the following criteria:

1. It had at least 20 years of complete flow data between 1990 and 2009 and active as of 2009.
2. It was listed as a GAGES-II reference gauge.

3. The catchment includes less than 5% imperviousness as measured by the National Land Cover Database (Jin et al., 2013).
4. It passed a manual survey of human impacts in the catchment by local Water Science Center evaluators (Falcone et al., 2010).

CAMELS uses the Daymet dataset (Thornton et al., 2022) to derive all meteorological features. Daymet is a daily, gridded dataset with a 1 km by 1 km grid resolution over the CONUS and southern Canada. All meteorological features in Daymet are derived from daily observations of precipitation and air temperature. For the subset of CAMELS used in this thesis, the value of each meteorological feature on a given day and for a particular catchment is an areal average value across all grids in the catchment. CAMELS also contains the same meteorological features calculated using two alternate spatial configurations: Hydrologic Response Units (HRU) and elevation bands. Due to its simplicity, this thesis uses the meteorological features calculated using areal averaging (or the "lumped" method). Further details on the Daymet data set, as well as the estimation methods used to derive the meteorological features from air temperature and precipitation observations, can be found in (Thornton et al., 2022; Thornton et al., 2000; Thornton et al., 1997; Thornton & Running, 1999).

Of the 671 catchments in the CAMELS dataset, 620 were used. The other 51 catchments were filtered out due to missing records or insufficient data (i.e., catchments containing fewer than 30 years of data).

The following three sub-sections briefly describe the formulation of MODWT, LSTM, and different quantities based on information theory that are used in the adopted feature selection method, EA.

3.2 Maximal Overlap Discrete Wavelet Transform

As mentioned previously, the DWT cannot be used to engineer new features because the length of the wavelet and scaling coefficients are less than the length of the original time series. This is because the DWT wavelet and scaling filter coefficients are computed using a convolution-like operation that results in coefficients of smaller length than the original time series. For this reason, the MODWT, a variant of the DWT that generates wavelet and scaling coefficients of the same length as the original time series, is used.

Given a time series $X \equiv \tilde{V}_0$ (e.g., precipitation), MODWT generates $J+1$ new decomposed time series $\widetilde{W_1}, \widetilde{W_2}, \ldots, \widetilde{W_J}$ and $\widetilde{V_J}$, each of the same length as X, using the following equations (Percival & Walden, 2000):

$$\widetilde{W}_{j,t} \equiv \sum_{l=0}^{L_j-1} \tilde{h}_l \, \tilde{V}_{j-1,t-2^{j-1}l \bmod N} \quad \textbf{(1)}$$

$$\tilde{V}_{j,t} \equiv \sum_{l=0}^{L_j-1} \tilde{g}_l \, \tilde{V}_{j-1,t-2^{j-1}l \bmod N} \quad \textbf{(2)}$$

where j is the decomposition level with $j = 1, \ldots, J$; t is the time index with $t = 0,1, \ldots, N-1$; $\tilde{h}$ is the wavelet filter; $\tilde{g}$ is the scaling filter; L_j is the length of $\tilde{h}$ and $\tilde{g}$; and mod refers to the modulo operator. The mod operation makes the time index wrap around to the beginning of $\tilde{V}_{j-1}$ when the index surpasses the end of the time series. Hence, the computation of the MODWT wavelet and scaling coefficients can be reduced to a circular convolution operation between the $\tilde{V}_{j-1}$ and the MODWT wavelet and scaling filters, respectively.

The j^{th} level MODWT wavelet and scaling coefficients $\widetilde{W_j}$ and $\tilde{V}_j$, respectively, have a length of (Bašta, 2014):

$$L_j \equiv \left(2^j - 1\right)(L-1) + 1. \quad \textbf{(3)}$$

Note that the MODWT scaling coefficient equation allows for the calculation of $\widetilde{V_1}, \widetilde{V_2}, \ldots, \widetilde{V_{J-1}}$. Typically, however, only the j^{th} level MODWT scaling coefficients are retained since the remaining information in the lower levels scaling coefficients are included in the wavelet coefficients. This is a direct result of the MODWT preserving the energy of the original time series. See Percival & Walden (2000).

The scaling filter can be computed from a known wavelet filter by the equation (Percival & Walden, 2000):

$$g_l \equiv (-1)^{l+1} h_{L-1-t},$$

(4)

Different wavelet (scaling) filters have different shapes, allowing for the extraction of similar features from a given time series. Figure 2 shows an example of different wavelet and scaling filters from the Symlet family.

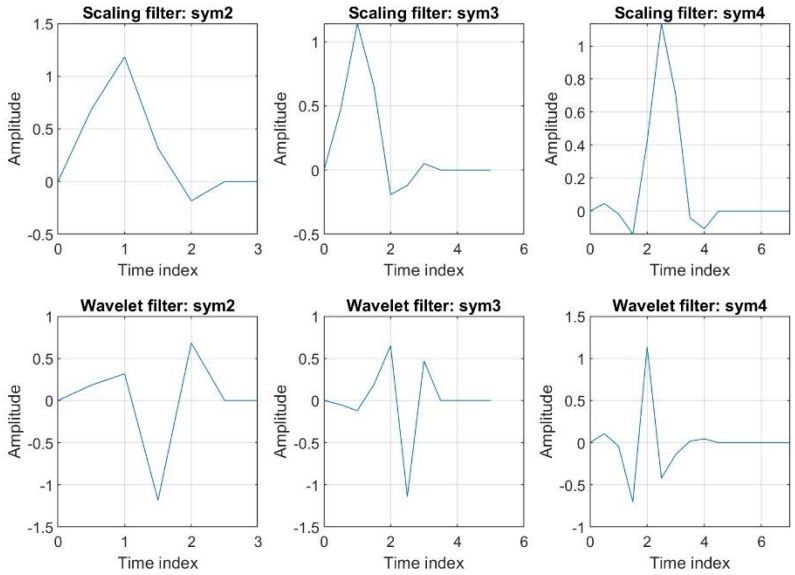


Figure 2. Some wavelet filters from the Symlet family

The intuition behind using the MODWT for streamflow forecasting is that it can uncover time-frequency information embedded in the explanatory variables (e.g., precipitation, air temperature) and convert this knowledge into new features with potentially predictive value. These engineered features may produce a better forecasting model (e.g., LSTM) than merely training the same model using the original features. The next sub-section shows how to identify and remove boundary coefficients that can introduce errors into WT-based forecasting models (Quilty & Adamowski, 2018).

3.2.1 Boundary Coefficients

The MODWT in its raw form suffers from look-ahead bias in the first L_j wavelet and scaling coefficients due to the circular convolution operation required to generate the MODWT wavelet and scaling coefficients (see Equations (1) and (2)). This look-ahead bias needs to be addressed before the wavelet and scaling coefficients can be used to train a model.

Circular convolution is a convolution operation where the original time series is considered periodic. For a finite-length time series, this entails that the time series repeats indefinitely, and the ends of the time series wrap around such that the observation that occurs after the final time index is the observation at the initial time index of the time series. An example of periodic extension of a time series is shown in Figure 3.

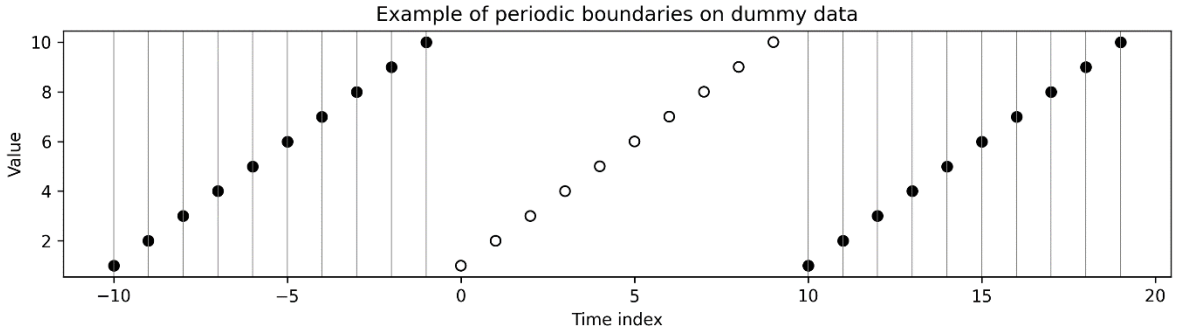


Figure 3 Example time series with 10 observations. Periodic extension is applied for time indices beyond the observed time index domain (0-9).

Circular convolution allows the convolution between a finite length time series and a wavelet or scaling filter in a way that is not restricted by the boundaries or "finiteness" of a given time series. The key result is that circular convolution allows the MODWT to generate wavelet and scaling coefficients that are the same length as the original time series. However, there is a cost to this advantage of MODWT. In particular, the first L_j values of the MODWT wavelet and scaling coefficients (for decomposition level j) depend on the last L_j values of the original time series. Since this requires that L_j observations from the future must be known to compute the first L_j values of the MODWT wavelet and scaling coefficients (for decomposition level j), look-ahead bias is present

when using the MODWT on a finite length time series to generate wavelet and scaling coefficients. However, this issue is easily addressed.

The simple solution is to remove the first L_j values of the wavelet and scaling coefficients (Quilty & Adamowski, 2018). This ensures that no "future data" is incorporated during model training.

Many studies in the literature have focused on building several forecast models using different wavelet filters and decomposition levels to identify the combination that results in the best-performing model (Dariane & Behbahani, 2024; Jahangir et al., 2023). In these cases, the issue of look-ahead bias due to the boundary coefficients can be resolved as follows:

1. Select a wavelet filter and maximum decomposition level J_{max}.
2. Decompose the features using the MODWT for the given wavelet filter and J_{max}.
3. Remove the first $L_{J_{max}}$ rows from the MODWT augmented dataset.
4. Split the dataset into training, validation, and test sets.

Since all MODWT coefficients and the target must have the same length for building a regression model, the maximum number of boundary-affected coefficients ($L_{J_{max}}$) determines how many observations need to be removed from the beginning of the MODWT wavelet, scaling coefficients, and target.

The removal of the $L_{J_{max}}$ records from the WT-engineered features and target is a sacrifice needed to avoid look-ahead bias. This drawback of the MODWT is a small sacrifice for low decomposition levels and wavelet filters of narrow lengths, especially for long time series. However, this level of "data loss" may render the MODWT infeasible for applications where available training data is scarce. In the next sub-section, details on LSTM's mathematical formulation is provided.

3.3 Long Short-Term Memory Networks

The main equations for LSTM and a description of its components are provided below. For a detailed review of LSTM, Yu et al. (2019) and Van Houdt et al. (2020) are recommended.

Consider an input sequence x_t of size d (i.e., the number of features) where the time index $t = 1, 2, \dots, T$, and T is the (input) sequence length. The LSTM updates its hidden state h_t and cell state c_t for each time step t by the following equations associated with each gate. Both h_t and c_t are vectors

of length M, where M is the number of hidden units in the LSTM layer. In what follows, the weights matrices (W) are of shape $M \times (d + M)$ and the bias vectors (b) are of length $d + M$ (the subscripts for W and b, related to each gate, have been suppressed for simplicity). The weights and biases are parameters that are learned by training the LSTM. Below, the main components of an LSTM cell are described in detail to provide intuition behind how the model processes features used as input (i.e., time series).

Forget Gate:

The forget gate determines the extent to which the previous cell state c_{t-1} is remembered (Hochreiter & Schmidhuber, 1997). This gate uses a sigmoid activation function that outputs values between 0 (completely forget) and 1 (completely remember), as shown in Equation (5):

$$f_t = \sigma\big(W_f \cdot [h_{t-1}, x_t] + b_f\big) \quad \textbf{(5)}$$

The variables in Equation (5) take on the following notation and meanings:

- f_t : Forget gate output at time t. This determines the extent to which the cell state c_{t-1} from the previous time step should be forgotten or retained. The forget gate output is a vector of values between 0 and 1, where 0 means "completely forget" and 1 means "completely retain."
- σ : Sigmoid activation function. The sigmoid function squashes its input to a range between 0 and 1, making it useful for converting its input to probabilities.
- W_f : Weight matrix for the forget gate.
- h_{t-1} : Hidden state from the previous time step $t - 1$. The hidden state contains information from the sequence up to that point.
- $[h_{t-1}, x_t]$: Concatenation of h_{t-1} and x_t. This combines information from both the previous hidden state and the current input.
- b_f : Bias term for the forget gate.

The resulting f_t is used to modulate the cell state c_{t-1}, deciding which information should be kept for future time steps.

Input Gate:

The input gate controls how much of the new computed information for the current step is stored in the cell state. It consists of a sigmoid layer and a tanh (hyperbolic tangent) layer. The sigmoid layer outputs the input gate values i_t and the tanh layer creates a new cell state candidate $\widetilde{c_t}$ through the following equations (Hochreiter & Schmidhuber, 1997):

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad \textbf{(6)}$$

$$\widetilde{c_t} = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \quad \textbf{(7)}$$

where:

- i_t : Input gate output at time t. This is a vector of values between 0 and 1, generated through the sigmoid activation function σ. The input gate decides which values in the cell state c_t should be updated.
- W_i : Weight matrix for the input gate.
- b_i : Bias term for the input gate.
- $\widetilde{c_t}$: Candidate cell state at time t. It's a vector that determines what the new cell state c_t should be.
- tanh: Hyperbolic tangent activation function. The tanh activation function maps its input to the between -1 and 1.
- W_c : Weight matrix for generating the candidate cell state.
- b_c : Bias term for the candidate cell state.

The input gate i_t and the candidate cell state $\widetilde{c_t}$ are used in conjunction with the forget gate f_t and output gate o_t (see below) to update the cell state c_t and hidden state h_t of the LSTM cell.

Cell State:

The cell state is updated by forgetting an estimated amount of information from the previous state and adding the new candidate state scaled by how much information to update in the new cell state (Hochreiter & Schmidhuber, 1997).

$$c_t = f_t \odot c_{t-1} + i_t \odot \widetilde{c_t} \quad \textbf{(8)}$$

The operations $+$ and $\odot$ are element-wise addition and product, respectively. Both operations can be summarized as follows:

- $\odot$: Element-wise product. When f_t multiplies c_{t-1} or i_t multiplies $\widetilde{c_t}$, the operation is done element-wise. This means that each element in f_t multiplies its corresponding element in c_{t-1}, and the same goes for i_t and $\widetilde{c_t}$.
- + : Element-wise addition. After the element-wise products are taken, the two resulting vectors are added together element-wise to produce c_t.

The variables in the cell state equation are defined as follows:

- c_t : Cell state at time t. The cell state serves as the "memory" of the LSTM cell, carrying information across time steps.
- f_t : Forget gate output at time t. This vector decides which parts of the previous cell state c_{t-1} should be kept or forgotten. Resulting values are between 0 and 1.
- c_{t-1} : Cell state from the previous time step $t-1$.
- i_t : Input gate output at time t. This vector decides which information from the candidate cell state $\widetilde{c_t}$ should be added to the cell state. Resulting values range between 0 and 1.
- $\widetilde{c_t}$: Candidate cell state at time t. It suggests what new information should be stored in the cell state c_t.

The cell state operates as follows:

1. $f_t * c_{t-1}$ is the portion of the cell state that decides to keep or "remember" information passed to it. The forget gate f_t scales the previous cell state c_{t-1}, effectively "forgetting" or "keeping" information.

2. $i_t * \widetilde{c_t}$ is the new candidate information to add to the cell state. The input gate i_t decides which parts of $\widetilde{c_t}$ are important to add.

The cell state c_t is a combination of the previous cell state c_{t-1} and the new candidate cell state $\widetilde{c_t}$, modulated by the forget gate f_t and input gate i_t, respectively.

Output Gate:

The output gate controls the amount of information from the current cell state c_t to send to the next hidden state. It has a sigmoid layer and applies the tanh activation function to the cell state; multiplying these two results gives the next hidden state h_t (Hochreiter & Schmidhuber, 1997):

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad \textbf{(9)}$$

$$h_t = o_t \odot \tanh(c_t) \quad \textbf{(10)}$$

The variables in the output gate equation are defined as follows:

- o_t : Output gate output at time t. This is a vector with values between 0 and 1 , generated through the sigmoid activation function σ. It controls which parts of the cell state c_t should be exposed to the next hidden state h_t and, potentially, to the output of the LSTM layer.
- W_o: Weight matrix for the output gate.
- b_o: Bias term for the output gate.
- The variables in the updated hidden state equation are defined as follows:
- h_t: Hidden state at time t.

In Equation (10), the output gate o_t is multiplied element-wise by $\tanh(c_t)$, which is the activated cell state c_t mapped between -1 and 1. This gives the updated hidden state h_t, which serves as the output of the LSTM cell at time t. Both c_t and h_t are then used in calculations for the next time step $t + 1$.

By regulating the flow of information in the model using Equations (5)-(10) above, the LSTM can learn to keep important information in the cell state and ignore irrelevant information in the features.

This property of LSTM allows it to learn long-term dependencies and avoid the exploding and vanishing gradient problem often encountered in the traditional RNN architecture (Razvan Pascanu et al., 2013).

After the LSTM layer has processed the features, the hidden state at time t, h_t, is passed to a regression layer to generate the streamflow forecast. This research uses separate models to forecast streamflow at times $t+1$ and $t+3$. Thus, there is only one output node in the regression layer. The parameters in the regression layer include regression weights, W_r (a vector of size M), and a bias, b_r (scalar). The regression layer parameters are trained simultaneously with the weights and biases of the LSTM layer. The regression layer equation can be written as (Hashemi et al., 2022):

$$\hat{y}_t = {W_r}^T h_t + b_r \quad \textbf{(11)}$$

When training LSTM models, the parameters can learn exactly how to map the features to the target on the training data. However, when the same model is used on the test data, it results in poor out-of-sample performance. This issue is known as overfitting and represents a model with poor generalization properties (Hastie et al., 2009). To address this issue, LSTMs often use a "dropout" mechanism during model training that can randomly drop (set to zero) the inputs, outputs, and/or recurrent connections according to a selected dropout rate (ranging between 0 and 1) to improve model generalization (Gal & Ghahramani, 2016). In other approaches, dropout is included only in the regression layer (Srivastava et al., 2014). Thus, this study considers dropout during model training to reduce overfitting, as described in (Kratzert et al., 2019) – see also Section 4.2.6.

To generate accurate forecasts using an ML model (e.g., LSTM), the model must be provided with features that contain useful information about the target variable. Often, only a subset of features account for most of the predictive information in a feature set. A feature selection algorithm is invaluable for extracting only the most relevant features for forecasting a target variable (e.g., streamflow). The following sub-section gives a general overview of useful feature selection concepts based on information theory, motivating the method used for feature selection in this research.

3.4 Feature Selection

3.4.1 Entropy, Mutual Information, and Conditional Mutual Information

The EA feature selection method used in all experiments was based on CMI. To estimate CMI, two quantities need to be estimated: (differential) entropy and mutual information, or MI.

MI is a quantity that indicates how closely related two variables are using their probability distributions (Hanchuan Peng et al., 2005). Specifically, given two continuous random variables, X and Y, the MI between these random variables quantifies the amount of information revealed about Y after observing X. Mathematically, MI is defined as follows (Cover & Thomas, 1991):

$$\mathrm{MI}(X,Y) = \int_{y\in Y}\int_{x\in X} p(x,y)\log\left[\frac{p(x,y)}{p(x)p(y)}\right]dxdy \qquad \textbf{(12)}$$

where $p()$ is a probability distribution and $\log$ represents the natural logarithm (meaning that MI is measured in nats). In the presence of a third variable, Z, the MI between X and Y conditional on Z can be estimated (Cover & Thomas, 1991):

$$\mathrm{CMI}(X,Y \mid Z) = \int_{z\in Z}\int_{y\in Y}\int_{x\in X} \log\left(\frac{p(z)p(x,y,z)}{p(x,z)p(y,z)}\right)p(x,y,z)\,dxdydz \qquad \textbf{(13)}$$

Both MI and CMI can be derived from differential entropy, an important concept in information theory (Cover & Thomas, 1991; Shannon, 1948). For d continuous random variables, their joint entropy is defined as (Cover & Thomas, 1991):

$$h(X_1, X_2, \dots, X_d) = -\int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty} p(x_1,\dots,x_d)\ln p\,(x_1,\dots,x_d)\,dx_1\dots dx_d \qquad \textbf{(14)}$$

MI between two random variables X and Y can be defined in terms of differential entropy as (Cover & Thomas, 1991):

$$\mathrm{MI}(X,Y) = h(X) + h(Y) - h(X,Y)$$

(15)

and CMI of X and Y, conditional on Z, can be written as (Cover & Thomas, 1991):

$$\mathrm{CMI}(X,Y \mid Z) = h(X,Z) + h(Y,Z) - h(Z) - h(X,Y,Z)$$

(16)

The computation of both MI and CMI requires estimating the differential entropy for one or more variables. This requires approximating probability density functions (PDF) for the relevant variables. There are multiple ways to approximate an unknown PDF using an observed sample. A common approach is to use non-parametric methods (Quilty et al., 2016). Two popular non-parametric methods for estimating PDFs are KDE (Sharma & Mehrotra, 2014)) and KNN (Frenzel & Pompe, 2007)). However, both KDE and KNN require the selection of hyper-parameters. KDE requires the specification of a kernel type and kernel parameters (e.g., width for the radial basis function kernel), while KNN requires the selection of the number of nearest neighbours. Identifying the optimal hyper-parameters for these methods is non-trivial and can require a substantial increase in computation time. Instead of KDE and KNN approaches, this research uses Edgeworth approximations to estimate PDFs since it does not require the specification of any hyper-parameters and is computationally efficient (Van Hulle, 2005).

Edgeworth approximation estimates the PDF of variable X by writing its characteristic function (i.e., the Fourier transform of the PDF) in terms of the characteristic function of a known PDF (e.g., normal distribution). Afterwards, the inverse Fourier transform is used to recover an approximation of the PDF. After estimating the PDF, it is used to estimate differential entropy via Equation (14). It is important to mention that Edgeworth approximations can estimate the PDF of X whether it is a vector or a matrix (univariate or multivariate), making it an ideal approach for estimating the entropy terms in CMI. Van Hulle (2005) provides additional details on the use of Edgeworth approximations for estimating differential entropy. Next, a general algorithm for applying CMI to feature selection is described.

3.4.2 Feature Selection Algorithm

This research uses the EA feature selection method (Quilty et al., 2016) to perform feature selection. In this method, both the CMI and MI are calculated using the Edgeworth approximation approach to estimating differential entropy (Van Hulle, 2005) via Equations (15) and (16) above.

An important facet of using the EA feature selection method is the use of a stopping criterion that halts the algorithm when it is determined that adding another feature (f) from $\boldsymbol{X}$ (e.g., MODWT engineered features) to the selected feature set ($\boldsymbol{S}$) is not expected to add sufficiently useful information about the target variable Y (e.g., streamflow). Earlier studies using the EA feature selection method considered the Hampel test criterion (Quilty et al., 2016) or the threshold approach (Sikorska-Senoner & Quilty, 2021; Jahangir et al., 2022). This research uses the threshold approach since it intuitively estimates the usefulness of adding a new feature to the selected feature set using information theoretic quantities.

The threshold approach involves measuring the CMI between the target variable, Y, and the already selected features, $\boldsymbol{S}$, conditioned on a new candidate feature, f, represented by $\mathrm{CMI}(f, Y|S)$. The ratio between $\mathrm{CMI}(f, Y|S)$ and the MI between the target variable and the selected feature set augmented by the new feature f, $[f, S]$, represented by, $\mathrm{MI}([f, S], Y)$, is taken. If the ratio does not exceed a threshold (τ, ranging between 0 and 1) or the ratio has increased over the past two iterations of the algorithm, it is deemed that the new feature does not add sufficient information about the target variable and the feature selection algorithm is stopped.

The EA method follows the typical workflow of a CMI-based feature selection method using the tolerance approach (e.g., Tsimpiris et al., 2012). The pseudocode for a generic CMI-based feature selection algorithm is presented below in Figure 4.

Algorithm 1 Feature Selection using Conditional Mutual Information with Threshold

Input: Data set D, Target variable Y, Threshold τ
Output: Selected feature subset S
Initialize an empty set $S = \emptyset$
while True **do**
    $max_cmi \leftarrow -\infty$
    $best_feature \leftarrow null$
    for each feature f in D not in S **do**
        $cmi \leftarrow$ compute CMI$(f, Y|S)$ ▷ Compute conditional mutual information of f and Y given features in S
        if $cmi > max_cmi$ **then**
            $max_cmi \leftarrow cmi$
            $best_feature \leftarrow f$
        end if
    end for
    $mi \leftarrow$ compute MI$(best_feature, Y)$ ▷ Compute mutual information of $best_feature$ and Y
    $ratio \leftarrow \frac{max_cmi}{mi}$
    if $ratio < \tau$ **then**
        break ▷ Break the loop if the ratio is less than the threshold
    end if
    Add $best_feature$ to S
end while
return S

Figure 4 Pseudocode for the adopted feature selection algorithm

The next chapter provides details on how WTs, EA-based feature selection, and LSTM methods are developed, tested, and applied to the CAMELS dataset for streamflow forecasting at horizons of one and three day(s) ahead .

Chapter 4 Experimental Setup

This chapter describes the different LSTM configurations used to determine the extent to which WTs improve streamflow forecasting performance on the CAMELS dataset. Details about the selection of WT parameters, feature selection, and dataset partitioning are also included, as are considerations regarding the training time of the different models using high-performance computing.

4.1 LSTM Configurations

Three LSTM configurations were considered in this thesis to evaluate the impact of WTs on streamflow forecasting performance. Since the overarching goal of this thesis is to compare the impact of WTs on the performance of streamflow forecasting models through a large-sample study, a baseline model is needed for comparison. Ideally, the baseline model should be the same as the WT-based model in every way (same number of hidden units, sequence length, etc.), except it should not use WTs for feature engineering. However, LSTMs have many hyper-parameters (e.g., number of hidden units, dropout rate, sequence length) whose selection can greatly impact the model's performance (Greff et al., 2017). Thus, a configuration that considered different values for the main LSTM hyper-parameters (number of hidden units, dropout rate, sequence length) was used as another benchmark. This configuration was used to check how performance improvements due to WTs compare against hyper-parameter selection. Thus, the results of this thesis can help a modeler who must focus on one task, or the other (feature engineering using WTs or hyper-parameter selection) make a more informed decision about where to spend their time.

In summary, the LSTM configurations explored in this thesis include:

1. Baseline LSTM (B-LSTM)
2. Wavelet LSTM (W-LSTM)
3. Grid Search LSTM (GS-LSTM)

Section 4.1.1 describes the architecture of B-LSTM. As noted above, B-LSTM was used as a benchmark for W-LSTM. Both B-LSTM and W-LSTM used the same LSTM architecture (number of hidden units, sequence length), with the difference being that W-LSTM used the MODWT for feature engineering (see Section 4.1.2).

GS-LSTM is the best-performing LSTM architecture from a set of pre-defined architectures that vary according to the number of hidden units, (regression layer) dropout rate, and sequence length. For GS-LSTM, the best-performing architecture depends on the training dataset. Thus, it is expected that the best architecture will be catchment-specific.

The purpose of GS-LSTM is to provide another benchmark for W-LSTM that enables the impact of WT-generated features on streamflow forecasting performance to be assessed. Hyper-parameter selection serves as a relevant benchmark for WT-based feature engineering as it is a widely accepted approach for improving out-of-sample (test set) performance (Pham et al., 2018).

4.1.1 Baseline LSTM

The baseline LSTM architecture is based on Kratzert et al. (2019). The authors trained a single LSTM model, with a single LSTM layer and regression layer, to simulate streamflow using data from 531 catchments included in the CAMELS dataset. One of the main objectives of their study was to learn universal streamflow characteristics (and encode this knowledge in the LSTM parameters), such that a single model could be used for streamflow simulation across catchments with different hydroclimatic conditions. The LSTM architecture from their study was selected based on k-fold cross-validation with $k = 4$ (Hastie et al., 2009). The architecture from their study is as follows:

1. **Number of hidden units**: 256
2. **Dropout rate (regression layer only)**: 0.4
3. **Sequence length**: 270

This architecture was selected for the baseline model in this research since Kratzert et al. (2019) demonstrated the ability of this architecture to learn how to produce accurate streamflow simulations from the same set of hydroclimatically diverse catchments adopted in this study (i.e., the CAMELS dataset). However, there are two limitations to selecting this baseline architecture in this research. First, this architecture was proposed for streamflow simulation (not forecasting). Second, the authors developed and trained a model for all catchments and selected the architecture based on its performance in this setting. In other words, the authors developed a single model that performed well across all catchments, while in this research, models were trained for each catchment separately.

These two limitations are not major barriers to selecting the abovementioned baseline architecture for this work. Firstly, to the author's knowledge, a benchmark LSTM model for one and

three day(s) ahead streamflow forecasting on the CAMELS dataset does not exist. Secondly, running a large-scale experiment to find an optimal baseline for each catchment and forecast horizon would be extremely laborious since it would entail developing and evaluating the performance of many possible architectures. Instead, starting from a well-established benchmark for streamflow simulation (in the absence of a benchmark for streamflow forecasting), such as the architecture in Kratzert et al. (2019), seemed reasonable.

Additionally, since this research is interested in evaluating multiple wavelet filters for each catchment (see Section 4.1.2), it is difficult to build a single model for the entire CAMELS dataset. This is the case since different wavelet filters are expected to be better suited to different catchments (Maheswaran & Khosa, 2012; Rathinasamy et al., 2013). Thus, building a single model for the CAMELS dataset using a single wavelet filter is not advised. However, as a result of training separate models according to different wavelet filters on each catchment (as done in this thesis), it can be determined which wavelet filter works "best" for each catchment. Afterwards, this knowledge can be used to engineer features according to the best wavelet filter for each catchment and combined into a single dataset (i.e., across all catchments in the CAMELS dataset) to train a single model. This research provides the basis for building such a model.

4.1.2 Wavelet LSTM

As indicated above, the architectures of W-LSTM and B-LSTM were selected to be identical. The only difference between the two models is that the W-LSTM includes additional features engineered via the MODWT.

The MODWT requires two parameters to be specified (see Section 3.2); specifically, the wavelet filter $\tilde{h}$ and maximum decomposition level J. Recall that the wavelet filter determines the shape of the convolution windows that are convolved with a given feature (e.g., precipitation) to generate the MODWT wavelet and scaling coefficients. The MODWT wavelet and scaling coefficients represent the MODWT-engineered features associated with a given feature in $\boldsymbol{X}$. The maximum decomposition level determines the number of MMODWT-engineered features that are created based on a given feature (see Section 2.3.1.1.1). Since there are multiple features in the CAMELS dataset (precipitation, maximum and minimum temperature, etc.), the number of MMODWT-engineered features for a dataset containing d features is $d(J+1)$. The CAMELS dataset includes seven features for each catchment.

Therefore, an additional $7(J+1)$ features are generated by using the MODWT for feature engineering on this dataset.

To determine the wavelet filter and maximum decomposition level, referred to as the tuple $(\tilde{h}, J)$, grid search was applied over a grid of candidate wavelet filters and maximum decomposition levels. In other words, an LSTM model was developed for each tuple $(\tilde{h}, J)$ and the tuple with the best performance on the validation set (see Section 4.2.1) was associated with the optimal MODWT parameters. The validation set is typically used for hyper-parameter selection (Hastie et al., 2009). Since searching for optimal MODWT parameters can be seen as a form of hyper-parameter selection, it is intuitive to use the validation set for selecting the best tuple $(\tilde{h}, J)$. The same approach to identifying the optimal MODWT parameters was adopted by Jahangir et al. (2023).

The *pywddf* Python package was used to engineer features using the MODWT (You, 2023). This package has built-in functionality to remove boundary-affected coefficients. The following two sub-sections discuss the selection of the candidate wavelet filters and maximum decomposition level.

4.1.2.1 Selection of Candidate Wavelet Filters

In this thesis, 33 wavelet filters were selected as candidates for use with MODWT, while the maximum decomposition level was fixed at six ($J = 6$). This resulted in 33 W-LSTM models trained for each catchment and forecast horizon (one and three day(s) ahead).

The set of 33 wavelet filters were selected from two sources: Percival and Walden (2000) and MATLAB's Wavelet Toolbox (*Wavelet Toolbox*, 2022). Only the "orthogonal" class of wavelet filters were used, as these are commonly used wavelet filters for the MODWT (Percival & Walden, 2000). The 33 candidate filters are listed in Table 1. The wavelet filters were also selected according to their length (see the sub-section below) to avoid producing a large number of boundary coefficients (see Equation (3)). Since the boundary coefficients need to be removed from the MODWT wavelet and scaling coefficients, wavelet filters with lengths up to 14 were considered to keep the number of removed boundary coefficients at a reasonable level (Quilty & Adamowski, 2018).

Table 1. Candidate wavelet filters used in experiments

Wavelet Family Name	Wavelet Family Short Name	Wavelet Filters Used in Experiments
Daubechies	"db"	db1, db2, db3, db4, db5, db6, db7
Symlets	"sym"	sym4, sym5, sym6, sym7
Coiflets	"coif"	coif1, coif2
Least Asymmetric	"la"	la8, la10, la12, la14
Fejér-Korovkin	"fk"	fk4, fk6, fk8, fk14
Best-localized Daubechies	"bl"	bl7
Morris minimum-bandwidth	"mb"	mb4_2, mb8_2, mb8_3, mb8_4, mb10_3, mb12_3, mb14_3
Han linear-phase moments	"han"	han2_3, han3_3, han4_5, han5_5

4.1.2.2 Selection of Maximum Decomposition Level

The maximum decomposition level was fixed to a value of $J = 6$. A notational distinction is made between the maximum decomposition level and a particular decomposition level by J and j, respectively.

The value $J = 6$ was selected by considering the interpretation of decomposition levels in relation to time scale. When a time series is decomposed using the MODWT, the level j wavelet coefficients represent changes in the time series on time scale of 2^{j-1} times the time step of the time series data. The scaling coefficients at level j represent averages on time scale 2^j (Percival and Walden, 2000).

In the context of the daily hydrometeorological observations included in the CAMELS dataset, the MODWT wavelet coefficients at decomposition level j results in high-frequency changes in a time series over the time scale 2^{j-1} days, while the scaling coefficients result in a smoothed or averaged version of the time series. The MODWT wavelet coefficients represent fluctuations or variations occurring roughly every one day at $j = 1$, two days at $j = 2$, four days at $j = 3$, etc. Furthermore, the MODWT wavelet and scaling coefficients at level j are associated with frequencies in the interval $[\frac{1}{2^{j+1}}, \frac{1}{2^j}]$ and $[0, \frac{1}{2^{j+1}}]$, respectively. This relationship is useful for identifying frequencies of interest and selecting an appropriate decomposition level.

This thesis adopted a decomposition level so that the final set of MODWT wavelet coefficients included the seasonal periodicity of the hydrometeorological time series in the CAMELS dataset. Capturing the seasonality of hydrometeorological features (precipitation, evapotranspiration, etc.) is useful for explaining streamflow seasonality, an important facet of water resources management (Gnann et al., 2020). Thus, a decomposition level of six ($J = 6$) was selected, which captures periodicities in the range of 64 to 128 days, encompassing the seasonal periodicity. This results in the MODWT scaling coefficients capturing periodicities at 128 days and beyond. Using a maximum decomposition level of six means that a total of seven MODWT-generated features were generated per meteorological feature. The seven MODWT-generated features correspond to a set of six wavelet coefficients (associated with decomposition levels one through six) in addition to scaling coefficients at a decomposition level of six.

In addition to the MODWT-generated features, the W-LSTM models also used the original features. This approach has been used with success in previous studies (Jahangir et al., 2023; Quilty & Adamowski, 2018, 2021; Rahman et al., 2020). Overall, the W-LSTM considered 56 features for each catchment and forecast horizon (one and three day(s) ahead). This includes the seven original features ($d = 7$) in addition to the $d(J + 1)$ MODWT generated features (see Section 2.3.1.1.1).

4.1.3 Grid Search LSTM

GS-LSTM involved training 32 different LSTM architectures to find the best-performing one in terms of validation set performance. The number 32 was determined so that the number of models trained was as close as possible to 33—the total number of models trained for W-LSTM per catchment and forecast horizon (one and three day(s) ahead).

All possible combinations of the following sets of hyper-parameters resulted in the 32 architectures for GS-LSTM:

1. **Number of hidden units**: 32, 64, 128, 256
2. **Dropout rate (regression layer only)**: 0, 0.5
3. **Sequence length**: 30, 60, 120, 270

As a dropout of 0.5 is typically used in LSTM models (Srivastava et al., 2014), the candidate values of 0 and 0.5 were used to compare no dropout with the presence of dropout. The values considered for the number of hidden units was chosen by starting with the number of hidden units used in the baseline architecture and moving downwards by powers of two. The lower limit of the number of hidden units was set to 32 to avoid considering models with too few hidden units. A similar number of hidden units were explored in other studies that used LSTM for hydrological applications (e.g., Kratzert et al., 2019; Hashemi et al., 2022). Similar to the approach taken for identifying candidates for the number of hidden units, the candidates considered for the sequence length were selected by working backwards from the sequence length used in the baseline LSTM architecture, down to a minimum lookback period of 30 days (1 month).

4.2 Forecast Pipeline and Settings

In this sub-section, the forecast pipeline (workflow) for the three LSTM configurations (B-LSTM, W-LSTM, and GS-LSTM) is presented. Steps common to all three configurations (e.g., dataset

partitioning, feature selection) are described first, followed by settings specific to each configuration. Figure 5 provides an overview of the main steps in the forecast pipeline, considering all three configurations.

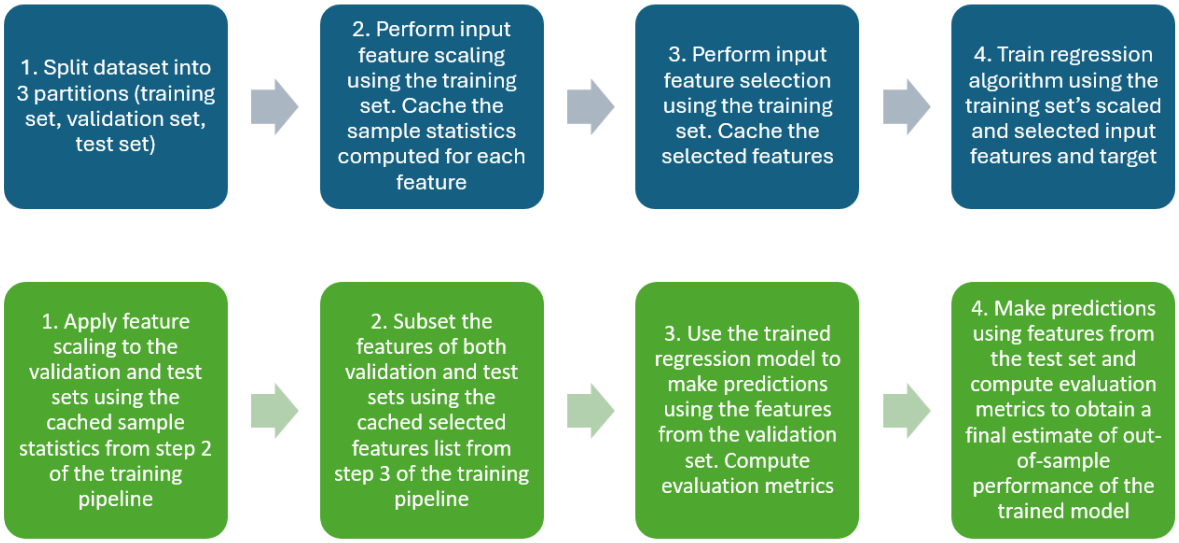


Figure 5 Training (navy blue) and validation and testing (green) pipelines common to all three configurations.

4.2.1 Dataset Partitioning

The data for each catchment were split into training, validation, and test sets containing 70, 15, and 15 percent of the data, respectively. The training set was used to tune the model parameters according to the mean squared error loss function (see Section 4.2.6). The validation set was used to identify the best hyper-parameters in the case of W-LSTM and GS-LSTM (this step was skipped for B-LSTM since its architecture was fixed). The test set was used to evaluate the out-of-sample performance of all models. The training, validation, and test sets were in chronological order and sorted from the oldest to the newest records.

While it is very difficult to identify an optimal dataset partitioning ratio, which could depend on the complexity of the model (Hastie et al., 2009), the ratio used in this research is common for ML models (Goodfellow et al., 2016) and hydrological applications (see, for example, Barzegar et al., 2019).

4.2.2 Handling Boundary Coefficients

Following the recommendations in Quilty & Adamowski (2018), the first L_J records in the MODWT wavelet and scaling coefficients were removed to avoid look-ahead bias (see Section 3.2). Thus, the first L_J records in the MODWT wavelet and scaling coefficients, the original features, and the target were removed prior to their use in the W-LSTM models. To ensure the B-LSTM and GS-LSTM models used the same number of training records as W-LSTM, the first L_J measurements were also removed from the original set of features and target used in these models. It can be argued that this data loss in the B-LSTM and GS-LSTM models is unwarranted. However, this thesis aims to compare any improvements in forecasting performance due to WTs. Thus, the effect of the WTs on model performance should be isolated as much as possible from other factors in the development of the different configurations. By using the same number of training records in each configuration, it is easier to achieve this goal.

Since different wavelet filters of different lengths were explored (see Section 4.1.2.1), L_J was calculated using the longest wavelet filter, which was 14. Since a decomposition level of six was used in all W-LSTM models, $L_J = 820$ (see Equation 3). Thus, 820 records were removed from the beginning of the training set in the B-LSTM, W-LSTM, and GS-LSTM models. It is possible that this loss of training records could have a non-negligible impact on the performance of B-LSTM and GS-LSTM. While it is outside the scope of this thesis, this impact could be assessed using Bayesian models, such as the recently proposed incremental Bayesian extreme learning machine (Q. Ren et al., 2023). The following sub-section briefly overviews the software and the settings used to train the different LSTM configurations.

4.2.3 Feature Normalization

After splitting the data into different partitions (Section 4.2.1) and removing boundary coefficients (Section 4.2.2), all features (excluding the target) were normalized using z-score normalization based on the training data. In other words, each feature's mean and standard deviation in the training set were used to scale the data across all partitions. Feature normalization is a common technique used to accelerate the training of neural networks, such as LSTM (Ioffe & Szegedy, 2015).

4.2.4 Feature Selection

The EA-based feature selection method was used to select features for B-LSTM, W-LSTM, and GS-LSTM (individually) for each catchment and forecast horizon (one and three day(s) ahead). However, since B-LSTM and GS-LSTM were developed using the same data (i.e., without WT-based feature engineering), the EA-based feature selection method was run only once for the original feature set, and the selected features were used in both models. For W-LSTM, 33 different wavelet filters were used to engineer features for each catchment and forecast horizon individually. Afterwards, feature selection was performed for the 33 datasets (see Section 4.2.4). Following earlier studies (Jahangir et al., 2023; Quilty & Adamowski, 2020), feature selection was done using the training data.

The EA-based feature selection method requires the specification of the tolerance value, τ, which influences the number of selected features. Higher tolerance values lead to fewer features being selected. Based on earlier studies (Sikorska-Senoner & Quilty, 2021; Jahangir et al., 2023), $\tau = 0.05$ was selected since this tolerance value was found to provide a good balance between model performance and the computational efficiency of the feature selection process. The hydroIVS R package was used for performing feature selection using the EA-based feature selection method (You, 2022).

4.2.5 Forecast Evaluation Metrics

Many evaluation metrics can be used for evaluating streamflow forecasts (Gauch et al., 2023). However, one of the most common metrics adopted for evaluating the performance of streamflow forecasting models is the Nash-Sutcliffe efficiency index or NSE (Knoben et al., 2019; Nash & Sutcliffe, 1970). The NSE can be calculated as follows (Moriasi et al., 2007):

$$NSE = 1 - \frac{\sum_{t=1}^{N}(Q_t - \hat{Q}_t)^2}{\sum_{t=1}^{N}(Q_t - \bar{Q})^2} \quad \textbf{(17)}$$

where Q_t is the observed streamflow at time t, $\hat{Q}_t$ is the forecasted streamflow at time t, and $\bar{Q}$ is the mean of the observed streamflow.

The NSE is a unitless metric that ranges from $-\infty$ to 1. A forecast with an NSE of 0 represents a forecast that is as useful as the mean of the target, while a forecast with an NSE of 1 exactly matches the target. Moriasi et al. (2007) provide different ranges for the NSE associated with different

qualitative rankings of the model's performance. For example, according to the authors, NSE within the range of 0.5 to 0.6 may indicate a satisfactory forecasting model. However, others have used an NSE of 0 to distinguish between good and poor models (for example, Knoben et al., 2019 and references therein).

Due to the ubiquity of NSE in hydrology, this metric is adopted to evaluate any improvements in streamflow forecasting performance due to WTs. Using a single metric is not expected to encapsulate the overall improvement in performance that may be realized by using WTs in a streamflow forecasting model. However, it provides an initial WT-based benchmark on the CAMELS dataset that new forecasting models can be evaluated against. Furthermore, earlier studies that used the CAMELS dataset to test the performance of new models also used NSE as the primary evaluation metric (e.g., Kratzert et al., 2019; Gauch et al., 2023).

4.2.6 LSTM Training Settings

LSTM models are highly customizable, with dozens of tunable hyper-parameters outside of those explored in GS-LSTM (number of hidden units, dropout rate, and sequence length). However, the main goal of this thesis is to compare, via a large-sample study, any improvements in streamflow forecasting performance due to the use of WTs for feature engineering by using a simple LSTM benchmark (i.e., B-LSTM). The choice of benchmark architecture, as described in Section 4.1.1, was intentionally kept simple. Thus, typical settings were used for the remaining LSTM hyper-parameters. If the goal is to follow a thorough hyper-parameter selection strategy, then hyper-parameter selection methods, such as Bayesian optimization (Frazier, 2018) or random search (Greff et al., 2017), can be used.

The Keras Python library (Chollet, 2015) was used to train, validate, and test the LSTM configurations (B-LSTM, W-LSTM, and GS-LSTM). Settings for some of the more prominent LSTM hyper-parameters are provided below for reference:

- **Maximum Epochs**: 100
- **Early Stopping**: True
- **Early Stopping Patience**: 10 epochs
- **Batch Size**: 32

- **Loss function**: Mean squared error
- **Optimizer**: Adam (Kingma & Ba, 2017)

All other unspecified hyper-parameters, such as the learning rate, were kept at the default values in Keras. Details on the different hyper-parameters listed above are not provided to maintain brevity and can instead be found elsewhere (Montavon et al., 2012). The next sub-section includes details on the forecast pipelines for the different LSTM configurations.

4.2.7 LSTM Configurations

4.2.7.1 Baseline LSTM

The B-LSTM followed the forecast pipeline listed below and was used for each catchment and forecast horizon (one and three day(s) ahead):

1. Split data for a given catchment into train, validation, and test sets (see Section 4.2.1).
2. Remove boundary coefficients from the training set (see Section 4.2.2).
3. Perform z-score normalization across all partitions based on the mean and standard deviation of the training set features (Section 4.2.3).
4. Use EA-based feature selection (see Section 4.2.4) to select features using the training set. Keep only the selected features.
5. Using the training set, train an LSTM model based on the features determined by feature selection.
6. Using the trained LSTM model, generate forecasts across all partitions and compute the NSE. Save the NSE, forecasts, and the trained model weights and biases.

The W-LSTM pipeline followed the same general forecast pipeline as B-LSTM but included feature engineering as the primary step instead.

4.2.7.2 Wavelet LSTM

For W-LSTM, the forecast pipeline followed these steps:

1. For each wavelet filter, $\tilde{h}$ (see Table 1), and a decomposition level of six ($J = 6$), engineer new features by applying the MODWT to each feature. Concatenate the original features ($d = 7$)

with the MODWT features (seven features per original feature; see Section 4.1.2.2), resulting in 56 features.

2. Split data for a given catchment into training, validation, and test sets (Section 4.2.1).
3. Remove boundary coefficients from the training set (Section 4.2.2).
4. Perform z-score normalization across all partitions based on the mean and standard deviation of the training set features (Section 4.2.3).
5. Perform feature selection on the training set using EA-based feature selection (see Section 4.2.4). Keep only the selected features.
6. Using the training set, train an LSTM model based on the features determined by feature selection.
7. Using the trained LSTM model, generate forecasts across all partitions and compute the NSE. Save the NSE, forecasts, and the trained model weights and biases.

The GS-LSTM and B-LSTM forecast pipelines were similar, except that GS-LSTM considered different hyper-parameter settings for the number of hidden units, the regression layer dropout rate, and the sequence length.

4.2.7.3 Grid Search LSTM

The forecast pipeline for GS-LSTM followed this process for each catchment and forecast horizon (one and three day(s) ahead):

1. Perform all steps in the B-LSTM pipeline, presented in Section 4.2.7.
2. Repeat the above step for all 32 hyper-parameter sets presented in Section 4.1.3.
3. Using the training set, train an LSTM model based on the features determined by feature selection (i.e., the same features as B-LSTM).
4. Using the trained LSTM model, generate forecasts across all partitions and compute the NSE. Save the NSE, forecasts, and the trained model weights and biases.
5. Select the hyper-parameter set that yielded the best validation set performance according to the NSE. This architecture was selected as the “best” GS-LSTM (for the given catchment and forecast horizon).

In what follows, details about the high-performance computing used during the experiments, along with the time taken to train the different model configurations.

4.3 High-Performance Computing and Model Training Times

Hardware for training the different model configurations (B-LSTM, W-LSTM, and GS-LSTM) consisted of the Cedar supercomputing cluster at Simon Fraser University in Burnaby, British Columbia (Canada). Nvidia P100 GPUs with 12GB of memory were used for model training. Multiple GPUs were used to reduce training time - one GPU per catchment.

Table 2 summarizes the total number of models trained during the experiments involved in this thesis.

Table 2. Training times for each LSTM configuration

Configuration	Number of Forecast Horizons	Number of Different Parameter/Hyper-parameter sets	Number of Catchments	Total Number of Models	Total Training Time (GPU Days)
Baseline LSTM	2	1	620	1240	2.81
Wavelet LSTM	2	33	620	40,920	191.38
Grid Search LSTM	2	32	620	39,680	92.48
Total				81,840	286.68

Overall, 81,840 LSTM models were trained, taking a total of ~287 GPU days. By parallelizing the computations using multiple GPUs, training time was reduced to about a week (wall clock time). The distribution of LSTM training times across 620 catchments, broken down by forecast horizon and experiment type (B-LSTM, W-LSTM and GS-LSTM), are presented in Figure 6, Figure 7 and Figure 8 below.

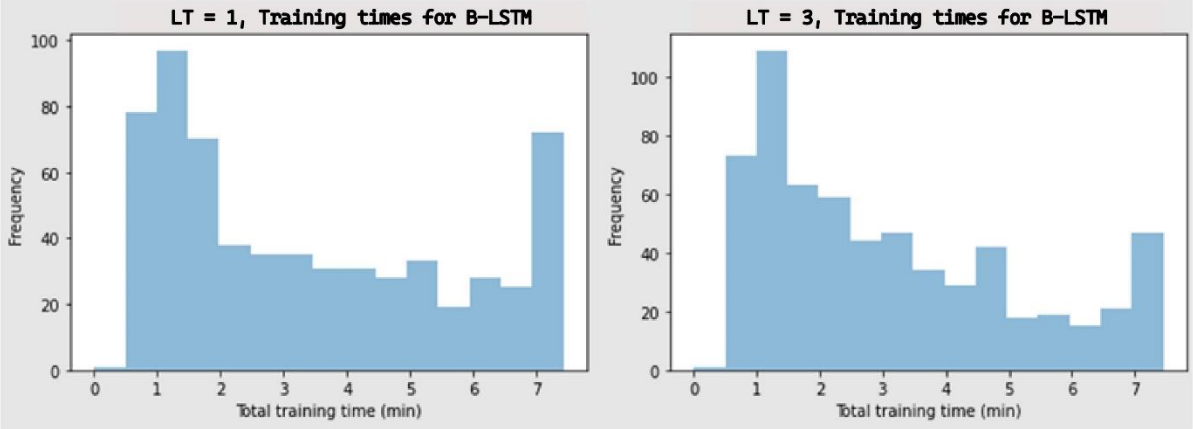


Figure 6 Histogram of training times for B-LSTM (one and three day(s) ahead forecast horizons/lead time (LT))

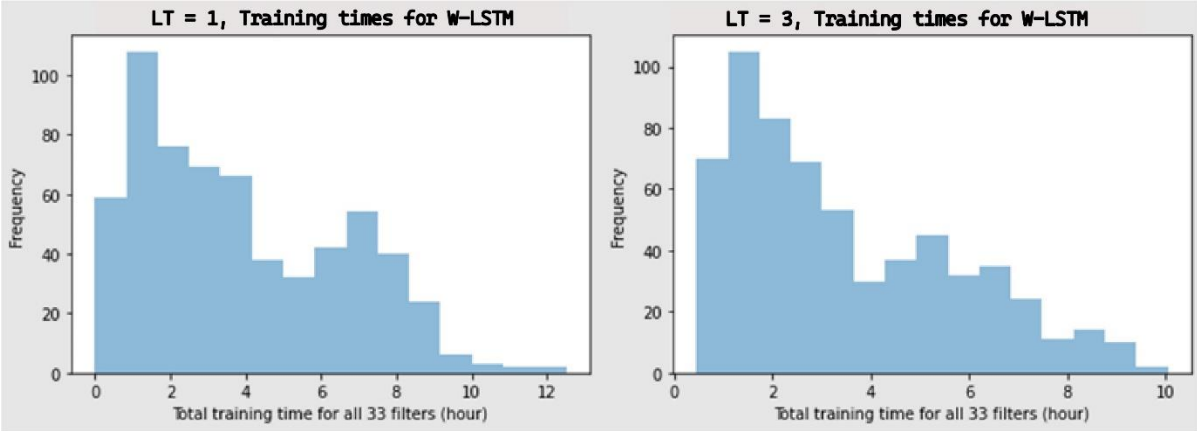


Figure 7 Histogram of training times for W-LSTM (one and three day(s) ahead forecast horizons/lead time (LT))

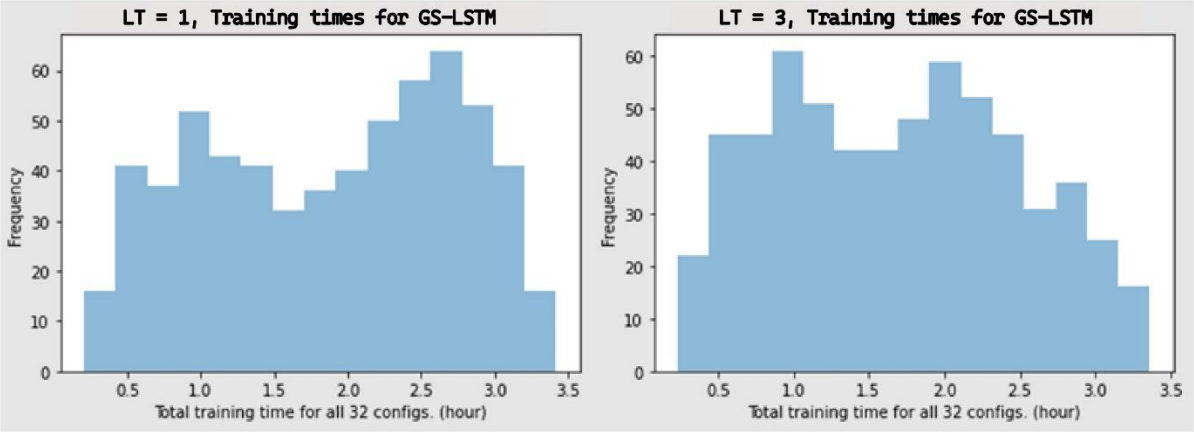


Figure 8 Histogram of training times for GS-LSTM (one and three day(s) ahead forecast horizons/lead time (LT))

The distribution of total time per catchment spent on selecting features is shown in Figure 9 for each forecast horizon. Note that the times shown in Figure 9 include time spent engineering features via the MODWT. However, the time taken to perform MODWT-based feature engineering is negligible compared to the time taken by feature selection.

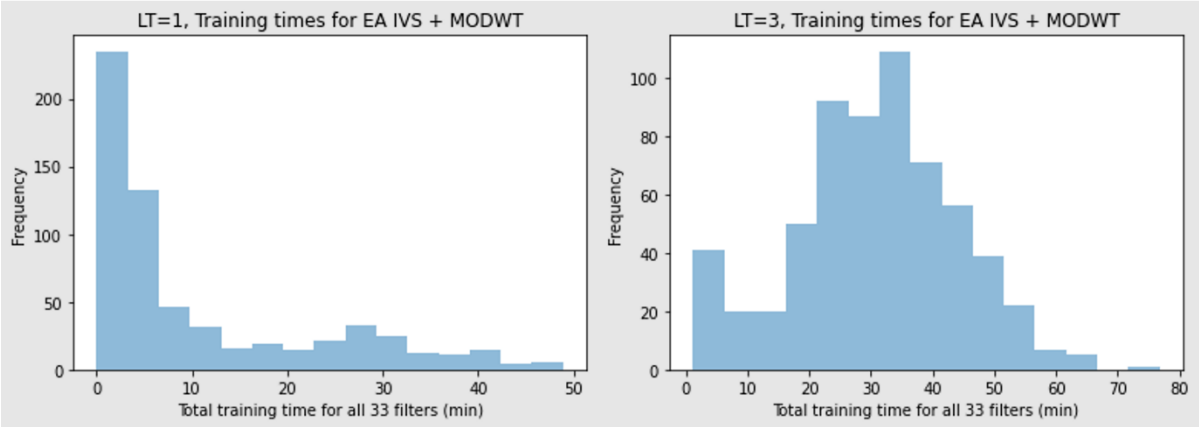


Figure 9. Training times for feature selection and MODWT for each forecast horizon/lead time (LT)

For the W-LSTM configuration, the combined computation time for performing both MODWT-based feature engineering and feature selection took 429.14 CPU hours or 17.88 CPU days.

4.4 Filtering Catchments for Detailed Analyses

Of the 620 CAMELS catchments considered in this thesis, only a subset of catchments were considered for further analysis based on the test set NSE achieved by the B-LSTM configuration. The rationale behind this decision follows. Although the improvement due to WT-generated features for a particular catchment might be substantial, if the performance of the B-LSTM model for the catchment is low, then even a large percent increase in model performance due to WT features (e.g., a 50% increase in test set NSE) would still result in a low performing model. In such cases, even the better-performing model would not be helpful for real-world applications (e.g., informing operations of a multi-purpose reservoir).

Hence, the 620 CAMELS catchments were filtered according to a simple criterion. For a given forecast horizon (one or three day(s) ahead), all catchments whose B-LSTM model had a test set NSE exceeding 0.4 were filtered. The filtering threshold value of 0.4 was motivated by a recommendation provided by Moriasi et al. (2007), where the authors suggested that a model with an NSE above 0.5 may be considered "satisfactory". The recommended NSE threshold of 0.5 was lowered to 0.4 in this thesis for two reasons. First, this thesis considers streamflow forecasting, whereas Moriasi et al. (2007) focused on simulating streamflow (and transport of sediment and nutrients). Forecasting is a considerably more challenging task than simulation (Girihagama et al., 2022; Tongal & Booij, 2018); hence, reducing the NSE threshold for filtering catchments from 0.5 to 0.4 seemed reasonable. Second, using 0.4 as the test NSE threshold to filter catchments resulted in a larger set of catchments post-filtering, enabling improved statistical inferences (i.e., due to a larger sample size) than if the threshold was kept at 0.5.

The following two sub-sections describe the results of the filtering process described above for both forecast horizons. Appendix B includes a list of station names associated with the filtered catchments.

4.4.1 One Day Ahead Forecasting

For the one day ahead forecast horizon, 90 catchments remained after filtering the original 620 catchments according to a test set NSE exceeding 0.4 for the B-LSTM. The 90 catchments identified by the filtering process are overlaid on a map (Figure 10) of the CONUS below.

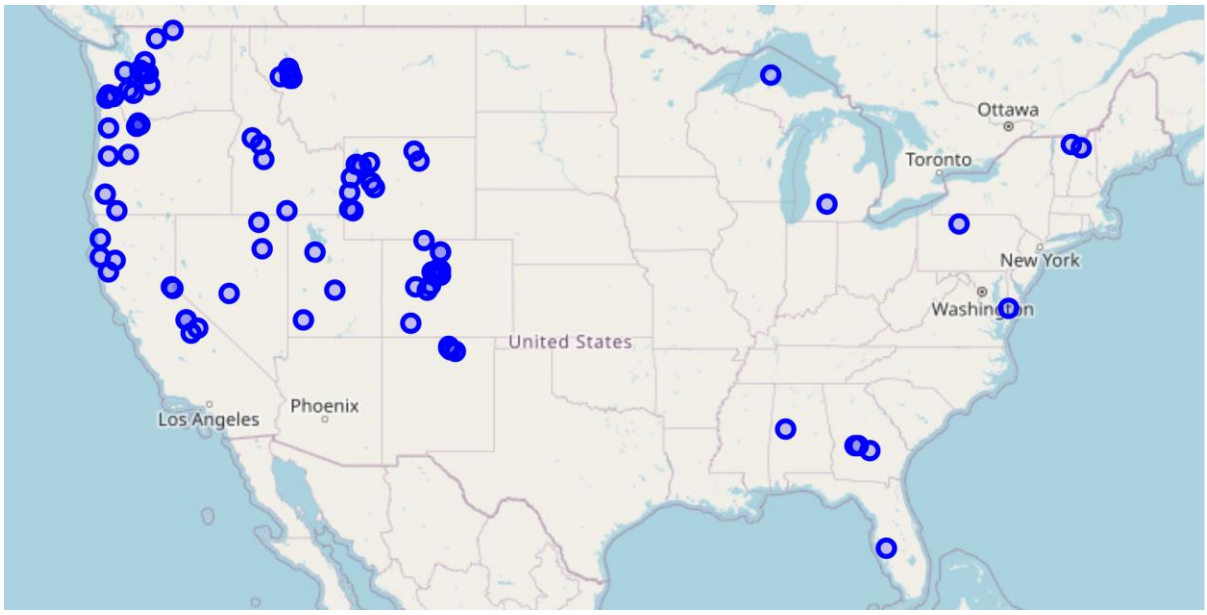


Figure 10. Catchments where B-LSTM achieved a test set NSE > 0.4 for a forecast horizon of one day

There appears to be a westward bias captured by the filtering process. Catchments located in the western part of the United States appear to be the best candidates for LSTM-based streamflow forecasting. Basins located in the Pacific Northwest are generally dominated by high mean annual precipitation and strong runoff seasonality (Hamlet & Lettenmaier, 1999). The Northwest region of the CONUS may be considered an example of high streamflow predictability because of the strong correlation between meteorological forcings and streamflow.

4.4.2 Three Days Ahead Forecasting

For the three day(s) ahead forecast horizon, 60 catchments remained after filtering the original 620 catchments. Figure 11 shows a similar pattern to what was observed for the one day ahead forecasting horizon. The western US seems most compatible with LSTM-based streamflow forecasting at the daily scale.



Figure 11. Catchments where B-LSTM achieved a test set NSE > 0.4 for a forecast horizon of three days

Chapter 5 Results and Discussions

This chapter presents the results of the experiments discussed in Chapter 4. Section 5.1 presents the impact of using WT-generated features on the performance of the B-LSTM across all 620 catchments from the CAMELS dataset. Kernel density estimation is used as the primary tool to compare the distribution of forecasting performance across all catchments for each model configuration considered in this thesis (B-LSTM, W-LSTM and GS-LSTM). Section 5.2 presents an assessment of the best-performing wavelet filters. In Section 5.3, W-LSTM is compared to GS-LSTM, and Section 5.4 presents a brief analysis of catchments where W-LSTM showed the greatest improvement in forecasting performance over B-LSTM. Section 5.5 summarizes the main results of the thesis, and Section 5.6 includes hydrographs for several catchments to demonstrate the difference between forecasts generated by W-LSTM and B-LSTM as a function of time.

Unless otherwise noted, all results in this chapter pertain to the test set. Thus, the results indicate the out-of-sample performance for all models explored in this thesis.

5.1 Impact of Different Wavelet Filters on Forecast Performance

This sub-section addresses this thesis's first objective by assessing how different wavelet filters impact streamflow forecast performance across the CAMELS dataset compared to B-LSTM.

After filtering all catchments to isolate those where B-LSTM achieved a test set NSE greater than 0.4, 90 catchments remained from the original 620 CAMELS catchments for the forecast horizon of 1 day, and 60 catchments remained for the forecast horizon of 3 days. The distribution of test set NSE across all 620 catchments and the filtered catchments (for both forecast horizons) for each of the 33 wavelet filters are shown in Figures 12 and 13, respectively. The distributions were estimated using kernel density estimation. Due to the estimation procedure used to compute the kernel density plots, non-zero densities occur beyond an NSE of 1 and below the minimum observed NSE.

Given that the kernel density estimation method used to generate Figures 12 and 13 adopted a Gaussian kernel, a non-zero density can be assigned to values beyond a particular threshold. For example, the lower-bound threshold on the test set NSE of 0.4 associated with the B-LSTM was violated in Figure 13. Thus, a modification was made to the original distribution of B-LSTM (based on kernel density estimation) such that it respected this threshold. However, W-LSTM does not have the same lower-bound threshold and naturally includes non-zero densities below a test set NSE of 0.4.

It is clear from Figures 12 and 13 that the W-LSTM has higher performance than B-LSTM by noticing that W-LSTM (for at least one wavelet filter) generally has higher densities than B-LSTM for high NSE values (i.e., NSE > 0.6). Although certain wavelet filters result in poor W-LSTM performance across several catchments, W-LSTM generally exhibits lower densities than B-LSTM for low NSE values at the one-day ahead forecast horizon (Figure 12). However, W-LSTM with the db1 wavelet filter is the only model to exhibit higher density estimates than B-LSTM for the three-day ahead forecast horizon across most NSE values (except at the tails) (Figure 12).

Numerical values of the median test set NSE percent improvement of W-LSTM over B-LSTM (median taken over catchments) were computed for both one day and three day(s) forecast horizons. Table 3 considers all 620 catchments while Table 4 considers filtered catchments where B-LSTM demonstrated a test set NSE greater than 0.4.

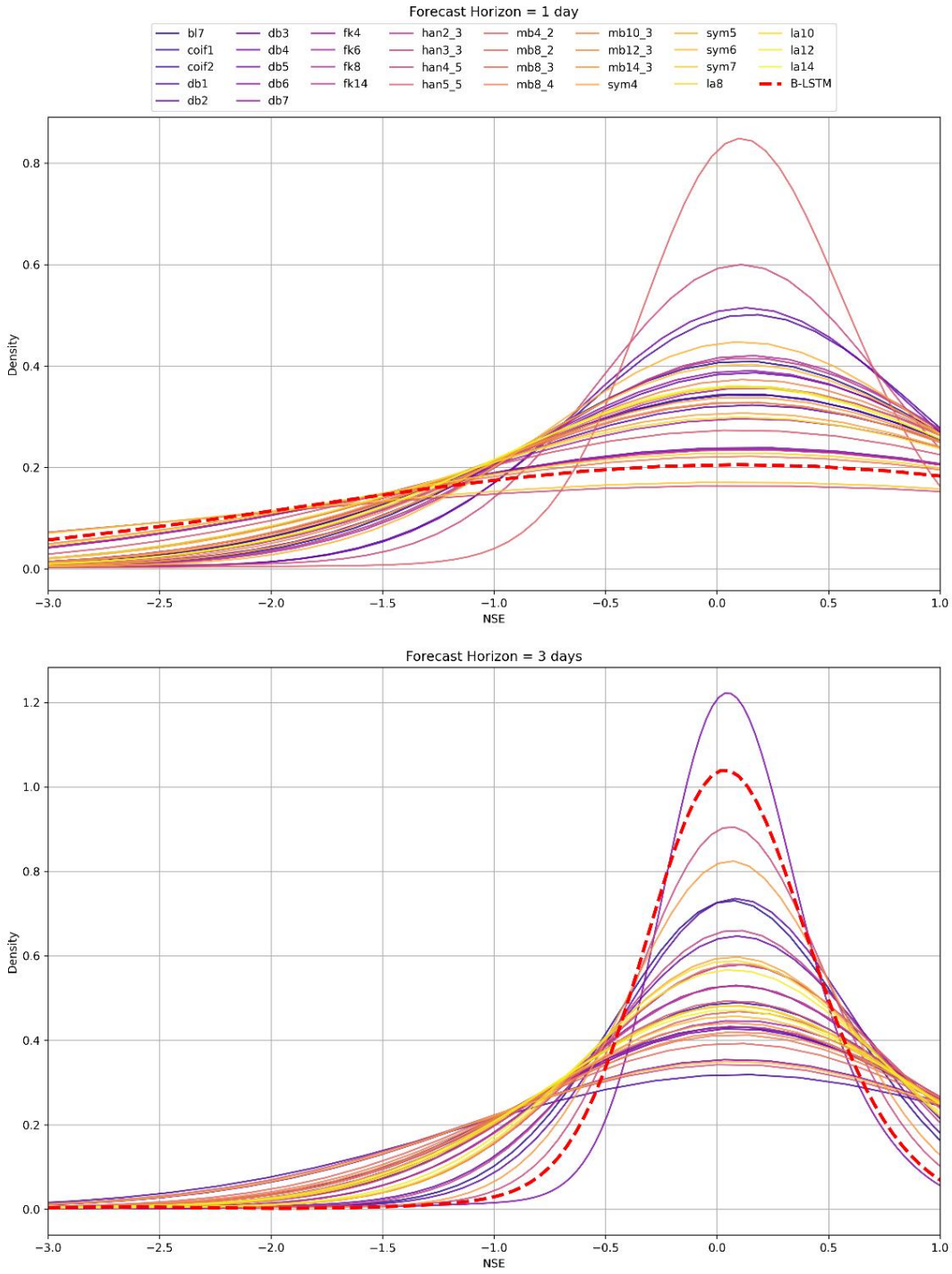


Figure 12. Distribution of test set NSE across all 620 CAMELS catchments for B-LSTM (red coloured line) and the W-LSTM using different wavelet filters

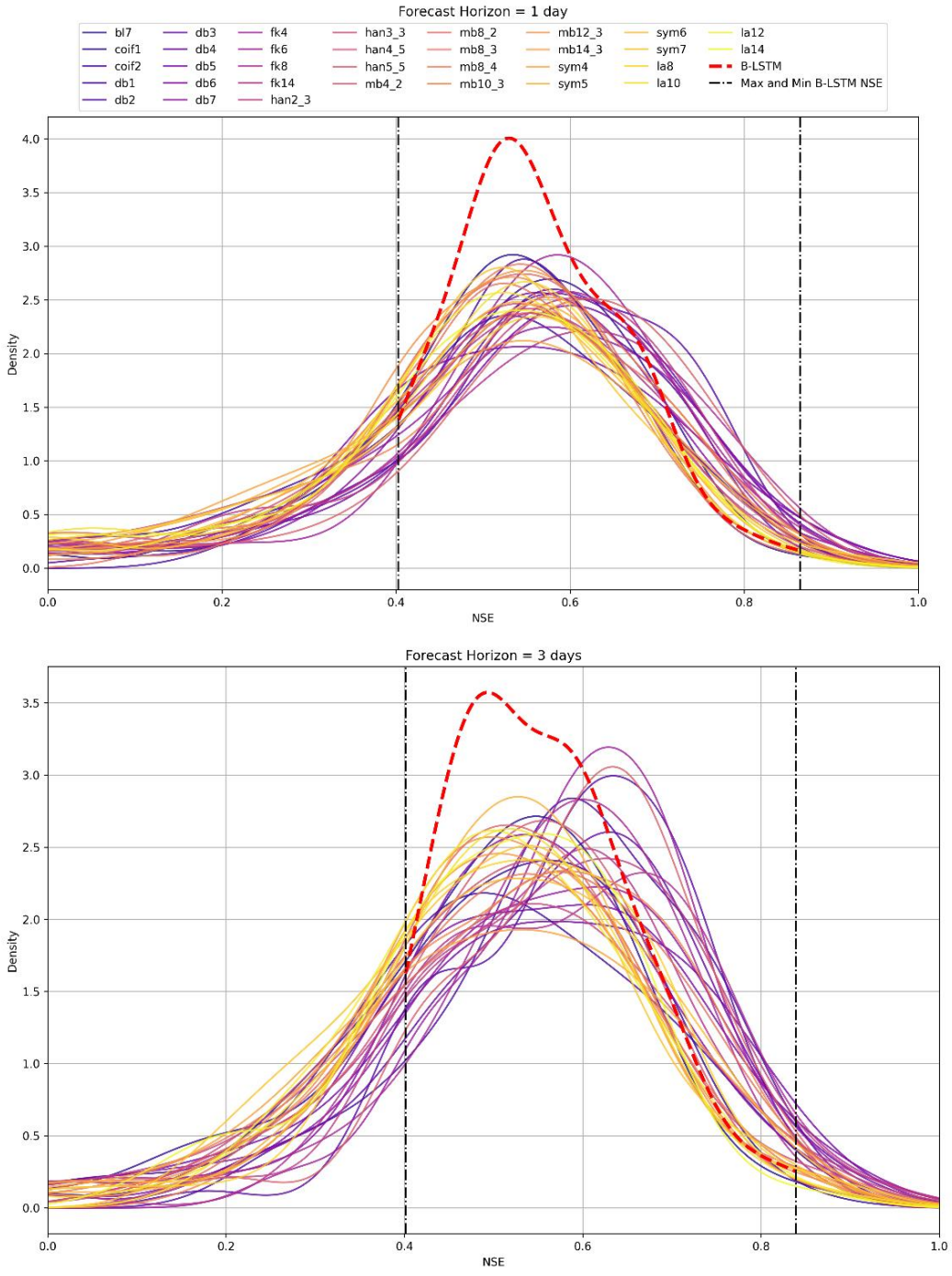


Figure 13. Distribution of test set NSE across 90 and 60 catchments (for forecast horizons 1 and 3 days, respectively) where the B-LSTM (red coloured line) achieved a test set NSE > 0.4; the different coloured lines represent different wavelet filters used in the W-LSTM.

Table 3. Median percent improvement (PI) in test set NSE over B-LSTM and median NSE for all wavelet filters and forecast horizons; all 620 catchments considered.

Wavelet filter	Median NSE PI over B-LSTM (LT1)	Median NSE (LT1)	Median NSE PI over B-LSTM (LT3)	Median NSE (LT3)
bi7	0.6761	0.0127	-0.9796	-0.0005
coif1	2.004	0.0363	2.899	0.0137
coif2	0.6206	0.034	2.3735	0.0032
db1	5.2925	0.0787	14.443	0.0372
db2	3.9967	0.0556	8.0526	0.0235
db3	2.7733	0.0566	9.2409	0.0241
db4	3.1437	0.0348	6.5108	0.0182
db5	2.4551	0.0394	3.1793	0.0135
db6	0.7276	0.0245	2.413	0.0093
db7	2.2096	0.0351	2.7242	0.0058
fk4	6.4453	0.0821	9.1878	0.0375
fk6	4.3456	0.0546	5.9635	0.0224
fk8	4.0209	0.0559	10.8056	0.0315
fk14	3.2779	0.0473	6.4022	0.0208
han2_3	4.53	0.0477	5.1685	0.0216
han3_3	-0.0482	0.0088	1.7883	0.0019
han4_5	0.9082	0.0293	0.3305	0
han5_5	0.3755	0.0007	0.6992	-0.0005
mb4_2	4.2993	0.0409	12.3866	0.0347

mb8_2	1.0943	0.0335	2.1613	0.0174
mb8_3	2.9836	0.05	8.5202	0.0308
mb8_4	1.5998	0.0361	6.2931	0.0257
mb10_3	2.5609	0.0464	9.3479	0.0239
mb12_3	0.2801	0.028	0.7925	0
mb14_3	-0.071	0.0124	0.2829	0
sym4	0.0919	0.0248	2.3318	0.0126
sym5	-0.1734	0.0014	2.1188	0
sym6	0.8067	0.0047	0.7826	0
sym7	0.0127	0	-0.2064	-0.0004
la8	-0.4541	0.0141	1.3318	0
la10	-0.6861	0.0112	0.5868	-0.0001
la12	-0.542	0.001	0.3279	-0.0005
la14	-0.176	0.0036	-1.0321	-0.0008

It is clear from Figures 12 and 13 that different wavelet filters lead to different performance levels when used in the W-LSTM. To provide a general indication of how the performance of W-LSTM compares to B-LSTM, the median NSE of W-LSTM and B-LSTM was taken across all catchments (Table 3 and Figure 14) as well as those catchments where B-LSTM achieved an NSE of 0.4 (Table 4 and Figure 15). Considering all catchments, it can be seen in Figure 14 and Table 3 that most wavelet filters lead to W-LSTM models with a higher median NSE than B-LSTM at both the one and three day(s) ahead forecast horizons. For the filtered catchments, Figure 15 and Table 4 show that six and 12 of the wavelet filters lead to W-LSTM models with a higher median NSE than B-LSTM for one and three day(s) ahead forecast horizons, respectively. If, on a per catchment basis, the wavelet filter leading to the best NSE is considered (i.e., the optimal W-LSTM), more substantial improvements in forecast performance are realized (Figure 14-15). Likewise, if choosing optimal hyper-parameters (Section

4.1.3) is considered instead of WTs (i.e., optimal GS-LSTM), a similar improvement in forecast performance is realized. Results pertaining to optimal W-LSTM and GS-LSTM models are explored in more detail in Sections 5.2 and 5.3, respectively.

Table 4. Median percent improvement (PI) in test set NSE over B-LSTM, as well as median NSE for all wavelet filters and forecast horizons; 90 and 60 catchments considered (for forecast horizons one and three day(s) ahead, respectively), where the B-LSTM achieved a test set NSE > 0.4.

Wavelet filter	Median NSE PI over B-LSTM (LT1) (%)	Median NSE (LT1)	Median NSE PI over B-LSTM (LT3)	Median NSE (LT3) (%)
bi7	-4.8916	0.5274	-7.754	0.5187
coif1	-1.0935	0.5487	-5.7016	0.5246
coif2	-2.1937	0.5169	-5.2485	0.4915
db1	-1.3789	0.5546	6.6751	0.5838
db2	1.2099	0.5845	4.9668	0.6117
db3	-0.3691	0.5731	5.1711	0.6105
db4	0.3825	0.5764	3.52	0.5529
db5	-2.4023	0.5448	-0.0083	0.5497
db6	-0.3155	0.5406	-2.9565	0.5353
db7	-0.7335	0.5521	-0.0832	0.5512
fk4	1.2111	0.576	4.4414	0.603
fk6	0.6125	0.5715	2.8799	0.5787
fk8	-0.1269	0.5786	4.3742	0.6087
fk14	-1.5815	0.5443	0.1613	0.5755
han2_3	1.265	0.5781	3.351	0.5878
han3_3	-4.6694	0.5313	-1.9034	0.5313

han4_5	-3.8314	0.5256	-4.7359	0.5296
han5_5	-3.3383	0.5376	-3.6746	0.5295
mb4_2	2.0931	0.5956	6.2628	0.6099
mb8_2	-4.4093	0.5302	-5.2149	0.5053
mb8_3	-2.8547	0.5373	0.3847	0.5553
mb8_4	-3.2419	0.5264	-5.2743	0.5245
mb10_3	-2.9114	0.554	1.5996	0.5538
mb12_3	-3.8725	0.525	-8.8012	0.5019
mb14_3	-5.4726	0.534	-5.4839	0.4908
sym4	-5.0192	0.5138	-4.6338	0.5012
sym5	-6.1208	0.529	-6.1113	0.5229
sym6	-4.1748	0.5237	-1.2746	0.527
sym7	-5.7907	0.5188	-6.3917	0.5121
la8	-5.5733	0.5254	-2.1643	0.5264
la10	-5.6052	0.5272	-7.5883	0.4968
la12	-5.4977	0.5129	-6.9591	0.5031
la14	-6.1429	0.5251	-4.9009	0.52

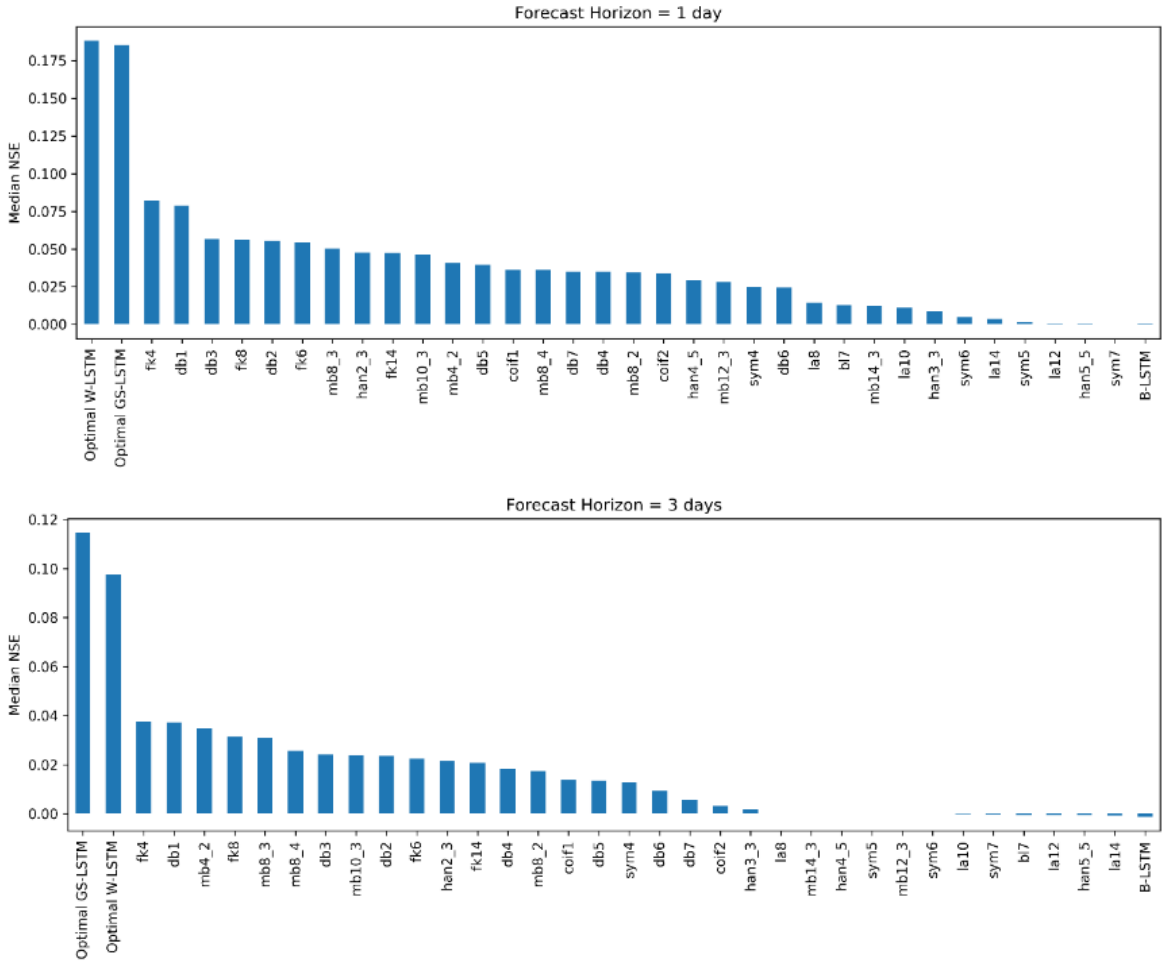


Figure 14. Median NSE across all 620 catchments for the various models considered in this thesis (see Section 4.1).

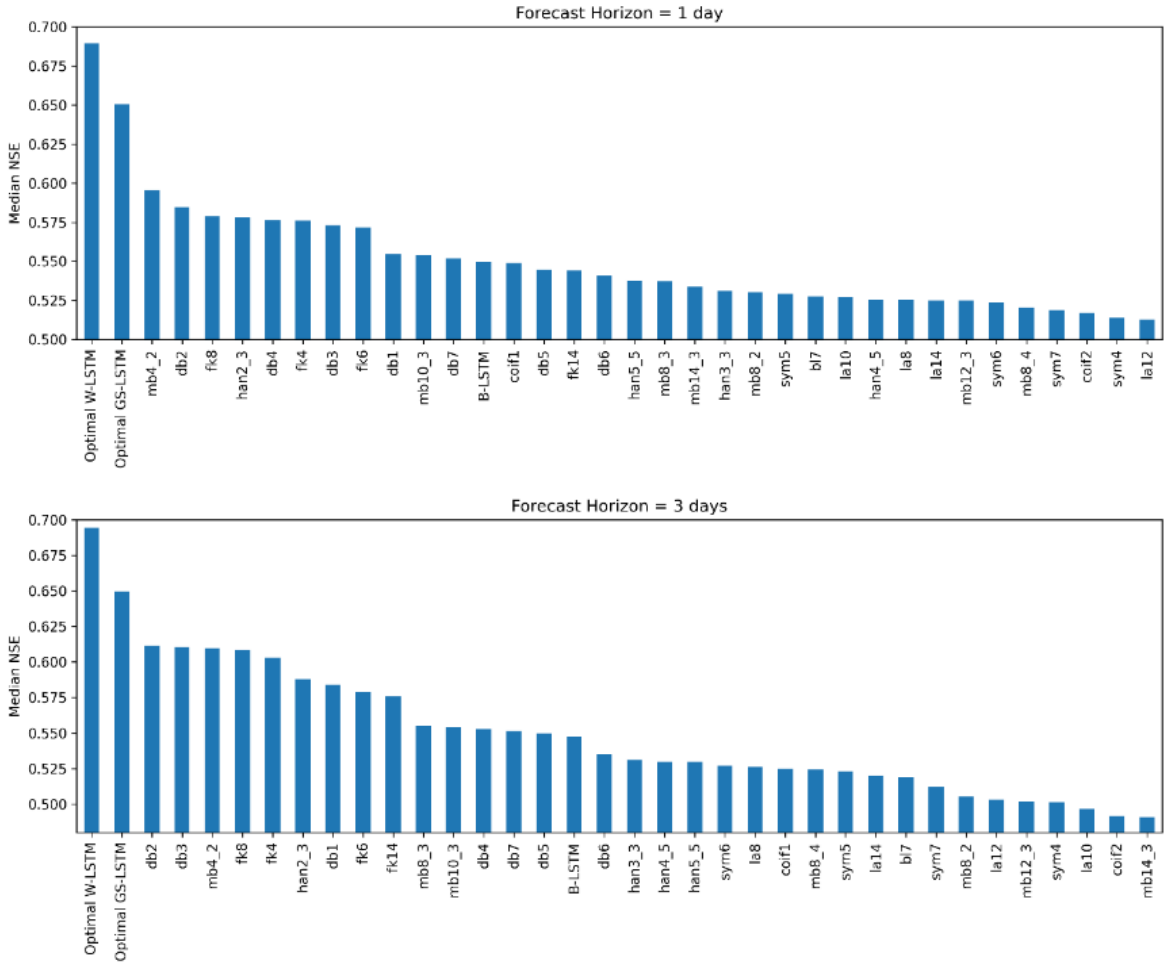


Figure 15. Similar to Figure 14, but only considering the filtered catchments (90 and 60 catchments, respectively, for one and three day(s) ahead forecast horizons) where the B-LSTM had an NSE > 0.4.

Furthermore, for each catchment and wavelet filter, a pairwise comparison was carried out between the W-LSTM model and the B-LSTM. The fraction of catchments where the W-LSTM had a higher NSE than the B-LSTM model was computed. These results are summarized in Figure 16 for all 620 catchments and Figure 17 for the filtered catchments (i.e., where B-LSTM had an NSE higher than 0.4). It is clear that certain wavelet filters lead to W-LSTM models that outperform B-LSTM in more than 50% of catchments (both when considering all 620 catchments, as well as when considering only the filtered catchments).

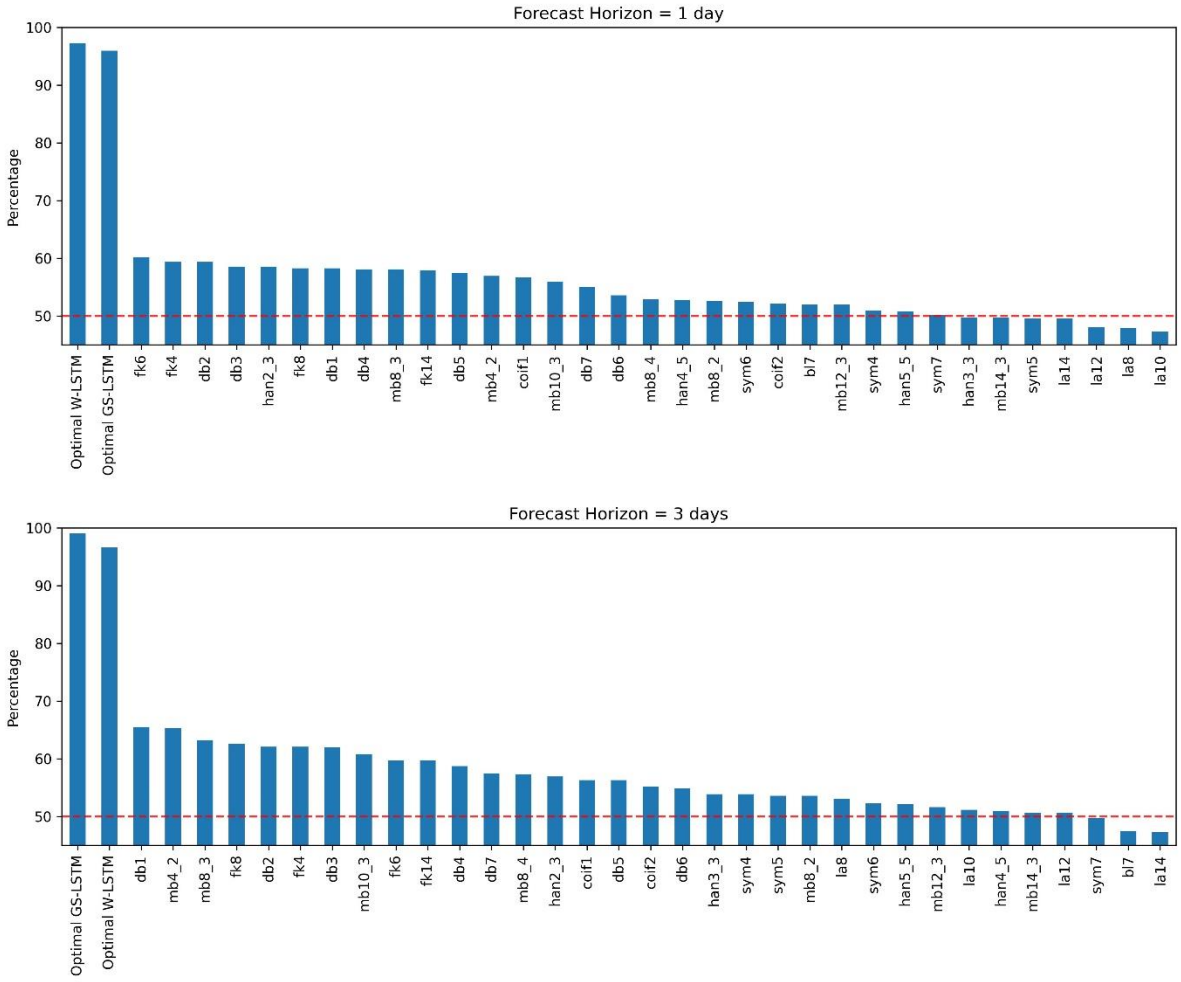


Figure 16. Fraction of catchments (out of 620) where the LSTM model specified in the x-axis outperformed the B-LSTM (in terms of test set NSE). Most W-LSTM models outperformed the B-LSTM for the majority of the 620 catchments. The red horizontal line indicates the 50 percent mark.

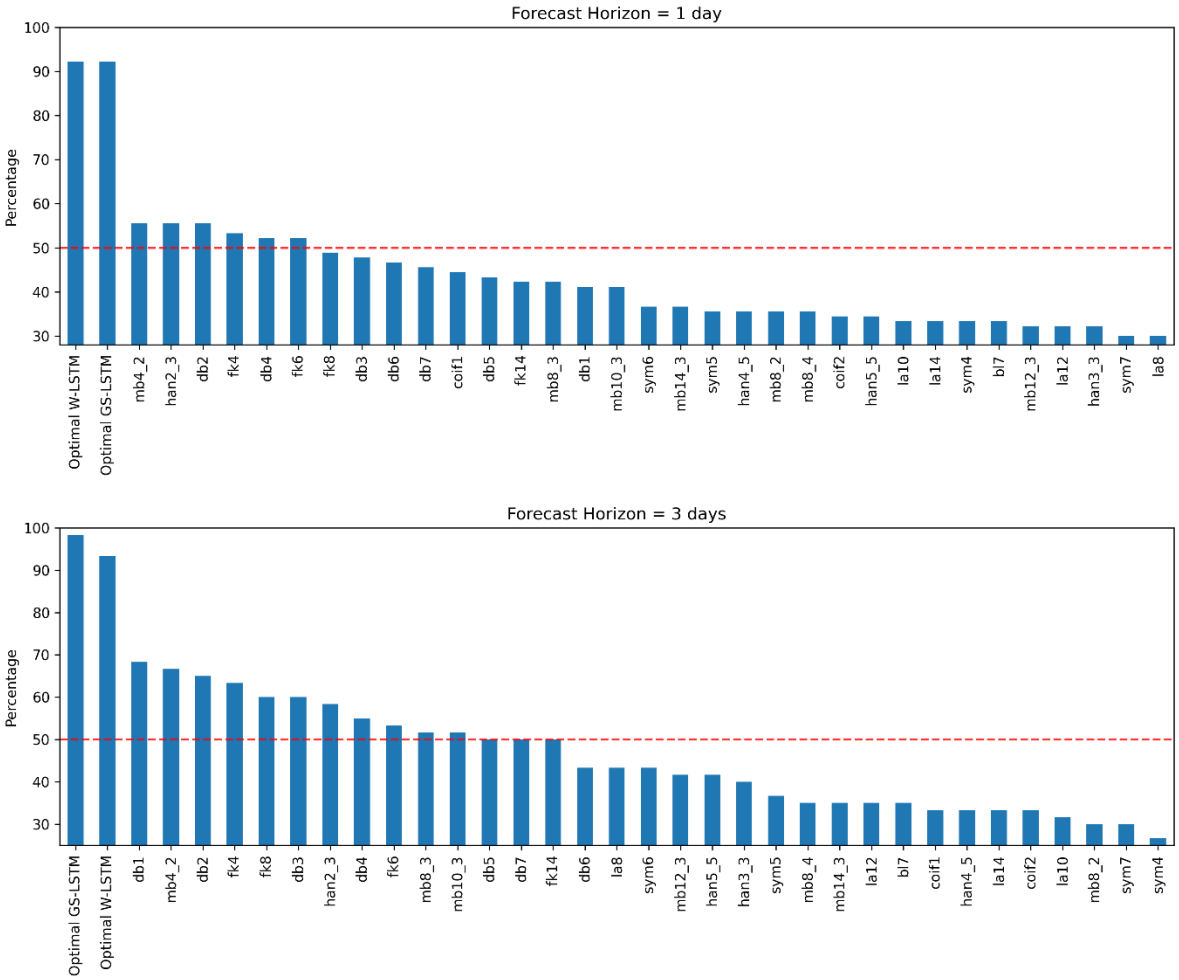


Figure 17. Similar to Figure 16, but now considering the fraction of filtered catchments (catchments where the B-LSTM achieved a test NSE > 0.4) where the LSTM model specified in the x-axis outperformed the B-LSTM (in terms of the test set NSE); 90 and 60 catchments were considered for forecast horizons one and three day(s) ahead, respectively.

The results presented in this sub-section indicate that certain wavelet filters tend to enhance the performance of B-LSTM more than others do across both catchments and forecast horizons, accomplishing the first objective of this thesis. Wavelet filters that performed well in general (across different catchments and forecast horizons) include “db1” (Daubechies family), “fk4” (Fejér-Korovkin family), “mb4_2” (Morris minimum-bandwidth family) and “han2_3” (Han linear-phase moments family). The reasons for such discrepancies among different wavelet filters in terms of enhancing forecast performance are not addressed in this thesis and are left as a direction for future research.

Thus, WTs evidently provide a positive impact on streamflow forecasting performance when considering the CAMELS dataset and LSTM as the benchmark model. This result is important as the CAMELS dataset includes a multitude of hydro-climatologically diverse catchments (see Section 5.4). Whether the performance gains provided by WTs (as seen in this sub-section) apply to other ML models (e.g., Transformers, XGBoost) is left to future research. However, the results presented herein can act as a starting point for future studies.

The next sub-section focuses on the results of the optimal W-LSTM models.

5.2 Optimal Wavelet Filters

For each of the 33 W-LSTM models (each corresponding to one of the 33 wavelet filters), the wavelet filter that led to the highest validation set NSE was used to identify the "optimal" wavelet filter for each catchment. For a given catchment, the W-LSTM model that used the optimal wavelet filter for feature engineering is hereafter referred to as the optimal W-LSTM model.

The distribution of NSE across all 620 CAMELS catchments, as well as the filtered catchments for B-LSTM, optimal W-LSTM, and GS-LSTM models, are shown in Figures 18 and 19, respectively (the filtered catchments include 90 and 60 catchments for one and three day(s) ahead forecast horizons, respectively). This sub-section focuses on the optimal W-LSTM performance, while a more detailed comparison between W-LSTM and GS-LSTM is the subject of the next sub-section.

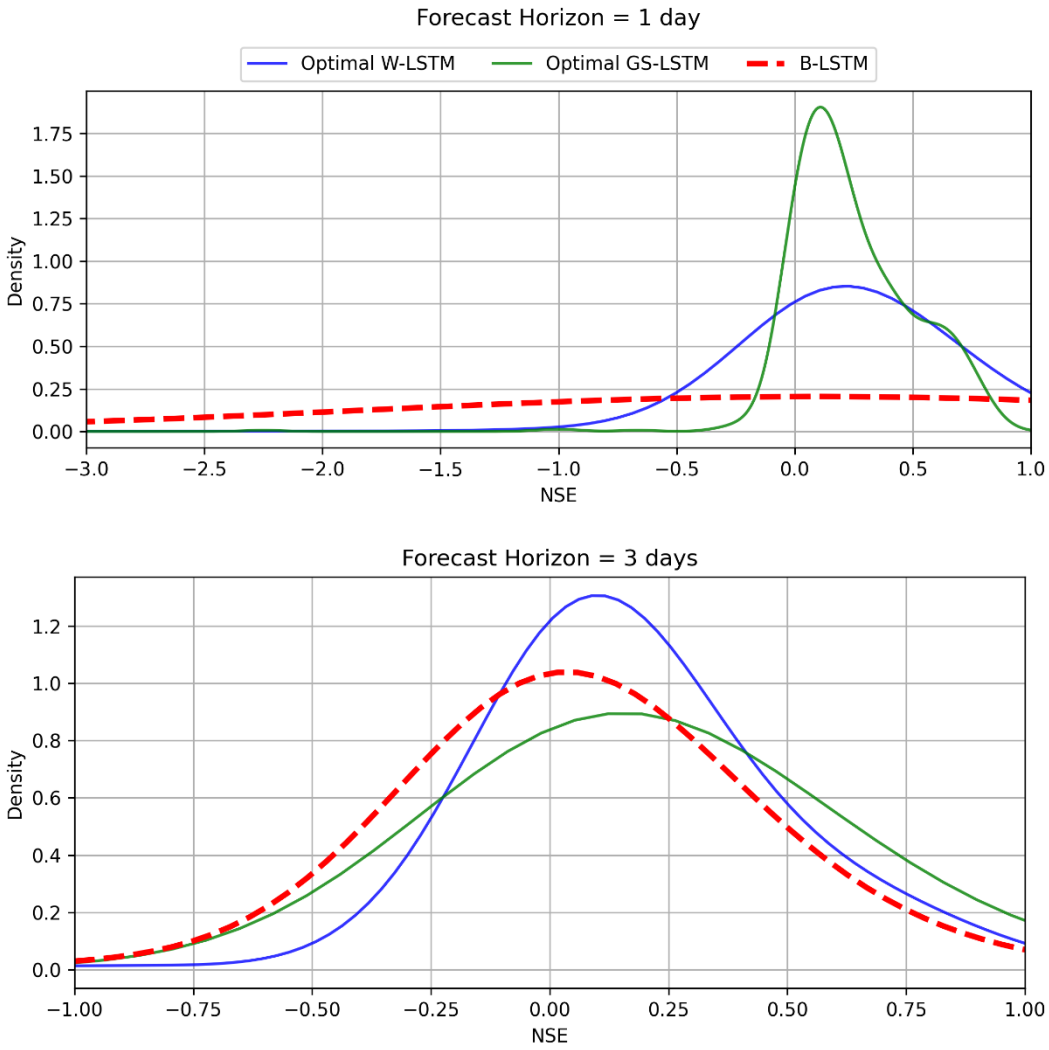


Figure 18. Comparing the distribution of test set NSE values across all 620 catchments for the optimal W-LSTM and GS-LSTM models against the B-LSTM.

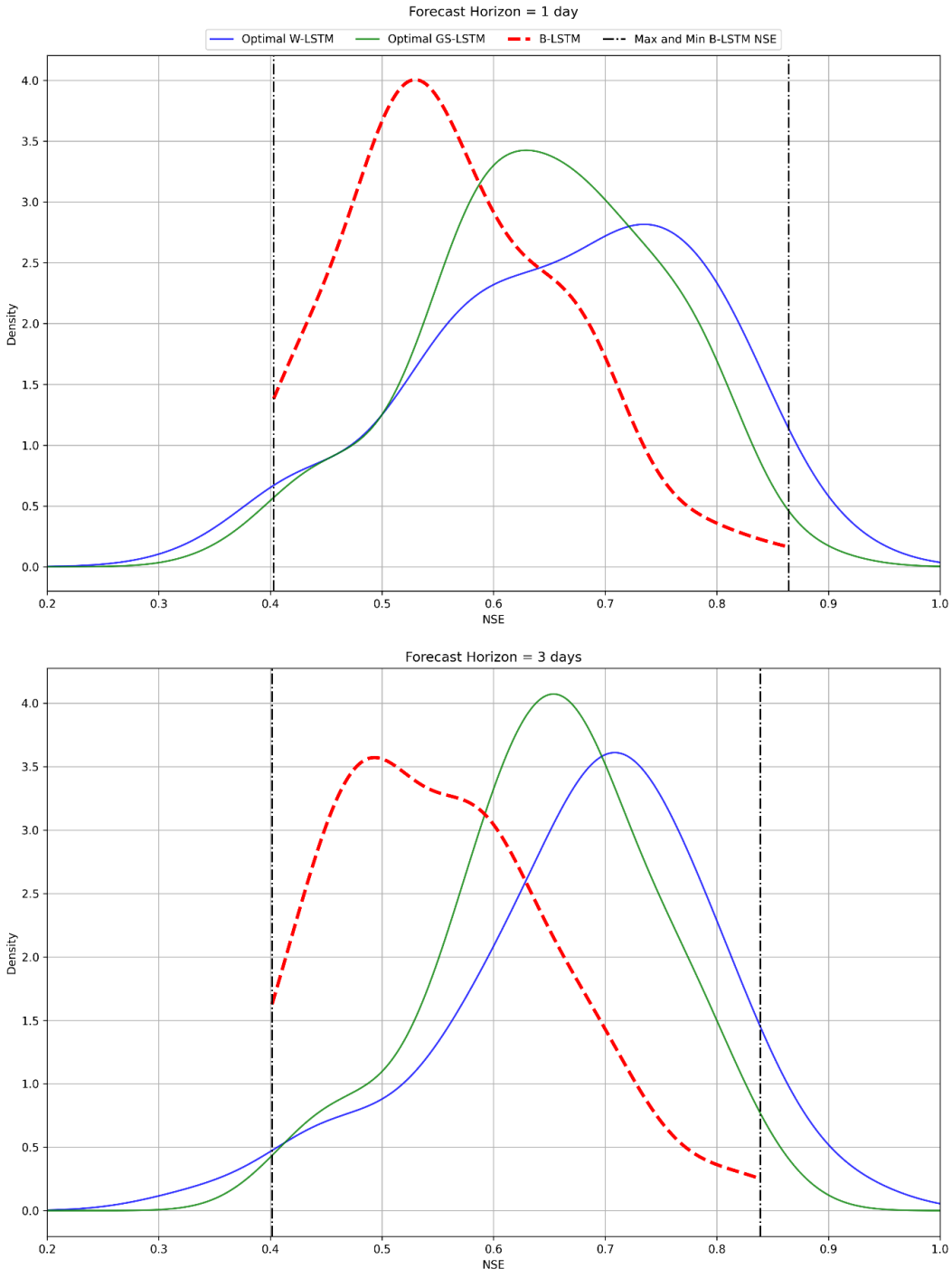


Figure 19. Similar to Figure 18, except the distribution of test set NSE values of the optimal W-LSTM and GS-LSTM models are compared against B-LSTM for the filtered catchments

(catchments where the B-LSTM achieves a test NSE > 0.4). There were 90 and 60 filtered catchments for the one and three day(s) ahead forecast horizons, respectively.

Across the entire 620 CAMELS catchments as well as the filtered catchments, the optimal W-LSTM models tended to have higher (lower) density for high (low) NSE values than B-LSTM. This indicates that W-LSTM models based on an optimal wavelet filter (that may be different for each catchment) tend to outperform the B-LSTM. However, to arrive at this optimal W-LSTM model, there is a necessary increase in computation time relative to B-LSTM. The increase in computation time is related to the number of wavelet filters under consideration (33 in this thesis) as well as the increased time required to perform feature engineering via WTs (see Section 4.1.2) and feature selection (see Section 4.2.4). Thus, for each catchment, 33 W-LSTM models needed to be trained and evaluated against the B-LSTM to estimate the increase in performance offered by WTs relative to B-LSTM.

In total, $33 \times 620 = 20{,}460$ W-LSTM models were trained (in addition to 620 B-LSTM models and $32 \times 620 = 19{,}840$ GS-LSTM models) to carry out the experiments. Table 5 shows the median NSE percent improvement of optimal W-LSTM and optimal GS-LSTM over B-LSTM for both one and three day(s) ahead forecast horizons, as well as for both unfiltered and filtered catchment sets.

Across 620 CAMELS catchments, at the one day ahead forecast horizon, the optimal W-LSTM models had a median NSE of 0.1880, whereas the B-LSTM models had a median NSE of approximately -0.0002. In other words, across all 620 CAMELS catchments, the optimal W-LSTM led to a relative percent improvement of 117,600% in the median NSE when compared to B-LSTM for a one day ahead forecast horizon. At the three days ahead forecast horizon, the optimal W-LSTM led to a relative improvement of 8406% in median NSE compared to B-LSTM. Given the very small median NSE of the B-LSTM, the absolute improvement in median NSE of 0.1882 and 0.0986 achieved by the optimal W-LSTM at one and three day(s) ahead forecast horizons (in comparison to the B-LSTM), respectively, is easier to evaluate.

Table 5. Median pairwise percent increase (PI) in NSE (over B-LSTM) and median NSE for optimal W-LSTM and optimal GS-LSTM for one and three days(s) ahead forecast horizons (LT1 and LT3, respectively)

LSTM Configuration	Median NSE PI over B-LSTM (LT1) (%)	Median NSE (LT1)	Median NSE PI over B-LSTM (LT3) (%)	Median NSE (LT3)
	1 Day Ahead Forecast Horizon		3 Days Ahead Forecast Horizon	
B-LSTM	0.000	0.000	0.000	0.000
Optimal W-LSTM	60.461	0.188	88.658	0.097
Optimal GS-LSTM	99.442	0.185	149.728	0.115
B-LSTM (Filtered)	0.000	0.550	0.000	0.547
Optimal W-LSTM (Filtered)	15.744	0.690	19.557	0.694
Optimal GS-LSTM (Filtered)	9.614	0.651	14.737	0.649

Considering the 90 filtered catchments at the one day ahead forecast horizon, the median NSE for the optimal W-LSTM models was 0.6895 while 0.5495 for B-LSTM. Thus, for this case, the optimal W-LSTM improved upon the median NSE of B-LSTM by 25%. For the 60 filtered catchments at the three days ahead forecast horizon, the optimal W-LSTM led to a 27% improvement in median NSE compared to the B-LSTM.

At the one day ahead forecast horizon, the optimal W-LSTM outperformed B-LSTM in 97.3% of the 620 CAMELS catchments, while for the 90 filtered catchments (at the same forecast horizon), the optimal W-LSTM outperformed the B-LSTM in 92.2% of the filtered catchments. At the three days ahead forecast horizon, these values are 96.6% and 93.3%, respectively.

These findings demonstrate that for most catchments (i.e., greater than 90%), optimal W-LSTM outperforms B-LSTM for filtered and unfiltered catchments at both forecast horizons (one and three day(s) ahead). These results, however, come at the expense of extra computational time.

Training optimal W-LSTM takes, on average, 3.7 hours, while training B-LSTM takes, on average, 3.3 minutes (i.e., training optimal W-LSTM takes, on average, 68 times longer than training B-LSTM). For a fairer comparison that considers computational cost, the optimal W-LSTM and GS-LSTM can be compared. Figures 18, 19, and Table 5 demonstrate that optimal W-LSTM provides similar performance and often superior performance (depending on the catchments considered and forecast horizon) to optimal GS-LSTM. This is an important finding since optimal GS-LSTM leverages hyper-parameter selection via grid search, which is a common approach for improving ML model performance (Bergstra & Bengio, 2012).

A more useful comparison value to consider (compared to the percent increase in median NSE for two model configurations) is the median of the pairwise NSE percent increase computed for each catchment. Pairwise NSE percent increase is computed by finding the percent increase in NSE of the optimal W-LSTM over B-LSTM for every catchment and then taking the median of those pairwise PI values. Doing this showed that across all 620 catchments, optimal W-LSTM provided a 60% median pairwise percent increase in NSE over B-LSTM for the one day forecast horizon and an 89% median pairwise percent increase in NSE for the three day(s) ahead forecast horizon. For the filtered catchments, optimal W-LSTM demonstrated a 16% median pairwise percent increase in NSE for the one day forecast horizon and a 20% median pairwise percent increase in NSE for the three day(s) ahead forecast horizon.

One notable finding is that optimal GS-LSTM outperformed optimal W-LSTM (in terms of median pairwise percent increase in NSE) when all 620 catchments were considered. However, optimal W-LSTM outperformed optimal GS-LSTM when only filtered catchments (catchments where B-LSTM achieved an NSE greater than 0.4) were considered (see Table 5). The non-filtered catchment results are not as important for this thesis because many of the catchments (of the 620 catchments) yielded very low NSE values for B-LSTM. Thus, the optimal GS-LSTM models that outperform optimal W-LSTM in these catchments often result in poorly performing models. For this reason, the key results are those associated with the filtered results (Table 5), which indicate that optimal W-LSTM

outperformed GS-LSTM (in terms of median pairwise percent increase in NSE) for both one and three day(s) ahead forecast horizons.

A more detailed comparison of optimal W-LSTM and optimal GS-LSTM models is provided in the following sub-section.

5.3 Comparing Optimal Wavelet Filters Against Hyper-Parameter Selection

The GS-LSTM model was used as an additional benchmark for W-LSTM. Similar to W-LSTM, GS-LSTM required the development of multiple models (32 in total) to find the combination of hyper-parameters (number of hidden units, dropout rate, and sequence length) that led to the best performance for each catchment and forecast horizon (one and three day(s) ahead). The goal of this experiment was to replicate a scenario where a modeler must choose (perhaps due to computational constraints) between two options:

1. W-LSTM - Developing LSTM models with a fixed architecture that use WT-engineered features based on different WT parameters (in this case, wavelet filters). Then, selecting the combination of WT parameters that leads to the best performance (see Section 4.1.2).
2. GS-LSTM - Not using WTs and developing different LSTM architectures (e.g., based on the number of hidden units, dropout rate, and sequence length), then selecting the combination of hyper-parameters that gives the best performance (see Section 4.1.3).

Thus, to provide a fair comparison between these two options with respect to computation time, a similar number of models should be developed for W-LSTM and GS-LSTM. Since this research developed 33 W-LSTM models for each catchment and forecast horizon, a similar number of architectures based on the number of hidden units, dropout rate, and sequence length was sought for GS-LSTM. The closest number of unique architectures that could be reached based on these three hyper-parameters was 32 (see Section 4.1.3).

Considering the optimal W-LSTM and GS-LSTM models, the distribution of NSE across all catchments and the filtered catchments is provided in Figures 18 and 19, respectively. Interestingly, when considering all catchments at the one day ahead forecast horizon, the optimal W-LSTM models have higher densities than GS-LSTM in the regions of the distribution where NSE is very low and high. This result is reversed at the three days ahead forecast horizon. Considering the filtered catchments (90 and 60 catchments, for the one and three day(s) ahead forecast horizons, respectively), the optimal W-

LSTM model has higher densities (compared to optimal GS-LSTM) in regions of the distributions where NSE is high. Thus, for the filtered catchments, there are a higher number of better-performing optimal W-LSTM models (compared to optimal GS-LSTM) when considering high NSE values. In this case, it is not surprising that the optimal W-LSTM leads to better performance than the optimal GS-LSTM, given the results in Section 5.2, where optimal W-LSTM is shown to improve upon B-LSTM in almost every catchment.

For the one and three day(s) ahead forecast horizons, the median NSE across all 620 CAMELS catchments for the optimal W-LSTM models were 0.1880 and 0.0974, whereas for the optimal GS-LSTM, the median NSE was 0.1853 and 0.1145, respectively. For the filtered catchments at the one and three day(s) forecast horizons, the median NSE for the optimal W-LSTM was 0.6895 and 0.6944. In comparison, the optimal GS-LSTM had median NSE values of 0.6506 and 0.6494, respectively. See Table 5 for a concise summary.

At the one day ahead forecast horizon, the optimal W-LSTM outperformed the optimal GS-LSTM in only 49.8% of 620 CAMELS catchments. However, for the 90 filtered catchments the optimal W-LSTM outperformed the optimal GS-LSTM in 60% of the catchments. Considering the forecast horizon of three days ahead, these values are 41.3% and 70%, respectively.

The results above reveal two important findings:

1. Considering all 620 CAMELS catchments, the optimal W-LSTM and GS-LSTM models outperform one another at about the same rate at the one day ahead forecast horizon while the optimal GS-LSTM performs better in higher number of catchments than the optimal W-LSTM at the three days ahead forecast horizon.
2. For the filtered catchments, where B-LSTM achieved an NSE > 0.4, the optimal W-LSTM outperforms the optimal GS-LSTM in a higher number of catchments at both the one and three day(s) ahead forecast horizons.

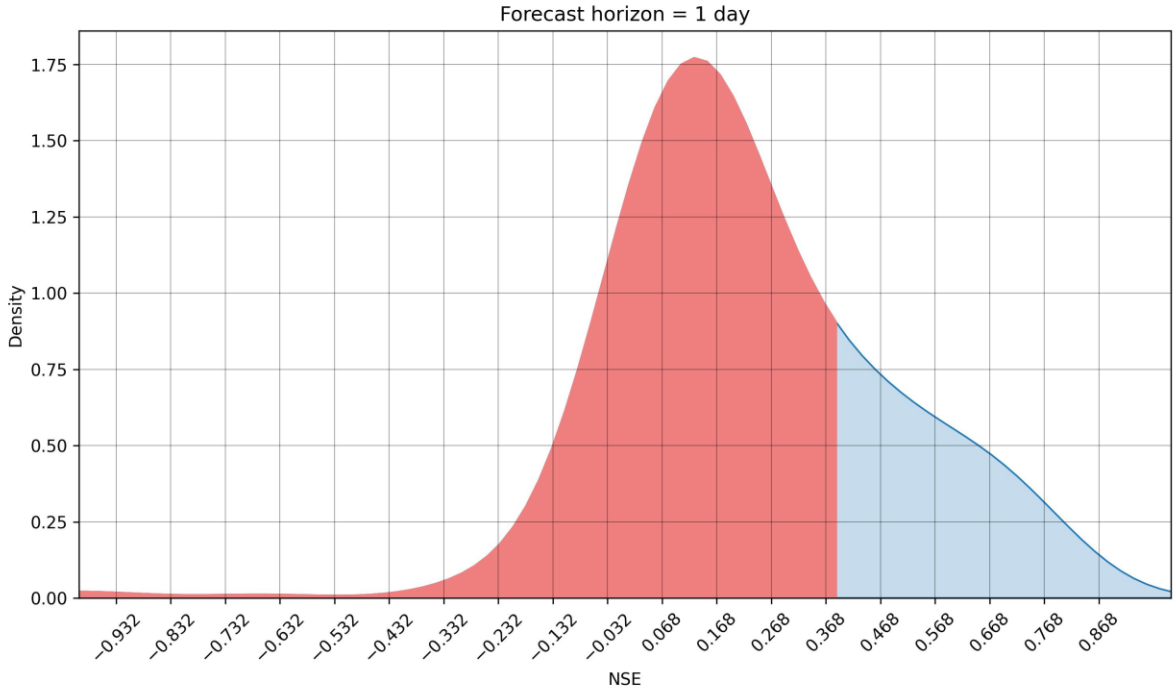
As noted in Section 5.2, the results across all 620 catchments that indicate a general superiority of optimal GS-LSTM over optimal W-LSTM hold less relevance compared to the results obtained using the filtered catchments (i.e., where B-LSTM achieved an NSE greater than 0.4). The reason for this is that for many of the 620 catchments where optimal GS-LSTM outperformed optimal W-LSTM, the NSE value achieved by optimal GS-LSTM was still too low to be considered a useful model for

practical streamflow forecasting (Moriasi et al., 2007). The filtering of catchments where B-LSTM achieved an NSE greater than 0.4 was done to address this problem, by eliminating catchments where B-LSTM had difficulty achieving satisfactory forecasting accuracy. Thus, this filtering technique can be used as a proxy to avoid training both optimal W-LSTM and optimal GS-LSTM, which would be of little practical value. Overall, the results from this thesis show that WT-based feature engineering can enhance short-term (one and three day(s) ahead) streamflow forecasting performance and are particularly beneficial for the CAMELS dataset and LSTM models that achieve NSE greater than 0.4.

To illustrate the point above, Figure 20 shows a kernel density estimation plot of the GS-LSTM model's NSE at one and three day(s) ahead forecast horizons for the catchments (of which there are 311 and 364, respectively) where optimal GS-LSTM outperformed optimal W-LSTM.

For one day ahead forecasting, of the 311 catchments where optimal GS-LSTM outperformed optimal W-LSTM, only 76 catchments (24%) produced an optimal GS-LSTM model that also had NSE greater than 0.4. For three days ahead forecasting, of the 364 catchments where optimal GS-LSTM outperformed optimal W-LSTM, only 39 catchments (11%) produced an optimal GS-LSTM model that also had NSE greater than 0.4. Thus, the bulk of the optimal GS-LSTM models that outperformed optimal W-LSTM still performed poorly (i.e., NSE < 0.4).

Since W-LSTM tends to perform better than GS-LSTM in catchments where B-LSTM achieves an NSE > 0.4, and GS-LSTM tends to perform better than W-LSTM in catchments where B-LSTM achieves an NSE < 0.4, it is expected that combining WT generated features with GS-LSTM would lead to even better performance. This topic is outside the scope of this thesis but represents an interesting line of future research.


Forecast horizon = 1 day
Density
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
−0.932
−0.832
−0.732
−0.632
−0.532
−0.432
−0.332
−0.232
−0.132
−0.032
0.068
0.168
0.268
0.368
0.468
0.568
0.668
0.768
0.868
NSE

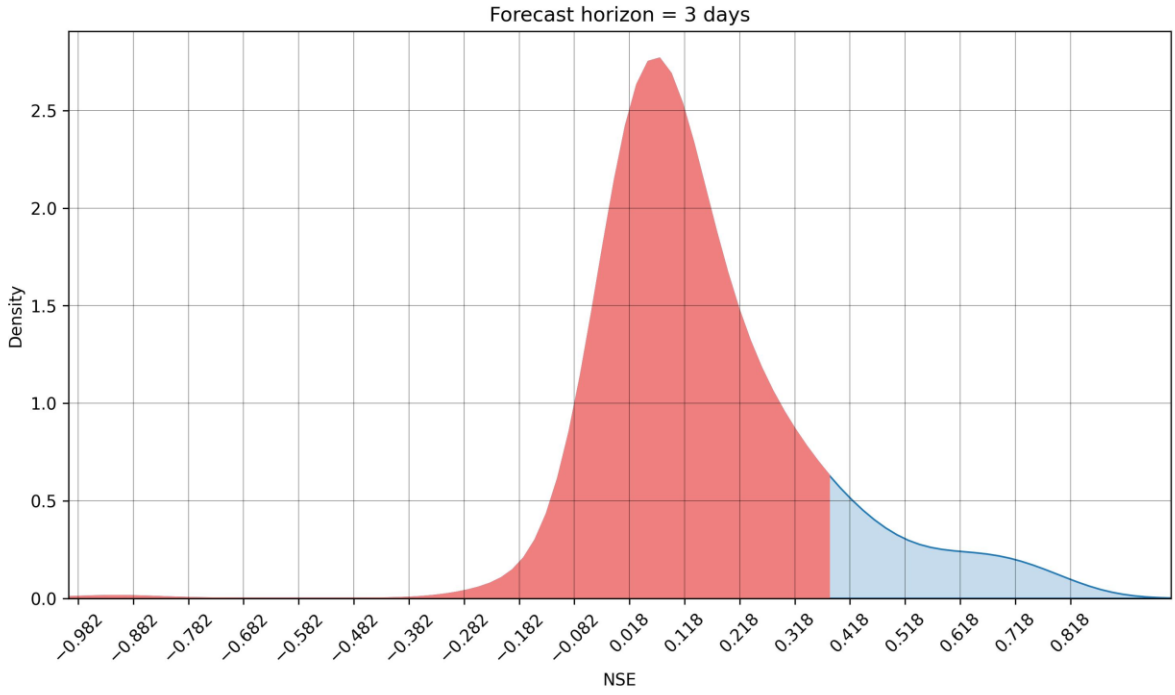

Forecast horizon = 3 days
Density
2.5
2.0
1.5
1.0
0.5
0.0
−0.982
−0.882
−0.782
−0.682
−0.582
−0.482
−0.382
−0.282
−0.182
−0.082
0.018
0.118
0.218
0.318
0.418
0.518
0.618
0.718
0.818
NSE

Figure 20. Distribution of the GS-LSTM model's NSE for catchments where optimal GS-LSTM outperformed optimal W-LSTM. The red region indicates the area under the probability density function to the left of the 0.4 NSE threshold.

So far, this chapter has described how different wavelet filters affected the performance of LSTM models for one and three day(s) ahead streamflow forecasting for the CAMELS dataset. This section has shown how models based on the best wavelet filters for each catchment (i.e., optimal W-LSTM) compare against models adopting hyper-parameter selection (i.e., GS-LSTM). The next sub-section explores the characteristics of the catchments where WT-generated features provided the greatest improvement in forecasting performance compared to B-LSTM.

5.4 Climates Where Wavelet Transforms Improve Performance

The results in Sections 5.1-5.3 demonstrate that, in general, WTs lead to substantial improvements in performance compared to the B-LSTM. To better contextualize the results, improvement in NSE between the optimal W-LSTM and B-LSTM is explored spatially, considering only catchments where B-LSTM achieved an NSE > 0.4.

Considering the one day ahead forecast horizon, Figure 21 demonstrates that catchments where the optimal W-LSTM led to the greatest improvements in NSE (compared to B-LSTM) form geographic clusters. Catchments with the largest improvement in NSE due to using WT-generated features in LSTM tend to be arid (i.e., catchments with infrequent rainfall, dry climate, and low forest cover). This is demonstrated in Figure 22. Whereas Figure 22 shows mean NSE improvement, Figure 23 shows the distribution of NSE improvement across catchments in each Koppen climate region (Kottek, 2006). Figure 24 demonstrates the correlation between improved forecasting performance offered by WTs and catchment aridity (as well as three other related measures).

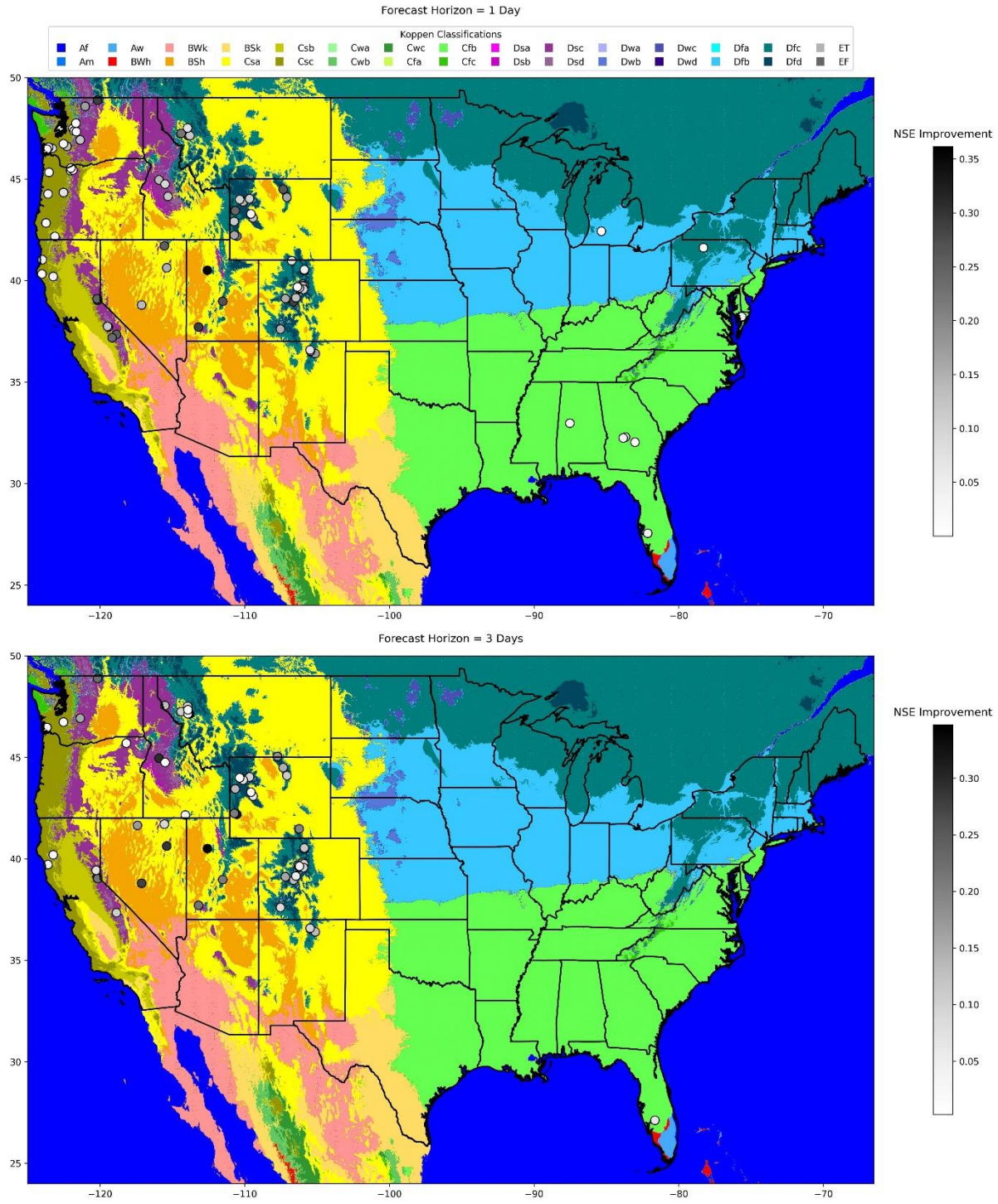


Figure 21. Geographic clustering appears for catchments where optimal W-LSTM showed the greatest absolute increase in NSE relative to B-LSTM (e.g., see Utah and Nevada).

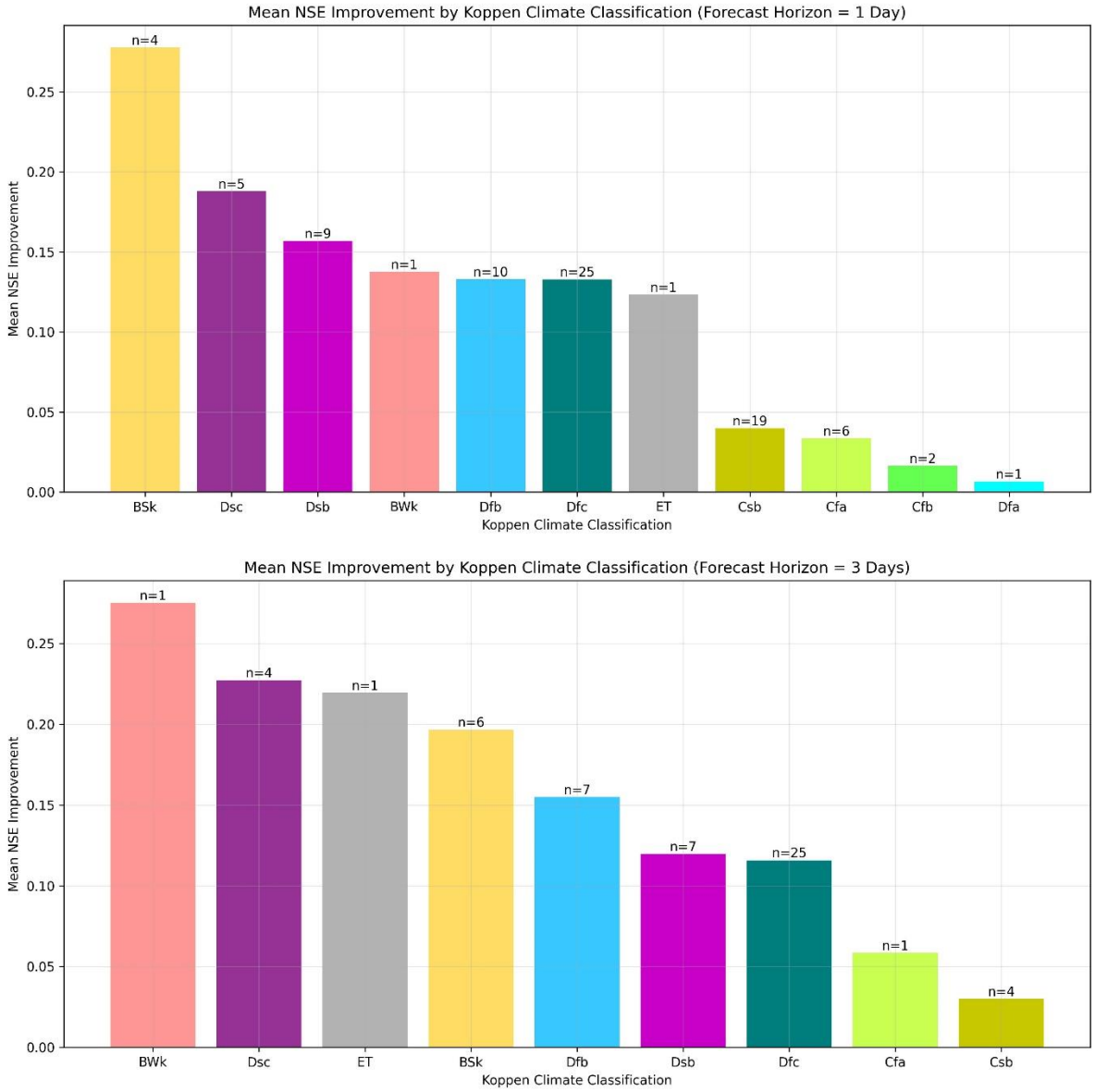


Figure 22. Mean absolute increase in NSE of optimal W-LSTM over B-LSTM across catchments within a particular Koppen climate region. The two bar charts presented here are based on filtered catchments where B-LSTM achieved a test set NSE greater than 0.4.

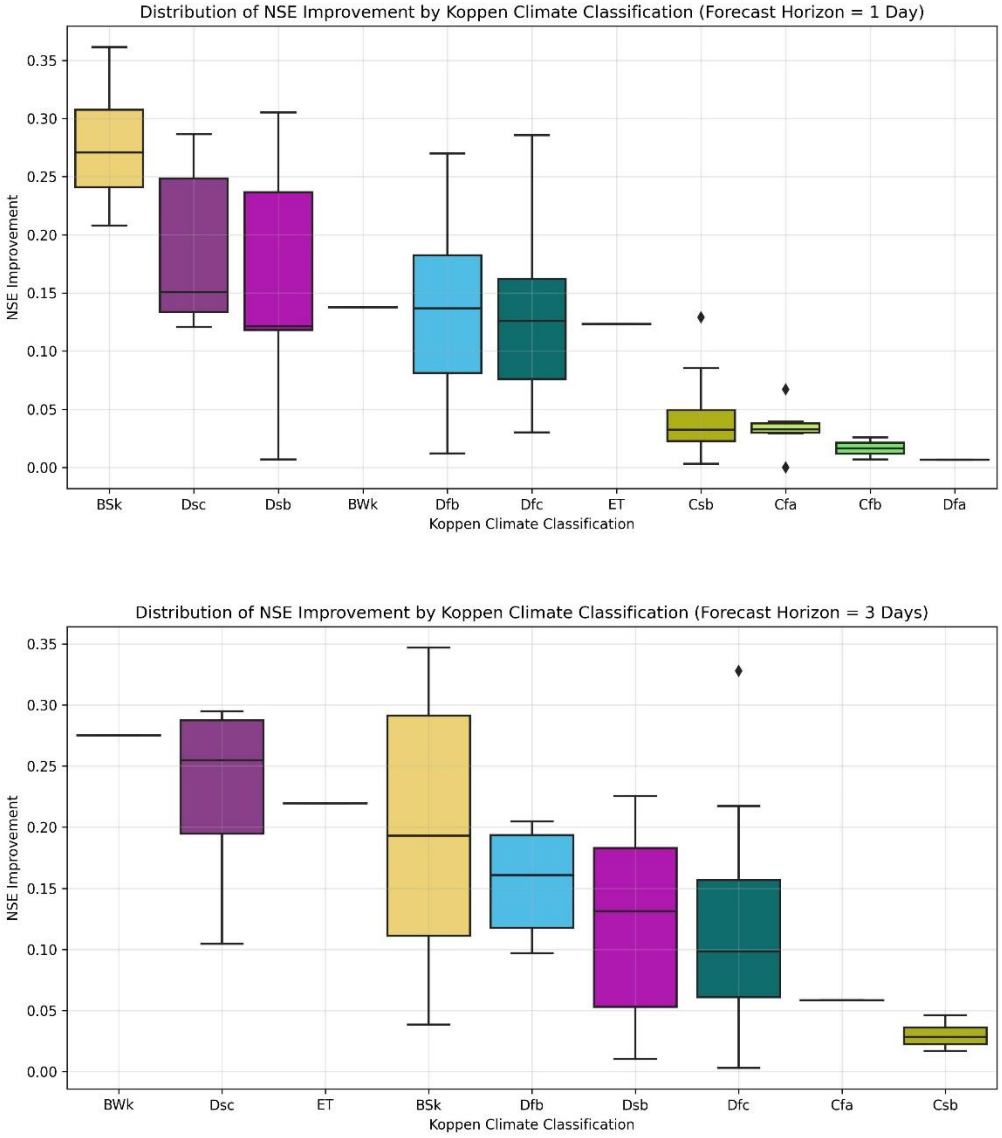


Figure 23. Boxplots of absolute increase in NSE of optimal W-LSTM over B-LSTM across catchments within a particular Koppen climate region. The two charts presented here are based on filtered catchments where B-LSTM achieved a test set NSE greater than 0.4.

In the discussion that follows, any Koppen classification region that includes only one catchment will be excluded. Hence, the Koppen regions BWk (Cold Desert Climate), ET (Tundra Climate), Cfa (Humid Subtropical Climate) and Dfa (Hot Summer Continental Climate) are not discussed below.

The Koppen climate classification system (Peel et al., 2007) is a widely used method for categorizing the world's climates based on measurable criteria, such as air temperature and precipitation patterns. Developed by the German climatologist Wladimir Köppen in the early 20th century (Köppen, 1884)This system classifies climates into five main types, which are further divided into subtypes based on specific criteria, such as seasonal precipitation and air temperature (see Appendix C). The main aim of the Koppen system is to facilitate the study of climatic patterns and their effects on the global distribution of vegetation and ecosystems.

Observing Figure 22, the Koppen climate regions that saw the greatest absolute increase in NSE due to the use of W-LSTM (over B-LSTM) were BSk (Cold Semi-Arid), Dsc (Cold Subarctic with Dry Summers) and Dsb (Warm Summer Mediterranean) (for the one day forecast horizon), and Dsc and BSk (for the three days ahead forecast horizon).

A common characteristic among the three Koppen climate classifications (Bsk, Dsc, and Dsb) where optimal W-LSTM outperformed B-LSTM more substantially than elsewhere is that they all describe climates with limited rainfall and dry summers. In particular, Figure 21 shows remarkable improvements in NSE in the tri-state region of Utah, Nevada, and California, as well as in Wyoming and Colorado. These regions all feature dry and arid climates.

At the one day ahead forecast horizon, Figure 24 shows that there is a close relationship between catchment characteristics relating to aridity and NSE improvement (i.e., the difference in performance between the optimal W-LSTM and B-LSTM) provided by the addition of WT-engineered features. Figure 24 is based on a sample size of 90 catchments - the portion of the 620 CAMELS catchments where B-LSTM achieved NSE > 0.4. At the three day(s) ahead forecast horizon, similar results were achieved as those presented in Figure 24. Thus, these similar results are not repeated here.

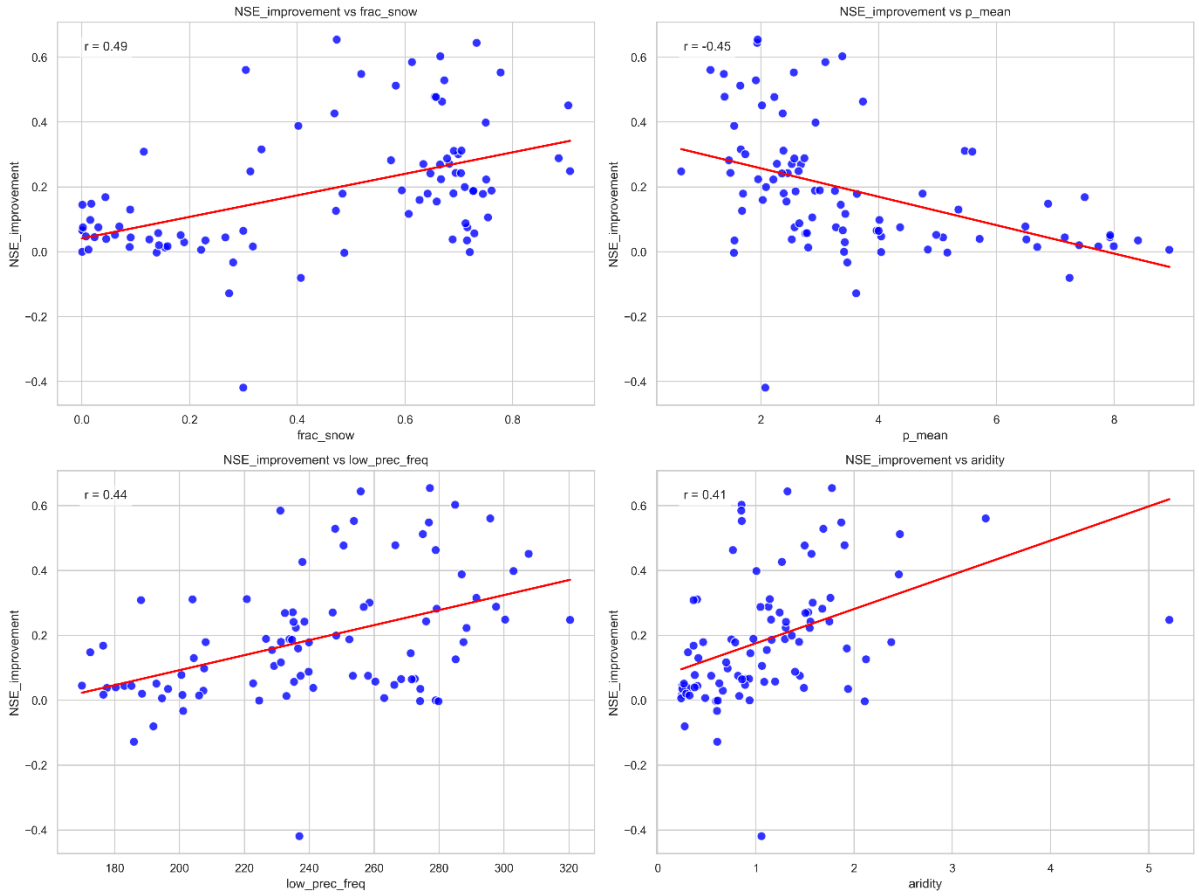


Figure 24. Correlation between catchment attributes and NSE improvement for the optimal W-LSTM compared to the B-LSTM for forecast horizon of one day ahead.

The results in Figure 24 are of practical interest as they may help guide researchers and practitioners when deciding whether including WT-generated features in an ML streamflow forecasting model may be worthwhile. Based on the collected evidence, WT-generated features seem to have the greatest impact on performance when forecasting streamflow in arid catchments. A key recommendation of this thesis is to consider using WT-generated features when forecasting streamflow in arid catchments. Aridity can be determined using the Priestley-Taylor method (Lhomme, 1997), and further characterization of a catchment may be performed using Koppen climate maps (Peel et al., 2007). If the findings from these two tools suggest that a catchment is arid, WT-generated features should be considered when building an ML model for streamflow forecasting.

Most of the plots in Figure 24 are related to precipitation (or a lack thereof). However, the top left plot in Figure 24 shows a relationship between a catchment's mean elevation and NSE improvement due to WT-generated features. A closer examination of this plot provides further justification for the observed relationship between NSE improvement (due to WT-generated features) and a catchment's aridity.

In conjunction with a catchment's aridity, the relationship between the mean elevation of a catchment and NSE improvement can be explained by the so-called "Rain Shadow Effect" in the Western United States (where most of the filtered catchments for the one day ahead forecast horizon are located).

Mountains and high elevations can contribute to the creation of "rain shadows" (Whiteman, 2000) where moist air masses are forced to rise over the mountain. As the air rises, it cools and loses moisture in the form of precipitation on the windward side of the mountain. By the time the air descends on the leeward side, it is dry, creating arid conditions in the high-elevation areas downwind.

The mountain ranges in the Western United States., including the Rockies, Sierra Nevada, and Cascades, exhibit the rain shadow effect prominently. These mountains block moisture from the Pacific Ocean, leading to moist conditions on their windward (western) sides and dry conditions on their leeward (eastern) sides. Thus, high-elevation areas in the interior west, such as parts of the Great Basin and the Colorado Plateau, can be very dry. However, the higher elevations of these mountains themselves can receive substantial snowfall (Siler et al., 2013; Siler & Durran, 2016; Stockham et al., 2018).

The relationship between a catchment's mean elevation and NSE improvement due to WT-generated features can be explained by the relationship between elevation and aridity. Since higher elevations experience greater aridity (and greater aridity is associated with greater improvement in NSE), catchments with higher elevations benefit most from the inclusion of WT-generated features.

This section has uncovered an important finding: WT-generated features seem to improve the NSE of short-term streamflow forecasts most in arid regions. The following sub-section summarizes the key results from Sections 5.1 to 5.4.

5.5 Summary of Experiments

This sub-section summarizes the key results of all experiments from this thesis using the cumulative distribution function (CDF) of the NSE for B-LSTM and the optimal W-LSTM and GS-LSTM models.

By computing the percent increase in NSE due to using optimal W-LSTM instead of B-LSTM for each catchment, the results can be used to plot a distribution of pairwise percent increase in NSE across the catchments. Such a distribution is shown in Figure 25 for the filtered catchments (i.e., where B-LSTM achieved an NSE greater than 0.4) for both one and three day(s) forecast horizons. Figures 26 and 27 show the empirical CDF for the B-LSTM and the W-LSTM models associated with different wavelet filters, as well as the optimal W-LSTM and GS-LSTM models. Figure 26 considers the entire 620 catchments for one and three day(s) ahead forecast horizons. Figure 27 considers the filtered catchments, of which there are 90 and 60 catchments associated with the one and three day(s) ahead forecast horizons, respectively (see Section 4.4). Note that for both Figures 26 and 27, lower NSE values in the negative range were not shown since models with negative NSE are not of interest to practical forecasting applications.

Figures 26 and 27 reveal some interesting results:

1. Overall, the optimal W-LSTM and many W-LSTM models based on different wavelet filters led to higher performance than B-LSTM.
2. The optimal W-LSTM and GS-LSTM models differ at the tails of the distribution. The optimal GS-LSTM had fewer models with low NSE, but the optimal W-LSTM had a higher number of models with high NSE values.

It is important to note that the empirical CDF does not provide a pairwise comparison between configurations (B-LSTM, W-LSTM, optimal W-LSTM, and optimal GS-LSTM). Instead, for each configuration, the empirical CDF provides an ordered ranking of the NSE achieved by that configuration for each catchment. However, as noted in Section 5.2, a pairwise comparison revealed that optimal W-LSTM outperformed the B-LSTM in over 92% of cases, both when considering all 620 catchments and when considering only the filtered catchments.

Evidently, the optimal W-LSTM substantially outperformed B-LSTM for most catchments. In some cases, the optimal W-LSTM led to an improvement in NSE of over 75%. Figure 25 also shows the different wavelet filters that led to the highest percentage increase in NSE when used instead of B-LSTM for the filtered catchments. The db2, db4, fk4, fk6, han2_3, and mb4_2 wavelet filters are also shown to effect an improvement in one and three day(s) ahead streamflow forecasts across the filtered catchments when considering the median percent improvement in NSE.

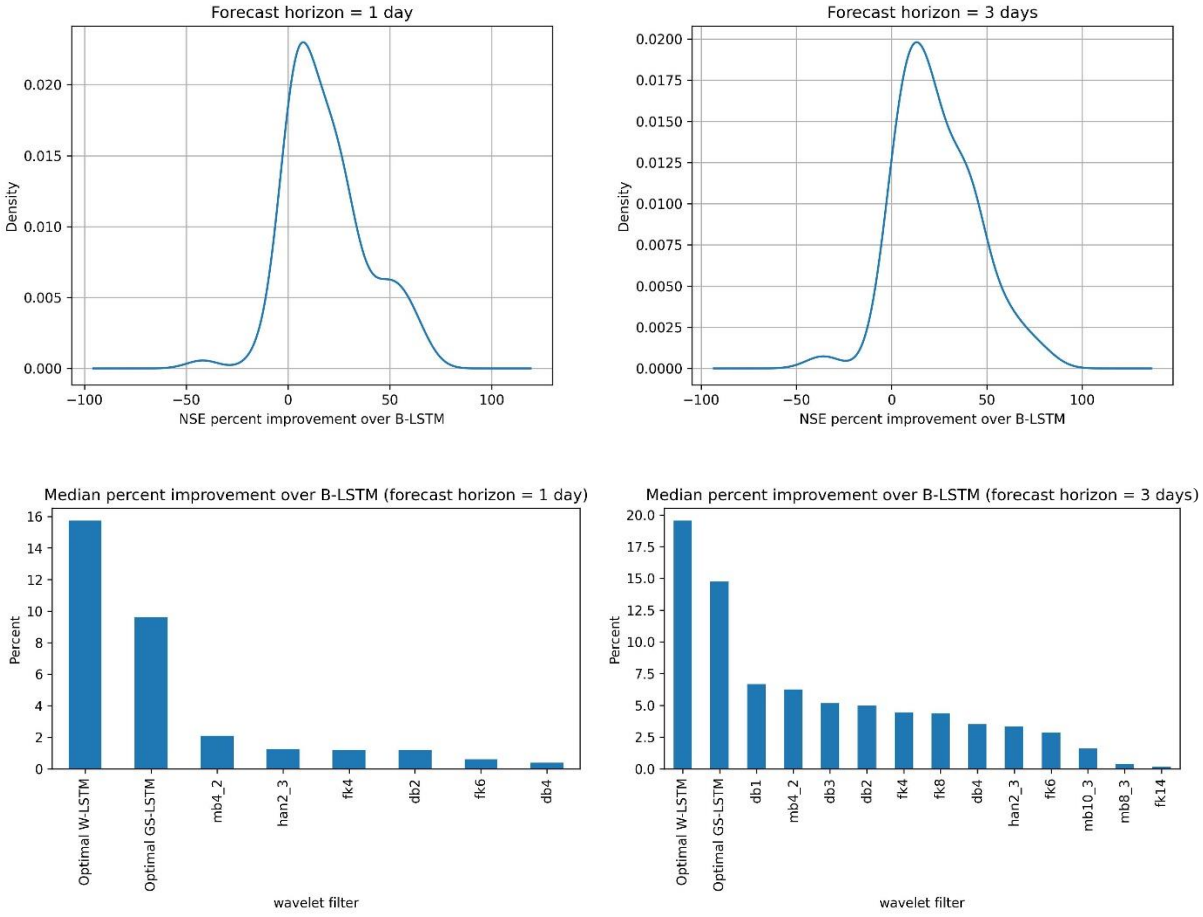


Figure 25. Top row: distribution of percent increase in NSE due to using optimal W-LSTM instead of B-LSTM across filtered catchments (i.e., where B-LSTM achieved an NSE greater than 0.4). Bottom row: Median percent improvement in NSE (across filtered catchments) when using different model configurations instead of B-LSTM.

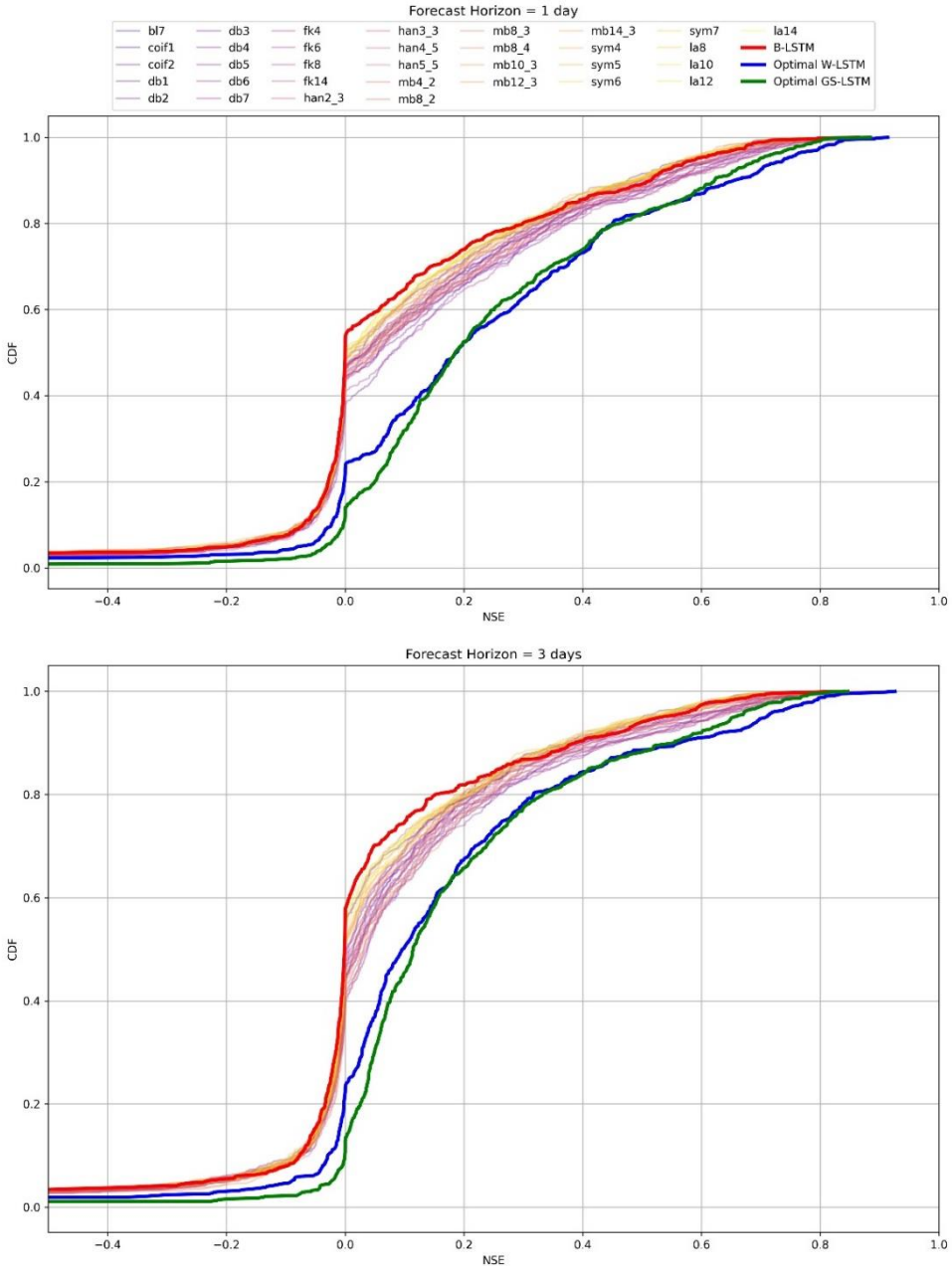


Figure 26. Empirical cumulative distribution function (CDF) plots for one and three day(s) ahead forecast horizons considering all catchments.

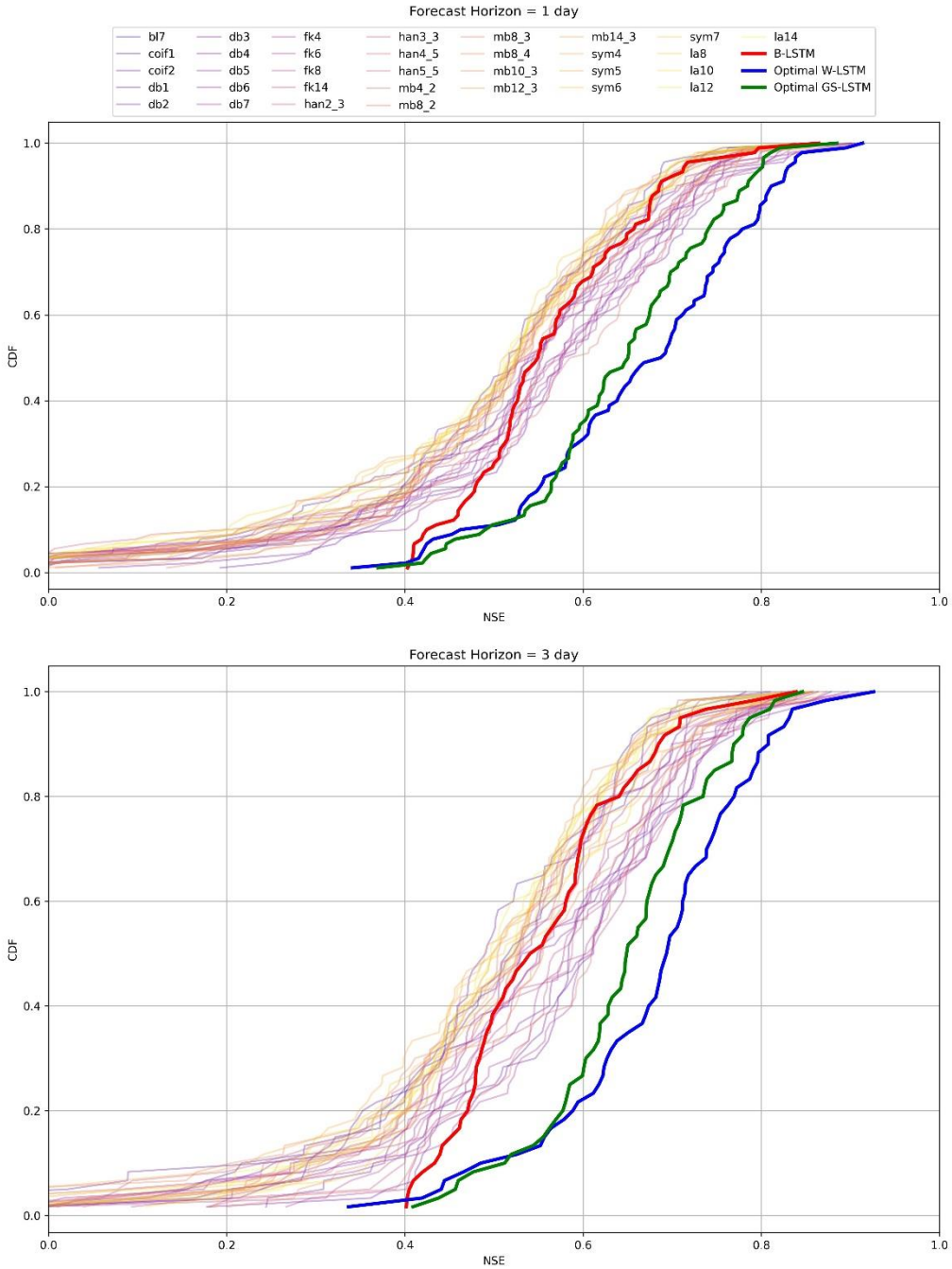


Figure 27 - Empirical cumulative distribution function (CDF) plots for one- and three day(s) ahead forecast horizons considering filtered catchments (90 and 60 catchments for one and three day(s) ahead forecast horizons).

Lastly, WT-generated features improved streamflow forecasting performance (as measured by NSE) most for catchments in dry climates. Figure 24 confirms this finding by showing a high correlation between improvements in NSE due to the use of optimal W-LSTM instead of B-LSTM and different catchment attributes (e.g., aridity, low precipitation frequency). Additionally, results from Section 5.4 indicate that the drier Koppen climate classifications B (arid climates) and D (snowy climates) benefit most from the use of WT-generated features.

Although it is infeasible to provide the hydrographs for all 620 catchments considered in this thesis, the following section provides a sample of hydrographs for catchments where optimal W-LSTM provided notable improvements in NSE over B-LSTM.

5.6 Hydrographs for Some Notable Catchments

The plots thus far have been useful for making general assessments on the quality of the B-LSTM, W-LSTM, and GS-LSTM configurations for large samples (the 620 catchments and the filtered sets associated with the one and three day(s) ahead forecast horizons). However, to judge the quality of the forecasts offered by a candidate model (e.g., W-LSTM) with respect to a baseline (e.g., B-LSTM), it is often useful to see how well the forecasts track the target (e.g., observed streamflow) through time. Hydrographs fulfill this task very well. Thus, Figures 28, 29, and 30 are provided to show the potential of WTs for improving the quality of streamflow forecasts. Each hydrograph is based on the test period, reflective of the out-of-sample performance that one could expect when using W-LSTM (or B-LSTM) in real-world applications. Example hydrographs for additional CAMELS catchments (where W-LSTM led to substantial improvements in forecast quality over B-LSTM) is provided in Appendix D.

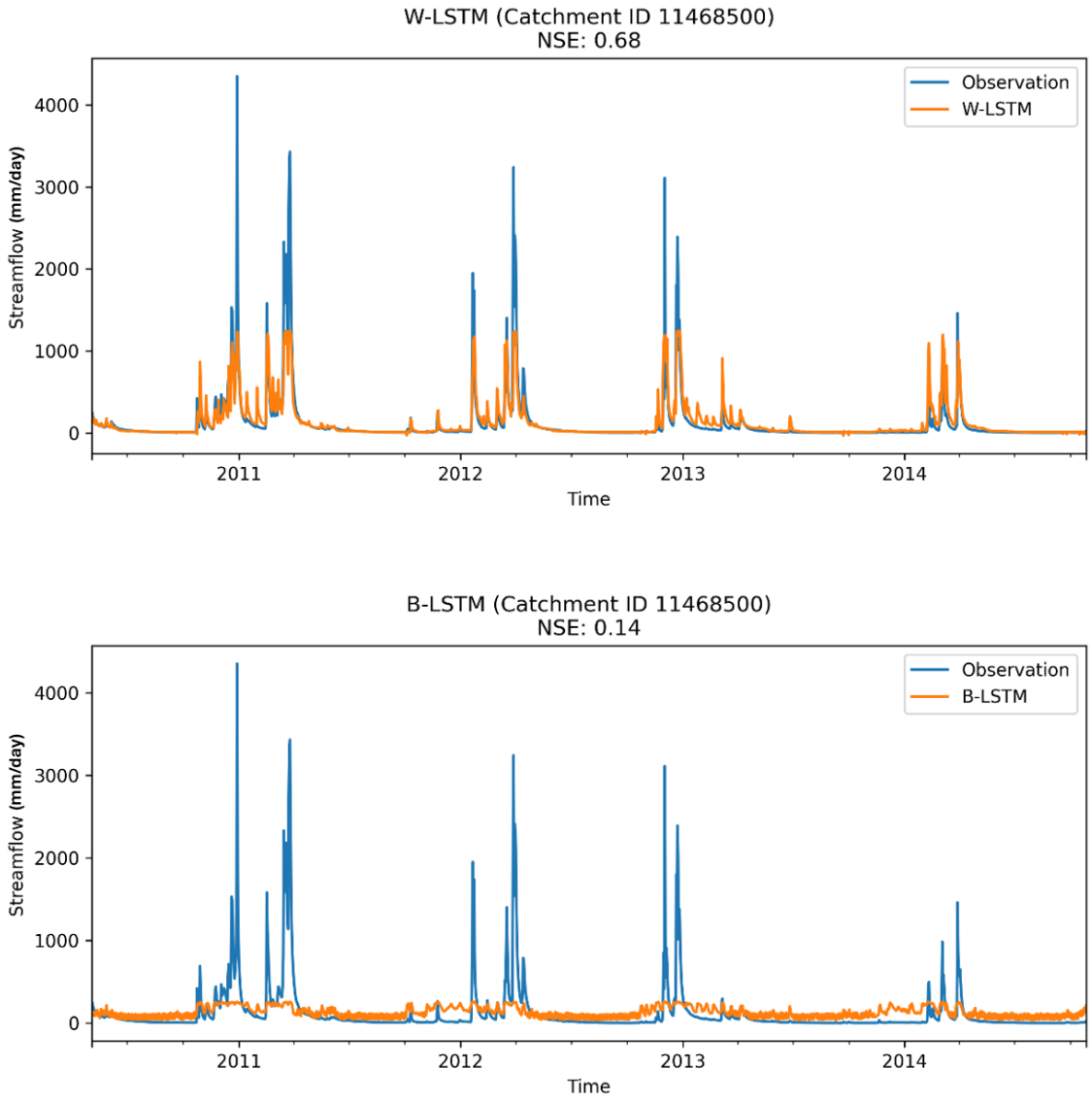


Figure 28. Comparison of optimal W-LSTM (top) to B-LSTM (bottom) for one day ahead streamflow forecasting of Noyo River near Fort Bragg, California. The blue line shows observed streamflow, while the orange lines show the forecasts.

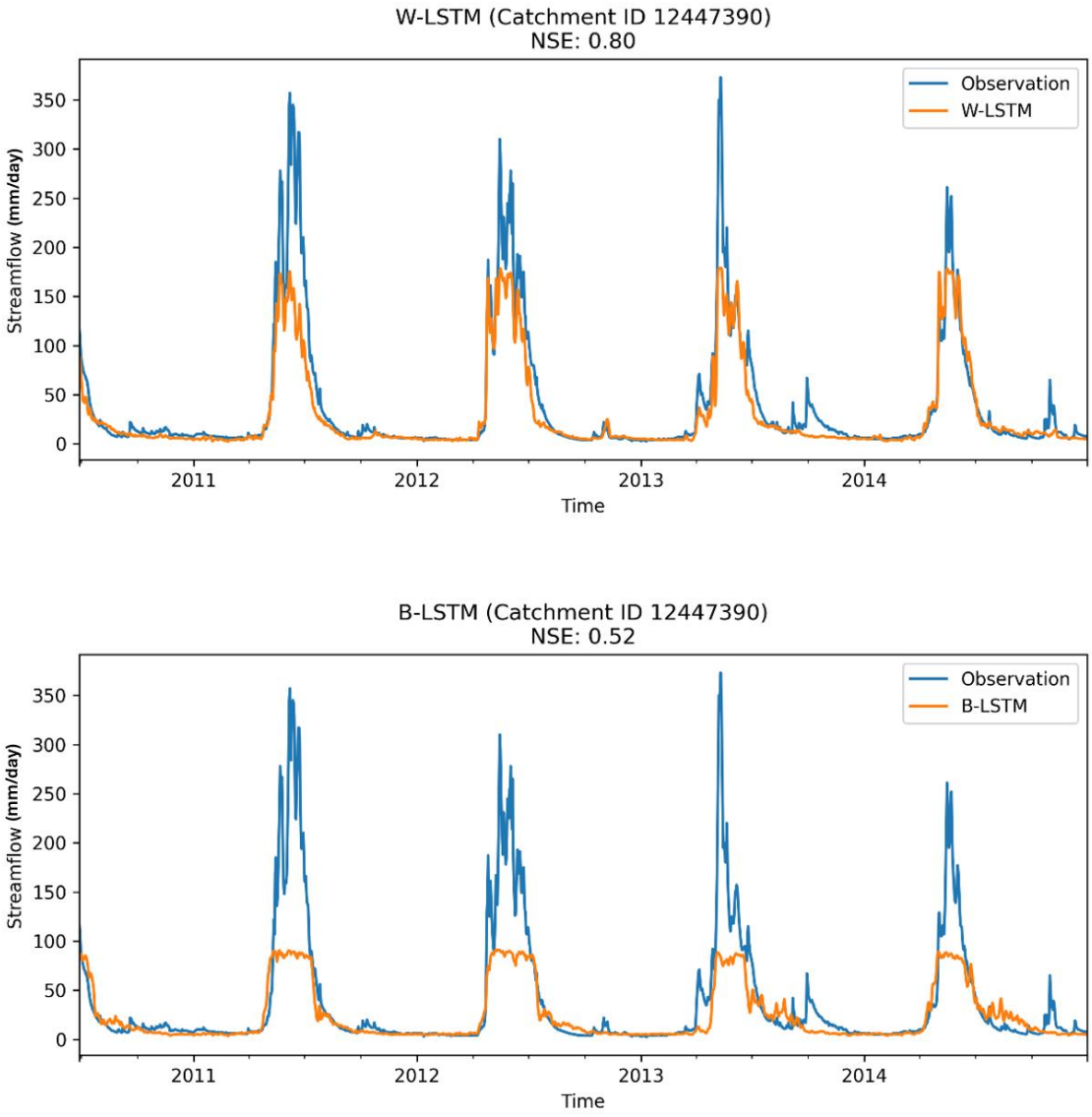


Figure 29. Comparison of optimal W-LSTM (top) to B-LSTM (bottom) for one day ahead streamflow forecasting of Andrews Creek near Mazama, Washington. The blue line shows observed streamflow, while the orange lines show the forecasts.

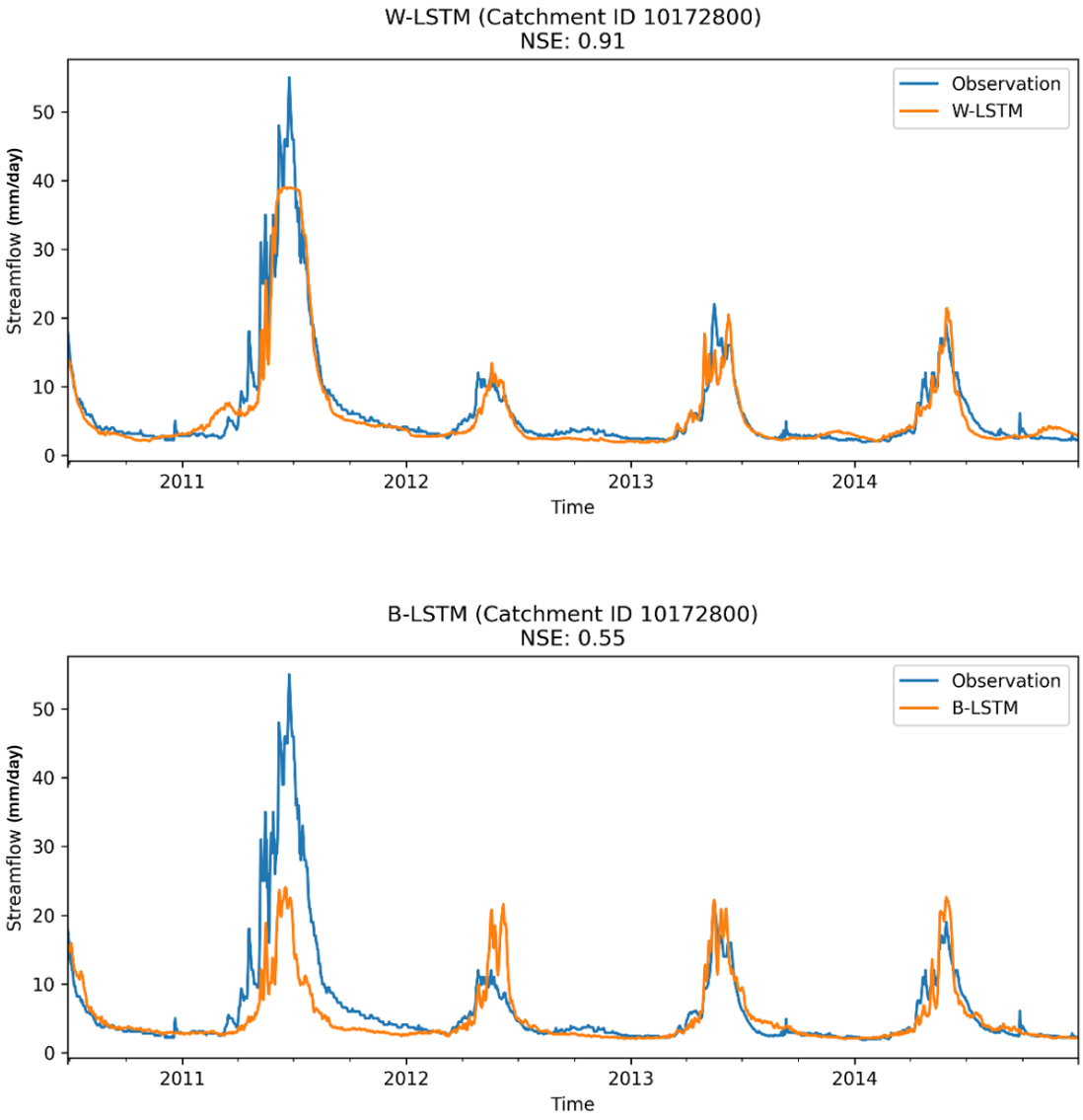


Figure 30. Comparison of optimal W-LSTM (top) to B-LSTM (bottom) for one day ahead streamflow forecasting of South Willow Creek near Grantsville, Utah. The blue line shows observed streamflow, while the orange lines show the forecasts.

The most important finding among the hydrographs shown in Figures 28, 29 and 30 is that streamflow peaks are better captured when WT-generated features are incorporated. This is a very important finding since planning water resources is often complicated by the difficulty of forecasting peak streamflow. Furthermore, spikes in streamflow can lead to flooding events. Thus, accurate

forecasts of peak streamflow events can help mitigate water-induced damage as well as potentially save property and lives (Ghobadi & Kang, 2022; Hunt et al., 2022).

This chapter has accomplished the main goal of this thesis: demonstrating, quantitatively and qualitatively, that WT-generated features can improve short-term (one and three day(s) ahead) streamflow forecasts using a large-sample case study. The final chapter completes this thesis by summarizing the main aspects of this research, providing its main conclusions, and recommending some future research directions.

Chapter 6 Summary, Conclusions, and Recommendations

This chapter begins by highlighting the main research problem this thesis sought to address, along with a summary of the experiments used to address the research problem and their main results. Afterwards, the conclusions from this research are stated before ending with a list of recommendations for future research.

6.1 Summary

6.1.1 Research Problem and Goal

WTs are common feature engineering tools used in hydrology that have been shown to improve the performance of ML-based streamflow forecasting models. However, there are no large-sample studies that compare the performance of ML-based streamflow forecasting models with and without WT-generated features. Most studies using WTs for streamflow forecasting only test their models on one to four sites. Thus, it is unclear if the reported improvements in streamflow forecasting models are due to the use of WTs or fortuitous site-specific artifacts. The goal of this thesis is to address this important gap in the streamflow forecasting literature. This thesis sought to address this gap by assessing the performance of short-term (one and three day(s) ahead) streamflow forecasting models on an open-source large-sample hydrology dataset, known as CAMELS.

Overall, 620 catchments from the CAMELS dataset were used to build the streamflow forecasting models. The models were built using LSTM due to its popularity in hydrology. Three different LSTM model configurations were considered to enable a large-sample assessment of improvements in streamflow forecasting performance due to WTs. The three different configurations included models without WT-generated features (baseline LSTM, or B-LSTM), with WT-generated features (WT-LSTM), and without WT-generated features (but including hyper-parameter (architecture) selection via grid search (GS-LSTM)).

The first configuration (B-LSTM) adopted the LSTM architecture since it has been shown in the literature to perform well for streamflow simulation for the CAMELS dataset (see Section 4.1.1). The second configuration (W-LSTM) used the same architecture as B-LSTM but included WT-generated features as input to the model (along with the features adopted in B-LSTM; see Section 4.1.2). The third configuration used the same features as B-LSTM but considered different LSTM

architectures (see Section 4.1.3). The third configuration was used to assess whether a modeler is better off spending time on a different strategy for improving streamflow forecasting performance instead of pursuing WTs.

6.1.2 Experiments

Three different experiments, each related to a specific objective, were undertaken. First, the performance of W-LSTM using different wavelet filters was compared against B-LSTM for each catchment. Second, the best-performing wavelet filter for each catchment (optimal W-LSTM) was compared against the best-performing LSTM architecture (optimal GS-LSTM) for each catchment. Third, the performance of optimal W-LSTM and B-LSTM for different Koppen climate regions were compared to identify any geographic clusters where WT-generated features improve streamflow forecasting the most. In these experiments, the three configurations were compared across all 620 catchments as well as a subset of catchments, referred to as filtered catchments. The filtered catchments include those where B-LSTM achieved an NSE greater than 0.4 (90 and 60 catchments for one and three day(s) ahead forecast horizons, respectively). This threshold was taken as an indication that a B-LSTM streamflow forecasting model had satisfactory performance. In other words, the threshold was used to identify B-LSTM models that may be useful in practice.

6.1.3 Main Findings

The goal of this thesis was achieved, showing that, in general, WT-generated features can be used to improve short-term (one and three days(s) ahead) streamflow forecasting performance on the CAMELS dataset. Notably, optimal W-LSTM outperformed B-LSTM in over 92% of the studied catchments for both one and three day(s) ahead forecasting horizons. In some cases, W-LSTM outperformed B-LSTM by over 75% when considering the NSE metric.

By accomplishing the first objective of this thesis, it was found that wavelet filters db2, db4 (Daubechies), fk4, fk6 (Fejér-Korovkin), han2_3 (Han linear-phase moments), and mb4_2 (Morris minimum-bandwidth) were identified as effective in enhancing streamflow forecasting performance across the filtered catchments and both forecast horizons. This result underscores the importance of selecting appropriate wavelet filters for each catchment, which requires the modeler to expend additional computational costs when using WT-generated features.

The optimal W-LSTM models demonstrated a substantial improvement in forecast performance compared to B-LSTM. Specifically, when undertaking a pairwise comparison across all catchments, the optimal W-LSTM models achieved a 60% and 89% increase in median NSE over the B-LSTM for one and three day(s) ahead forecast horizons, respectively. Similarly, for the filtered catchments (i.e., where B-LSTM achieved an NSE > 0.4), the improvements were 16% and 20%, respectively. These results highlight the remarkable advantage of incorporating WT-generated features in LSTM models for short-term streamflow forecasting across the CAMELS dataset.

It was found that the optimal W-LSTM and GS-LSTM models outperformed one another at about the same rate across all catchments for the one day ahead forecasting horizon, but at the three days ahead forecasting horizon, GS-LSTM outperformed W-LSTM. However, when considering the filtered catchments (i.e., catchments where B-LSTM achieved NSE > 0.4), W-LSTM outperformed GS-LSTM in 60% and 70% of the catchment for one and three day(s) ahead forecasting horizons, respectively. GS-LSTM tended to outperform W-LSTM in catchments where the B-LSTM achieved an NSE < 0.4, opposite the finding when B-LSTM achieved an NSE > 0.4. This is a very important finding as it indicates that combining WT-generated features with GS-LSTM is likely to lead to models that will outperform B-LSTM as well as optimal W-LSTM and GS-LSTM.

Furthermore, this research found that the LSTM-based streamflow forecasting models for catchments with arid (Koppen climate classification B) and snowy (Koppen climate classification D) climates benefitted most from the use of WT-generated features. Furthermore, high linear correlation coefficients were found between different catchment attributes (e.g., catchment mean elevation) and the improvement in NSE due to the use of optimal W-LSTM instead of B-LSTM. It was found that several catchment attributes (e.g., mean elevation, faction of snow, mean daily precipitation) exhibited high linear correlation with NSE improvement due to the use of WT-generated features (Section 5.4), providing insight into the relationship between W-LSTM performance and different catchment attributes (or geographical factors).

6.2 Conclusions

In conclusion, the integration of WT-generated features into LSTM models represents a promising advancement for short-term streamflow forecasting. Results from the CAMELS dataset demonstrates that substantial improvements in forecasting performance can be achieved across a wide range of catchments when WT-generated features are used in LSTM models. The largest improvements

in streamflow forecasting performance due to WT-generated features tend to be catchments in arid and snowy climates. Despite the increased computational requirements needed to identify optimal wavelet filters for each catchment, the benefits of enhanced forecasting performance are clear. The findings presented in this thesis are novel, as there have been no large-sample studies assessing the impact of WT-generated features on streamflow forecasting performance. Thus, the results from this thesis not only advance the current understanding of how WTs impact streamflow forecasting models but also open avenues for future research aimed at optimizing predictive models for better management of water resources in various hydroclimatological conditions.

6.3 Recommendations

To build upon the new knowledge uncovered in this thesis, there are several avenues that future research can explore:

1. Explore a similar experimental setup adopted in this thesis for the Caravan dataset (Kratzert et al., 2023). This would extend the results of this thesis from the CONUS across the globe, revealing which wavelet filters perform best on which catchments. At the same time, such experimentation would uncover a richer understanding of catchment attributes that are linked to improvements in streamflow forecasting performance due to the use of WT-generated features.
2. Using the results from this thesis and point 1, build a single global scale LSTM model for streamflow forecasting, using catchment attributes and the optimal wavelet filter for each catchment to derive catchment-specific features for the model.
3. Modify the LSTM architecture to account for uncertainty in the model forecasts to enable accurate and reliable streamflow forecasts. This can be done using custom probabilistic output layers (Klotz et al., 2022) or different deep learning formulations, such as conditional variational autoencoders (Jahangir & Quilty, 2024).

By following the above recommendations, a probabilistic model can be developed for accurate and reliable short-term streamflow forecasting across the globe that can compete with the current state-of-the-art (Nearing et al., 2024), which currently does not take advantage of WT-generated features.

References

Abbasi, M., Farokhnia, A., Bahreinimotlagh, M., & Roozbahani, R. (2021). A hybrid of Random Forest and Deep Auto-Encoder with support vector regression methods for accuracy improvement and uncertainty reduction of long-term streamflow prediction. *Journal of Hydrology*, *597*, 125717. https://doi.org/10.1016/j.jhydrol.2020.125717

Abrahart, R. J., & See, L. (2000). Comparing neural network and autoregressive moving average techniques for the provision of continuous river flow forecasts in two contrasting catchments. *Hydrological Processes*, *14*(11–12), 2157–2172. https://doi.org/10.1002/1099-1085(20000815/30)14:11/12<2157::AID-HYP57>3.0.CO;2-S

Addor, N., Do, H. X., Alvarez-Garreton, C., Coxon, G., Fowler, K., & Mendoza, P. A. (2020). Large-sample hydrology: Recent progress, guidelines for new datasets and grand challenges. *Hydrological Sciences Journal*, *65*(5), 712–725. https://doi.org/10.1080/02626667.2019.1683182

Adib, A., Zaerpour, A., & Lotfirad, M. (2021). On the reliability of a novel MODWT-based hybrid ARIMA-artificial intelligence approach to forecast daily Snow Depth (Case study: The western part of the Rocky Mountains in the U.S.A). *Cold Regions Science and Technology*, *189*, 103342. https://doi.org/10.1016/j.coldregions.2021.103342

Adnan, R. M., Jaafari, A., Mohanavelu, A., Kisi, O., & Elbeltagi, A. (2021). Novel Ensemble Forecasting of Streamflow Using Locally Weighted Learning Algorithm. *Sustainability*, *13*(11). https://doi.org/10.3390/su13115877

Adnan, R. M., Mostafa, R. R., Dai, H.-L., Heddam, S., Masood, A., & Kisi, O. (2023). Enhancing accuracy of extreme learning machine in predicting river flow using improved reptile search

algorithm. *Stochastic Environmental Research and Risk Assessment*, *37*(8), 3063–3083. https://doi.org/10.1007/s00477-023-02435-y

Adnan, R. M., R. Mostafa, R., Kisi, O., Yaseen, Z. M., Shahid, S., & Zounemat-Kermani, M. (2021). Improving streamflow prediction using a new hybrid ELM model combined with hybrid particle swarm optimization and grey wolf optimization. *Knowledge-Based Systems*, *230*, 107379. https://doi.org/10.1016/j.knosys.2021.107379

Adnan, R., Yuan, X., Kisi, O., & Curtef, V. (2017). Application of Time Series Models for Streamflow Forecasting. *Civil and Environmental Research*. https://www.semanticscholar.org/paper/Application-of-Time-Series-Models-for-Streamflow-Adnan-Yuan/4815d81de0ec064e1820ffde6cbe9b0170c87e3a

Agarwal, S., Roy, P., Choudhury, P., & Debbarma, N. (2022). Comparative study on stream flow prediction using the GMNN and wavelet-based GMNN. *Journal of Water and Climate Change*, *13*(9), 3323–3337. https://doi.org/10.2166/wcc.2022.226

Alonso Brito, G. R., Rivero Villaverde, A., Lau Quan, A., & Ruíz Pérez, M. E. (2021). Comparison between SARIMA and Holt–Winters models for forecasting monthly streamflow in the western region of Cuba. *SN Applied Sciences*, *3*(6), 671. https://doi.org/10.1007/s42452-021-04667-5

Alquraish, M. M., & Khadr, M. (2021). Remote-Sensing-Based Streamflow Forecasting Using Artificial Neural Network and Support Vector Machine Models. *Remote Sensing*, *13*(20), Article 20. https://doi.org/10.3390/rs13204147

Amini, A., Dolatshahi, M., & Kerachian, R. (2024). Real-time rainfall and runoff prediction by integrating BC-MODWT and automatically-tuned DNNs: Comparing different deep learning models. *Journal of Hydrology*, *631*, 130804. https://doi.org/10.1016/j.jhydrol.2024.130804

Amiri, E. (2015). Forecasting daily river flows using nonlinear time series models. *Journal of Hydrology*, *527*, 1054–1072. https://doi.org/10.1016/j.jhydrol.2015.05.048

Anning, D. W., & Parker, J. T. C. (2009). *Predictive models of the hydrological regime of unregulated streams in Arizona* (2009th–1269th ed., U.S. Geological Survey Open-File Report, p. 33). https://pubs.usgs.gov/of/2009/1269/

Apaydin, H., Feizi, H., Sattari, M. T., Colak, M. S., Shamshirband, S., & Chau, K.-W. (2020). Comparative Analysis of Recurrent Neural Network Architectures for Reservoir Inflow Forecasting. *Water*, *12*(5), Article 5. https://doi.org/10.3390/w12051500

Aristizabal, F., Salas, F., Petrochenkov, G., Grout, T., Avant, B., Bates, B., Spies, R., Chadwick, N., Wills, Z., & Judge, J. (2023). Extending Height Above Nearest Drainage to Model Multiple Fluvial Sources in Flood Inundation Mapping Applications for the U.S. National Water Model. *Water Resources Research*, *59*(5), e2022WR032039. https://doi.org/10.1029/2022WR032039

Arsenault, R., Martel, J.-L., Brunet, F., Brissette, F., & Mai, J. (2023). Continuous streamflow prediction in ungauged basins: Long short-term memory neural networks clearly outperform traditional hydrological models. *Hydrology and Earth System Sciences*, *27*(1), 139–157. https://doi.org/10.5194/hess-27-139-2023

ASCE Task Committee on Application of Artificial Neural Networks in Hydrology. (2000). Artificial Neural Networks in Hydrology. II: Hydrologic Applications. *Journal of Hydrologic Engineering*, *5*(2), 124–137. https://doi.org/10.1061/(ASCE)1084-0699(2000)5:2(124)

Aussem, A., Campbell, J., & Murtagh, F. (1998). Wavelet-based feature extraction and decomposition strategies for financial forecasting. *Journal of Computational Intelligence in Finance*, *6*(2), 5–12.

Azarpira, F., & Shahabi, S. (2021). Evaluating the capability of hybrid data-driven approaches to forecast monthly streamflow using hydrometric and meteorological variables. *Journal of Hydroinformatics*, *23*(6), 1165–1181. https://doi.org/10.2166/hydro.2021.105

Barzegar, R., Aalami, M. T., & Adamowski, J. (2021). Coupling a hybrid CNN-LSTM deep learning model with a Boundary Corrected Maximal Overlap Discrete Wavelet Transform for multiscale Lake water level forecasting. *Journal of Hydrology*, *598*, 126196. https://doi.org/10.1016/j.jhydrol.2021.126196

Barzegar, R., Asghari Moghaddam, A., Adamowski, J., & Ozga-Zielinski, B. (2018). Multi-step water quality forecasting using a boosting ensemble multi-wavelet extreme learning machine model. *Stochastic Environmental Research and Risk Assessment*, *32*(3), 799–813. https://doi.org/10.1007/s00477-017-1394-z

Bašta, M. (2014). Additive Decomposition and Boundary Conditions in Wavelet-Based Forecasting Approaches. *Acta Oeconomica Pragensia*, *2014*(2), 48–70. https://ideas.repec.org//a/prg/jnlaop/v2014y2014i2id431p48-70.html

Bengio, Y., Simard, P., & Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, *5*(2), 157–166. IEEE Transactions on Neural Networks. https://doi.org/10.1109/72.279181

Bergstra, J., & Bengio, Y. (2012). *Random Search for Hyper-Parameter Optimization*.

Booras, K., McIntyre, A. R., Joshua Weiss, W., Howells, C., & Palmer, R. N. (2018). Incorporating Streamflow Forecasts with Aggregate Drought Indices for the Management of Water Supply. *Journal of Water Resources Planning and Management*, *144*(1), 04017078. https://doi.org/10.1061/(ASCE)WR.1943-5452.0000849

Box, G. E. P., & Jenkins, G. M. (1976). *Time Series Analysis: Forecasting and Control*. Holden-Day.

Breiman, L. (2001). Random Forests. *Machine Learning*, *45*(1), 5–32. https://doi.org/10.1023/A:1010933404324

Brigode, P., Oudin, L., & Perrin, C. (2013). Hydrological model parameter instability: A source of additional uncertainty in estimating the hydrological impacts of climate change? *Journal of Hydrology*, *476*, 410–425. https://doi.org/10.1016/j.jhydrol.2012.11.012

Carlson, R. F., MacCormick, A. J. A., & Watts, D. G. (1970). Application of Linear Random Models to Four Annual Streamflow Series. *Water Resources Research*, *6*(4), 1070–1078. https://doi.org/10.1029/WR006i004p01070

Cassagnole, M., Ramos, M.-H., Zalachori, I., Thirel, G., Garçon, R., Gailhard, J., & Ouillon, T. (2021). Impact of the quality of hydrological forecasts on the management and revenue of hydroelectric reservoirs – a conceptual approach. *Hydrology and Earth System Sciences*, *25*(2), 1033–1052. https://doi.org/10.5194/hess-25-1033-2021

Castangia, M., Grajales, L. M. M., Aliberti, A., Rossi, C., Macii, A., Macii, E., & Patti, E. (2023). Transformer neural networks for interpretable flood forecasting. *Environmental Modelling & Software*, *160*, 105581. https://doi.org/10.1016/j.envsoft.2022.105581

Castellano-Méndez, M., González-Manteiga, W., Febrero-Bande, M., Manuel Prada-Sánchez, J., & Lozano-Calderón, R. (2004). Modelling of the monthly and daily behaviour of the runoff of the Xallas river using Box–Jenkins and neural networks methods. *Journal of Hydrology*, *296*(1), 38–58. https://doi.org/10.1016/j.jhydrol.2004.03.011

Castelletti, A., Bertoni, F., Giuliani, M., & Reed, P. (2020). *Informed water infrastructure design: Improving coupled dam sizing and operation by streamflow forecasts*. 15835. EGU General Assembly Conference Abstracts. https://doi.org/10.5194/egusphere-egu2020-15835

Chandrashekar, G., & Sahin, F. (2014). A survey on feature selection methods. *Computers & Electrical Engineering*, *40*(1), 16–28. https://doi.org/10.1016/j.compeleceng.2013.11.024

Chang, F.-J., & Guo, S. (2020). Advances in Hydrologic Forecasts and Water Resources Management. *Water*, *12*(6), Article 6. https://doi.org/10.3390/w12061819

Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785–794. https://doi.org/10.1145/2939672.2939785

Chidepudi, S. K. R., Massei, N., Jardani, A., Henriot, A., Allier, D., & Baulon, L. (2023). A wavelet-assisted deep learning approach for simulating groundwater levels affected by low-frequency variability. *Science of The Total Environment*, *865*, 161035. https://doi.org/10.1016/j.scitotenv.2022.161035

Chiew, F. (2010). Lumped Conceptual Rainfall-Runoff Models and Simple Water Balance Methods: Overview and Applications in Ungauged and Data Limited Regions. *Geography Compass*. https://doi.org/10.1111/j.1749-8198.2009.00318.x

Chollet, F. (2015). *Keras*. https://github.com/fchollet/keras

Chong, K. L., Lai, S. H., & El-Shafie, A. (2019). Wavelet Transform Based Method for River Stream Flow Time Series Frequency Analysis and Assessment in Tropical Environment. *Water Resources Management*, *33*(6), 2015–2032. https://doi.org/10.1007/s11269-019-02226-7

Clark, M. P., & Hay, L. E. (2004). Use of Medium-Range Numerical Weather Prediction Model Output to Produce Forecasts of Streamflow. *Journal of Hydrometeorology*, *5*(1), 15–32. https://doi.org/10.1175/1525-7541(2004)005<0015:UOMNWP>2.0.CO;2

Coulibaly, P. (2010). Reservoir Computing approach to Great Lakes water level forecasting. *Journal of Hydrology*, *381*(1), 76–88. https://doi.org/10.1016/j.jhydrol.2009.11.027

Cover, T. M., & Thomas, J. A. (1991). *Elements of Information Theory*.

Cristianini, N., & Ricci, E. (2008). Support Vector Machines. In M.-Y. Kao (Ed.), *Encyclopedia of Algorithms* (pp. 928–932). Springer US. https://doi.org/10.1007/978-0-387-30162-4_415

Dariane, A. B., & M. Behbahani, M. R. (2024). Maximum energy entropy: A novel signal preprocessing approach for data-driven monthly streamflow forecasting. *Ecological Informatics, 79*, 102452. https://doi.org/10.1016/j.ecoinf.2023.102452

Davison, J. H., Hwang, H.-T., Sudicky, E. A., Mallia, D. V., & Lin, J. C. (2018). Full Coupling Between the Atmosphere, Surface, and Subsurface for Integrated Hydrologic Simulation. *Journal of Advances in Modeling Earth Systems*, *10*(1), 43–53. https://doi.org/10.1002/2017MS001052

Day, G. N. (1985). Extended Streamflow Forecasting Using NWSRFS. *Journal of Water Resources Planning and Management*, *111*(2), 157–170. https://doi.org/10.1061/(ASCE)0733-9496(1985)111:2(157)

Dehghani, A., Moazam, H. M. Z. H., Mortazavizadeh, F., Ranjbar, V., Mirzaei, M., Mortezavi, S., Ng, J. L., & Dehghani, A. (2023). Comparative evaluation of LSTM, CNN, and ConvLSTM for hourly short-term streamflow forecasting using deep learning approaches. *Ecological Informatics, 75*, 102119. https://doi.org/10.1016/j.ecoinf.2023.102119

Difi, S., Elmeddahi, Y., Hebal, A., Singh, V. P., Heddam, S., Kim, S., & Kisi, O. (2023). Monthly streamflow prediction using hybrid extreme learning machine optimized by bat algorithm: A case study of Cheliff watershed, Algeria. *Hydrological Sciences Journal*, *68*(2), 189–208. https://doi.org/10.1080/02626667.2022.2149334

Drisya, J., Kumar, D. S., & Roshni, T. (2021). Hydrological drought assessment through streamflow forecasting using wavelet enabled artificial neural networks. *Environment, Development and Sustainability*, *23*(3), 3653–3672. https://doi.org/10.1007/s10668-020-00737-7

Edossa, D. C., & Babel, M. S. (2011). Application of ANN-Based Streamflow Forecasting Model for Agricultural Water Management in the Awash River Basin, Ethiopia. *Water Resources Management*, *25*(6), 1759–1773. https://doi.org/10.1007/s11269-010-9773-y

Fan, Y. R. (2020). Uncertainty Quantification in Hydrologic Predictions: A Brief Review. *JOURNAL OF ENVIRONMENTAL INFORMATICS LETTERS*, *2*(2), Article 2. http://www.jeiletters.org/index.php?journal=mys&page=article&op=view&path%5B%5D=201900019

Fan, Y. R., Huang, G. H., Li, Y. P., Wang, X. Q., Jin, L., & Li, Z. (2016). Development of PCA-based cluster quantile regression (PCA-CQR) framework for streamflow prediction: Application to the Xiangxi river watershed, China. *Applied Soft Computing*, *51*. https://doi.org/10.1016/j.asoc.2016.11.039

Fang, W., Huang, S., Ren, K., Huang, Q., Huang, G., Cheng, G., & Li, K. (2019). Examining the applicability of different sampling techniques in the development of decomposition-based streamflow forecasting models. *Journal of Hydrology*, *568*, 534–550. https://doi.org/10.1016/j.jhydrol.2018.11.020

Feng, D., Liu, J., Lawson, K., & Shen, C. (2022). Differentiable, Learnable, Regionalized Process-Based Models With Multiphysical Outputs can Approach State-Of-The-Art Hydrologic Prediction Accuracy. *Water Resources Research*, *58*(10), e2022WR032404. https://doi.org/10.1029/2022WR032404

Fernando De Toledo, J., Valadares Siqueira, H., Biuk, L. H., Sacchi, R., Azambuja, R. D. R., Junior, R. A., & Asano, P. T. L. (2023). Climate Indices Impact in Monthly Streamflow Series Forecasting. *IEEE Access*, *11*, 21451–21464. IEEE Access. https://doi.org/10.1109/ACCESS.2023.3237982

Frazier, P. I. (2018). *A Tutorial on Bayesian Optimization* (arXiv:1807.02811). arXiv. https://doi.org/10.48550/arXiv.1807.02811

Frenzel, S., & Pompe, B. (2007). Partial Mutual Information for Coupling Analysis of Multivariate Time Series. *Physical Review Letters*, *99*(20), 204101. https://doi.org/10.1103/PhysRevLett.99.204101

Gal, Y., & Ghahramani, Z. (2016). *A Theoretically Grounded Application of Dropout in Recurrent Neural Networks* (arXiv:1512.05287). arXiv. https://doi.org/10.48550/arXiv.1512.05287

Gauch, M., Kratzert, F., Gilon, O., Gupta, H., Mai, J., Nearing, G., Tolson, B., Hochreiter, S., & Klotz, D. (2023). In Defense of Metrics: Metrics Sufficiently Encode Typical Human Preferences Regarding Hydrological Model Performance. *Water Resources Research*, *59*(6), e2022WR033918. https://doi.org/10.1029/2022WR033918

Ghobadi, F., & Kang, D. (2022). Multi-step ahead probabilistic forecasting of daily streamflow using Bayesian deep learning: A multiple case study. *Water*, *14*(22), 3672. https://doi.org/10.3390/w14223672

Gibbs, M. S., McInerney, D., Humphrey, G., Thyer, M. A., Maier, H. R., Dandy, G. C., & Kavetski, D. (2018). State updating and calibration period selection to improve dynamic monthly streamflow forecasts for an environmental flow management application. *Hydrology and Earth System Sciences*, *22*(1), 871–887. https://doi.org/10.5194/hess-22-871-2018

Girihagama, L., Naveed Khaliq, M., Lamontagne, P., Perdikaris, J., Roy, R., Sushama, L., & Elshorbagy, A. (2022). Streamflow modelling and forecasting for Canadian watersheds using LSTM networks with attention mechanism. *Neural Computing and Applications*, *34*(22), 19995–20015. https://doi.org/10.1007/s00521-022-07523-8

Gnann, S. J., Howden, N. J. K., & Woods, R. A. (2020). Hydrological signatures describing the translation of climate seasonality into streamflow seasonality. *Hydrology and Earth System Sciences*, *24*(2), 561–580. https://doi.org/10.5194/hess-24-561-2020

Golan, J. S. (1995). The Moore-Penrose Pseudoinverse. In J. S. Golan, *Foundations of Linear Algebra* (Vol. 11, pp. 198–203). Springer Netherlands. https://doi.org/10.1007/978-94-015-8502-6_16

Graves, A. (2012). *Supervised Sequence Labelling with Recurrent Neural Networks* (Vol. 385). Springer. https://doi.org/10.1007/978-3-642-24797-2

Graves, A., Mohamed, A., & Hinton, G. (2013). *Speech Recognition with Deep Recurrent Neural Networks* (arXiv:1303.5778). arXiv. https://doi.org/10.48550/arXiv.1303.5778

Greff, K., Srivastava, R. K., Koutník, J., Steunebrink, B. R., & Schmidhuber, J. (2017). LSTM: A Search Space Odyssey. *IEEE Transactions on Neural Networks and Learning Systems*, *28*(10), 2222–2232. https://doi.org/10.1109/TNNLS.2016.2582924

Gupta, H. V., Perrin, C., Blöschl, G., Montanari, A., Kumar, R., Clark, M., & Andréassian, V. (2014). Large-sample hydrology: A need to balance depth with breadth. *Hydrology and Earth System Sciences*, *18*(2), 463–477. https://doi.org/10.5194/hess-18-463-2014

Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. *The Journal of Machine Learning Research*, *3*(null), 1157–1182.

Hamlet, A. F., & Lettenmaier, D. P. (1999). Effects of Climate Change on Hydrology and Water Resources in the Columbia River Basin1. *JAWRA Journal of the American Water Resources Association*, *35*(6), 1597–1623. https://doi.org/10.1111/j.1752-1688.1999.tb04240.x

Hanchuan Peng, Fuhui Long, & C. Ding. (2005). Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *27*(8), 1226–1238. https://doi.org/10.1109/TPAMI.2005.159

Hashemi, R., Brigode, P., Garambois, P.-A., & Javelle, P. (2022). How can we benefit from regime information to make more effective use of long short-term memory (LSTM) runoff models?

Hydrology and Earth System Sciences, *26*(22), 5793–5816. https://doi.org/10.5194/hess-26-5793-2022

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning*. Springer. https://doi.org/10.1007/978-0-387-84858-7

Haykin, S. S. (1999). *Neural Networks: A Comprehensive Foundation 2Nd Ed.* Prentice-Hall Of India Pvt. Limited.

Herrera, P. A., Marazuela, M. A., & Hofmann, T. (2022). Parameter estimation and uncertainty analysis in hydrological modeling. *WIREs Water*, *9*(1), e1569. https://doi.org/10.1002/wat2.1569

Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, *9*(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

Huang, G.-B., Zhu, Q.-Y., & Siew, C.-K. (2006). Extreme learning machine: Theory and applications. *Neurocomputing*, *70*(1), 489–501. https://doi.org/10.1016/j.neucom.2005.12.126

Hulle, M. M. V. (2005). Edgeworth Approximation of Multivariate Differential Entropy. *Neural Computation*, *17*(9), 1903–1910. https://doi.org/10.1162/0899766054323026

Hunt, K. M. R., Matthews, G. R., Pappenberger, F., & Prudhomme, C. (2022). Using a long short-term memory (LSTM) neural network to boost river streamflow forecasts over the western United States. *Hydrology and Earth System Sciences*, *26*(11), 5449–5472. https://doi.org/10.5194/hess-26-5449-2022

Ikram, R. M. A., Hazarika, B. B., Gupta, D., Heddam, S., & Kisi, O. (2023). Streamflow prediction in mountainous region using new machine learning and data preprocessing methods: A case study. *Neural Computing and Applications*, *35*(12), 9053–9070. https://doi.org/10.1007/s00521-022-08163-8

Ioffe, S., & Szegedy, C. (2015). *Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift* (arXiv:1502.03167). arXiv. https://doi.org/10.48550/arXiv.1502.03167

J. A. Criswell. (2022). Multiple Input Single Target Streamflow Forecast by Neurowavelet Networks. *2022 IEEE 2nd International Conference on Information Communication and Software Engineering (ICICSE)*, 29–36. https://doi.org/10.1109/ICICSE55337.2022.9828938

Jahangir, M. S., & Quilty, J. (2024). Generative deep learning for probabilistic streamflow forecasting: Conditional variational auto-encoder. *Journal of Hydrology*, *629*, 130498. https://doi.org/10.1016/j.jhydrol.2023.130498

Jahangir, M. S., You, J., & Quilty, J. (2023). A quantile-based encoder-decoder framework for multi-step ahead runoff forecasting. *Journal of Hydrology*, *619*, 129269. https://www.sciencedirect.com/science/article/pii/S0022169423002111

Jolliffe, I. T. (2002). *Principal Component Analysis*. Springer Science & Business Media.

Kabir, S., Patidar, S., & Pender, G. (2020). Investigating capabilities of machine learning techniques in forecasting stream flow. *Proceedings of the Institution of Civil Engineers - Water Management*, *173*(2), 69–86. https://doi.org/10.1680/jwama.19.00001

Kambalimath S, S., & Deka, P. C. (2021). Performance enhancement of SVM model using discrete wavelet transform for daily streamflow forecasting. *Environmental Earth Sciences*, *80*(3), 101. https://doi.org/10.1007/s12665-021-09394-z

Karsavran, Y., Erdik, T., & Ozger, M. (2023). An improved technique for streamflow forecasting between Turkish straits. *Acta Geophysica*. https://doi.org/10.1007/s11600-023-01216-z

Karunanithi, N., Grenney, W. J., Whitley, D., & Bovee, K. (1994). Neural Networks for River Flow Prediction. *Journal of Computing in Civil Engineering*, *8*(2), 201–220. https://doi.org/10.1061/(ASCE)0887-3801(1994)8:2(201)

Keating, C., Lee, D., Bazo, J., & Block, P. (2021). Leveraging multi-model season-ahead streamflow forecasts to trigger advanced flood preparedness in Peru. *Natural Hazards and Earth System Sciences*, *21*(7), 2215–2231. https://doi.org/10.5194/nhess-21-2215-2021

Khatun, A., Sahoo, B., & Chatterjee, C. (2023). Two novel error-updating model frameworks for short-to-medium range streamflow forecasting using bias-corrected rainfall inputs: Development and comparative assessment. *Journal of Hydrology*, *618*, 129199. https://doi.org/10.1016/j.jhydrol.2023.129199

Khazaee Poul, A., Shourian, M., & Ebrahimi, H. (2019). A Comparative Study of MLR, KNN, ANN and ANFIS Models with Wavelet Transform in Monthly Stream Flow Prediction. *Water Resources Management*, *33*(8), 2907–2923. https://doi.org/10.1007/s11269-019-02273-0

Kingma, D. P., & Ba, J. (2017). *Adam: A Method for Stochastic Optimization* (arXiv:1412.6980). arXiv. https://doi.org/10.48550/arXiv.1412.6980

Kisi, O., Alizamir, M., & Shiri, J. (2021). Conjunction Model Design for Intermittent Streamflow Forecasts: Extreme Learning Machine with Discrete Wavelet Transform. In R. C. Deo, P. Samui, O. Kisi, & Z. M. Yaseen (Eds.), *Intelligent Data Analytics for Decision-Support Systems in Hazard Mitigation: Theory and Practice of Hazard Mitigation* (pp. 171–181). Springer. https://doi.org/10.1007/978-981-15-5772-9_9

Klingler, C., Schulz, K., & Herrnegger, M. (2021). LamaH-CE: LArge-SaMple DAta for Hydrology and Environmental Sciences for Central Europe. *Earth System Science Data*, *13*(9), 4529–4565. https://doi.org/10.5194/essd-13-4529-2021

Klotz, D., Kratzert, F., Gauch, M., Keefe Sampson, A., Brandstetter, J., Klambauer, G., Hochreiter, S., & Nearing, G. (2022). Uncertainty estimation with deep learning for rainfall–runoff modeling. *Hydrology and Earth System Sciences*, *26*(6), 1673–1693. https://doi.org/10.5194/hess-26-1673-2022

Knoben, W. J. M., Freer, J. E., & Woods, R. A. (2019). Technical note: Inherent benchmark or not? Comparing Nash–Sutcliffe and Kling–Gupta efficiency scores. *Hydrology and Earth System Sciences*, *23*(10), 4323–4331. https://doi.org/10.5194/hess-23-4323-2019

Köppen, W. (1884). Die Wärmezonen der Erde, nach der Dauer der heissen, mässigen und kalten Zeit und nach der Wirkung der Wärme auf die organische Welt betrachtet. *Meteorologische Zeitschrift*, *1*, 215–226.

Kottek, M. G., Jürgen; Beck, Christoph; Rudolf, Bruno; Rubel, Franz. (2006). World Map of the Köppen-Geiger climate classification updated. *Meteorologische Zeitschrift*, *15*(3), 259–263. https://doi.org/10.1127/0941-2948/2006/0130

Kratzert, F., Klotz, D., Herrnegger, M., Sampson, A. K., Hochreiter, S., & Nearing, G. S. (2019). Toward Improved Predictions in Ungauged Basins: Exploiting the Power of Machine Learning. *Water Resources Research*, *55*(12), 11344–11354. https://doi.org/10.1029/2019WR026065

Kratzert, F., Klotz, D., Shalev, G., Klambauer, G., Hochreiter, S., & Nearing, G. (2019). Towards learning universal, regional, and local hydrological behaviors via machine learning applied to large-sample datasets. *Hydrology and Earth System Sciences*, *23*(12), 5089–5110. https://doi.org/10.5194/hess-23-5089-2019

Kratzert, F., Nearing, G., Addor, N., Erickson, T., Gauch, M., Gilon, O., Gudmundsson, L., Hassidim, A., Klotz, D., Nevo, S., Shalev, G., & Matias, Y. (2023). Caravan—A global community dataset for large-sample hydrology. *Scientific Data*, *10*(1), Article 1. https://doi.org/10.1038/s41597-023-01975-w

Kreibich, H., Müller, M., Schröter, K., & Thieken, A. H. (2017). New insights into flood warning reception and emergency response by affected parties. *Natural Hazards and Earth System Sciences*, *17*(12), 2075–2092. https://doi.org/10.5194/nhess-17-2075-2017

Kuhn, M., & Johnson, K. (2019). *Feature Engineering and Selection: A Practical Approach for Predictive Models*. CRC Press.

Lan Anh, N. T., Boxall, J. B., Saul, A. J., & Willem, P. (2010, September 27). An evaluation of three lumped conceptual rainfall-runoff models at catchment scale. *Role of Hydrology in Managing Consequences of a Changing Global Environment*. BHS 3rd International Conference. https://doi.org/10.7558/bhs.2010.ic109

Langevin, C. D., Hughes, J. D., Banta, E. R., Provost, A. M., Niswonger, R. G., & Panday, S. (2017). *MODFLOW 6 Modular Hydrologic Model* [Computer software]. U.S. Geological Survey. https://doi.org/10.5066/F76Q1VQV

Lhomme, J.-P. (1997). A THEORETICAL BASIS FOR THE PRIESTLEY-TAYLOR COEFFICIENT. *Boundary-Layer Meteorology*, *82*(2), 179–191. https://doi.org/10.1023/A:1000281114105

Li, B., Sun, T., Tian, F., & Ni, G. (2023). Enhancing process-based hydrological models with embedded neural networks: A hybrid approach. *Journal of Hydrology*, *625*, 130107. https://doi.org/10.1016/j.jhydrol.2023.130107

Li, J., & Yuan, X. (2023). Daily Streamflow Forecasts Based on Cascade Long Short-Term Memory (LSTM) Model over the Yangtze River Basin. *Water*, *15*(6), Article 6. https://doi.org/10.3390/w15061019

Liu, C., Liu, D., & Mu, L. (2022). Improved Transformer Model for Enhanced Monthly Streamflow Predictions of the Yangtze River. *IEEE Access*, *10*, 58240–58253. IEEE Access. https://doi.org/10.1109/ACCESS.2022.3178521

Liu, L., Vervoort, R. W., Johnson, F., & Marshall, L. (2023). *Comparing the observed and predicted actual evapotranspiration in lumped conceptual hydrological models*.

Liu, J., Bian, Y., Lawson, K. & Shen, C. Probing the limit of hydrologic predictability with the

Transformer network. *J. Hydrol.* **637**, 131389 (2024).

Lü, X., Zhong, M., & Zha, D. (2022). Runoff Forecasting Using Machine-Learning Methods: Case Study in the Middle Reaches of Xijiang River. *Frontiers in Big Data*. https://doi.org/10.3389/fdata.2021.752406

Luo, C. (2021). Comparing Five Kinematic Wave Schemes for Open-Channel Routing for Wide-Tooth-Comb-Wave Hydrographs. *Journal of Hydrologic Engineering*, *26*(4), 04021009. https://doi.org/10.1061/(ASCE)HE.1943-5584.0002079

M. Gada, Z. Haria, A. Mankad, K. Damania, & S. Sankhe. (2021). Automated Feature Engineering and Hyperparameter optimization for Machine Learning. *2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS)*, *1*, 981–986. https://doi.org/10.1109/ICACCS51430.2021.9441668

Madsen, T. (2017). *Evaluation of a distributed streamflow forecast model at multiple watershed scales*. https://dr.lib.iastate.edu/handle/20.500.12876/29753

Maheswaran, R., & Khosa, R. (2012). Comparative study of different wavelets for hydrologic forecasting. *Computers & Geosciences*, *46*, 284–295. https://doi.org/10.1016/j.cageo.2011.12.015

Man, Y., Yang, Q., Shao, J., Wang, G., Bai, L., & Xue, Y. (2023). Enhanced LSTM Model for Daily Runoff Prediction in the Upper Huai River Basin, China. *Engineering*, *24*, 229–238. https://doi.org/10.1016/j.eng.2021.12.022

Martel, J., Demeester, K., Brissette, F., Poulin, A., & Arsenault, R. (2017). HMETS-A simple and efficient hydrology model for teaching hydrological modelling, flow forecasting and climate change impacts. *International Journal of Engineering Education*. https://www.semanticscholar.org/paper/HMETS-A-simple-and-efficient-hydrology-model-for-Martel-Demeester/3148822c65e7a195f65fb2137bcb804177bbfd22

May, R. J., Dandy, G. C., Maier, H. R., & Nixon, J. B. (2008). Application of partial mutual information variable selection to ANN forecasting of water quality in water distribution systems. *Environmental Modelling & Software*, *23*(10), 1289–1299. https://doi.org/10.1016/j.envsoft.2008.03.008

May, R. J., Maier, H. R., Dandy, G. C., & Fernando, T. M. K. G. (2008). Non-linear variable selection for artificial neural networks using partial mutual information. *Environmental Modelling & Software*, *23*(10), 1312–1326. https://doi.org/10.1016/j.envsoft.2008.03.007

Mazarei Behbahani, M. R., & Mazarei, A. (2023). A new criteria for determining the best decomposition level and filter for wavelet-based data-driven forecasting frameworks-validating using three case studies on the CAMELS dataset. *Stochastic Environmental Research and Risk Assessment*, *37*(12), 4827–4842. https://doi.org/10.1007/s00477-023-02531-z

Mehdizadeh, S., Fathian, F., & Adamowski, J. F. (2019). Hybrid artificial intelligence-time series models for monthly streamflow modeling. *Applied Soft Computing*, *80*, 873–887. https://doi.org/10.1016/j.asoc.2019.03.046

Modaresi, F., Araghinejad, S., & Ebrahimi, K. (2018). A Comparative Assessment of Artificial Neural Network, Generalized Regression Neural Network, Least-Square Support Vector Regression, and K-Nearest Neighbor Regression for Monthly Streamflow Forecasting in Linear and Nonlinear Conditions. *Water Resources Management*, *32*(1), 243–258. https://doi.org/10.1007/s11269-017-1807-2

Molavipour, S., Bassi, G., & Skoglund, M. (2021). Neural Estimators for Conditional Mutual Information Using Nearest Neighbors Sampling. *IEEE Transactions on Signal Processing*, *69*, 766–780. https://doi.org/10.1109/TSP.2021.3050564

Molinos, J. G., Viana, M., Brennan, M., & Donohue, I. (2015). Importance of Long-Term Cycles for Predicting Water Level Dynamics in Natural Lakes. *PLOS ONE*, *10*(3), e0119253. https://doi.org/10.1371/journal.pone.0119253

Momeneh, S., & Nourani, V. (2023). Performance evaluation of artificial neural network model in hybrids with various preprocessors for river streamflow forecasting. *AQUA - Water Infrastructure, Ecosystems and Society*, *72*(6), 947–968. https://doi.org/10.2166/aqua.2023.010

Montavon, G., Orr, G., & Müller, K.-R. (2012). *Neural Networks: Tricks of the Trade*. Springer.

Moriasi, D. N., Arnold, J. G., Van Liew, M. W., Bingner, R. L., Harmel, R. D., & Veith, T. L. (2007). Model evaluation guidelines for systematic quantification of accuracy in watershed simulations. *Transactions of the ASABE*, *50*(3), 885–900. https://handle.nal.usda.gov/10113/9298

Mote, P. W., & Salathé, E. P. (2010). Future climate in the Pacific Northwest. *Climatic Change*, *102*(1), 29–50. https://doi.org/10.1007/s10584-010-9848-z

Motoda, H. L., Hiroshi (Ed.). (2007). *Computational Methods of Feature Selection*. Chapman and Hall/CRC. https://doi.org/10.1201/9781584888796

Mount, N. J., Maier, H. R., Toth, E., Elshorbagy, A., Solomatine, D., Chang, F.-J., & Abrahart, R. J. (2016). Data-driven modelling approaches for socio-hydrology: Opportunities and challenges within the Panta Rhei Science Plan. *Hydrological Sciences Journal*, *61*(7), 1192–1208. https://doi.org/10.1080/02626667.2016.1159683

N. Moriasi, D., G. Arnold, J., W. Van Liew, M., L. Bingner, R., D. Harmel, R., & L. Veith, T. (2007). Model Evaluation Guidelines for Systematic Quantification of Accuracy in Watershed Simulations. *Transactions of the ASABE*, *50*(3), 885–900. https://doi.org/10.13031/2013.23153

Nash, J. E., & Sutcliffe, J. V. (1970). River flow forecasting through conceptual models part I — A discussion of principles. *Journal of Hydrology*, *10*(3), 282–290. https://doi.org/10.1016/0022-1694(70)90255-6

Nearing, G., Cohen, D., Dube, V., Gauch, M., Gilon, O., Harrigan, S., Hassidim, A., Klotz, D., Kratzert, F., Metzger, A., Nevo, S., Pappenberger, F., Prudhomme, C., Shalev, G., Shenzis, S., Tekalign, T. Y., Weitzner, D., & Matias, Y. (2024). Global prediction of extreme floods in ungauged watersheds. *Nature*, *627*(8004), 559–563. https://doi.org/10.1038/s41586-024-07145-1

Nevo, S., Morin, E., Gerzi Rosenthal, A., Metzger, A., Barshai, C., Weitzner, D., Voloshin, D., Kratzert, F., Elidan, G., Dror, G., Begelman, G., Nearing, G., Shalev, G., Noga, H., Shavitt, I., Yuklea, L., Royz, M., Giladi, N., Peled Levi, N., … Matias, Y. (2022). Flood forecasting with machine learning models in an operational framework. *Hydrology and Earth System Sciences*, *26*(15), 4013–4032. https://doi.org/10.5194/hess-26-4013-2022

Newman, A. J., Clark, M. P., Sampson, K., Wood, A., Hay, L. E., Bock, A., Viger, R. J., Blodgett, D., Brekke, L., Arnold, J. R., Hopson, T., & Duan, Q. (2015). Development of a large-sample watershed-scale hydrometeorological data set for the contiguous USA: Data set characteristics and assessment of regional variability in hydrologic model performance. *Hydrology and Earth System Sciences*, *19*(1), 209–223. https://doi.org/10.5194/hess-19-209-2015

Ng, K. W., Huang, Y. F., Koo, C. H., Chong, K. L., El-Shafie, A., & Najah Ahmed, A. (2023). A review of hybrid deep learning applications for streamflow forecasting. *Journal of Hydrology*, *625*, 130141. https://doi.org/10.1016/j.jhydrol.2023.130141

Nguyen, H. (2022). Daily Streamflow Forecasting Using Extreme Learning Machine and Optimization Algorithm. Case Study: Tra Khuc River in Vietnam. *Geographia Technica*, *17*, 148–163. https://doi.org/10.21163/GT_2022.172.13

Ni, L., Wang, D., Singh, V. P., Wu, J., Wang, Y., Tao, Y., & Zhang, J. (2020). Streamflow and rainfall forecasting by two long short-term memory-based models. *Journal of Hydrology*, *583*, 124296. https://doi.org/10.1016/j.jhydrol.2019.124296

Ni, L., Wang, D., Wu, J., Wang, Y., Tao, Y., Zhang, J., & Liu, J. (2020). Streamflow forecasting using extreme gradient boosting model coupled with Gaussian mixture model. *Journal of Hydrology*, *586*, 124901. https://doi.org/10.1016/j.jhydrol.2020.124901

Nifa, K., Boudhar, A., Ouatiki, H., Elyoussfi, H., Bargam, B., & Chehbouni, A. (2023). Deep Learning Approach with LSTM for Daily Streamflow Prediction in a Semi-Arid Area: A Case Study of Oum Er-Rbia River Basin, Morocco. *Water*, *15*(2). https://doi.org/10.3390/w15020262

Nourani, V., Najafi, H., Amini, A. B., & Tanaka, H. (2021). Using Hybrid Wavelet-Exponential Smoothing Approach for Streamflow Modeling. *Complexity*, *2021*, 6611848. https://doi.org/10.1155/2021/6611848

Olah, C. (2015, August 27). *Understanding LSTM Networks—Colah's blog* [Blog]. Colah's Blog. https://colah.github.io/posts/2015-08-Understanding-LSTMs/

Olfatmiri, Y., Bahreinimotlagh, M., Jabbari, E., Kawanisi, K., Hasanabadi, A., & Al Sawaf, M. B. (2022). Application of Acoustic tomographic data to the short-term streamflow forecasting using data-driven methods and discrete wavelet transform. *Journal of Hydrology*, *609*, 127739. https://doi.org/10.1016/j.jhydrol.2022.127739

Oo, H. T., Zin, W. W., & Kyi, C. C. T. (2020). Analysis of Streamflow Response to Changing Climate Conditions Using SWAT Model. *Civil Engineering Journal*, *6*(2), Article 2. https://doi.org/10.28991/cej-2020-03091464

Papacharalampous, G., & Tyralis, H. (2020). Hydrological time series forecasting using simple combinations: Big data testing and investigations on one-year ahead river flow predictability. *Journal of Hydrology*, *590*, 125205. https://doi.org/10.1016/j.jhydrol.2020.125205

Papacharalampous, G., & Tyralis, H. (2022). A review of machine learning concepts and methods for addressing challenges in probabilistic hydrological post-processing and forecasting. *Frontiers in Water*, *4*. https://www.frontiersin.org/articles/10.3389/frwa.2022.961954

Papacharalampous, G., Tyralis, H., & Koutsoyiannis, D. (2019). Comparison of stochastic and machine learning methods for multi-step ahead forecasting of hydrological processes. *Stochastic Environmental Research and Risk Assessment*, *33*(2), 481–514. https://doi.org/10.1007/s00477-018-1638-6

Papacharalampous, G., Tyralis, H., Pechlivanidis, I. G., Grimaldi, S., & Volpi, E. (2022). Massive feature extraction for explaining and foretelling hydroclimatic time series forecastability at the global scale. *Geoscience Frontiers*, *13*(3), 101349. https://doi.org/10.1016/j.gsf.2022.101349

Peel, M. C., Finlayson, B. L., & McMahon, T. A. (2007). Updated world map of the Köppen-Geiger climate classification. *Hydrology and Earth System Sciences*, *11*(5), 1633–1644. https://doi.org/10.5194/hess-11-1633-2007

Percival, D. B., & Walden, A. T. (2000). *Wavelet Methods for Time Series Analysis*. Cambridge University Press. https://doi.org/10.1017/CBO9780511841040

Perrin, C., Michel, C., & Andréassian, V. (2003). Improvement of a parsimonious model for streamflow simulation. *Journal of Hydrology*, *279*(1), 275–289. https://doi.org/10.1016/S0022-1694(03)00225-7

Petty, T. R., & Dhingra, P. (2017). Streamflow Hydrology Estimate Using Machine Learning (SHEM). *Jawra Journal of the American Water Resources Association*. https://doi.org/10.1111/1752-1688.12555

Pham, H., Guan, M., Zoph, B., Le, Q., & Dean, J. (2018). Efficient Neural Architecture Search via Parameters Sharing. *Proceedings of the 35th International Conference on Machine Learning*, 4095–4104. https://proceedings.mlr.press/v80/pham18a.html

Pham, L. T., Luo, L., & Finley, A. (2021). Evaluation of random forests for short-term daily streamflow forecasting in rainfall- and snowmelt-driven watersheds. *Hydrology and Earth System Sciences*, *25*(6), 2997–3015. https://doi.org/10.5194/hess-25-2997-2021

Pham, L. T., Luo, L., & Finley, A. O. (2020). *Evaluation of Random Forest for short-term daily streamflow forecast in rainfall and snowmelt driven watersheds* [Preprint]. Catchment hydrology/Modelling approaches. https://doi.org/10.5194/hess-2020-305

Pidwirny, M. (2021). *Appendix 2: Köppen Climate Classification System*. https://pressbooks.bccampus.ca/physgeoglabmanual1/back-matter/appendix-2-koppen-climate-classification-system/

Prasad, R., Deo, R. C., Li, Y., & Maraseni, T. (2017). Input selection and performance optimization of ANN-based streamflow forecasts in the drought-prone Murray Darling Basin region using IIS and MODWT algorithm. *Atmospheric Research*, *197*, 42–63. https://doi.org/10.1016/j.atmosres.2017.06.014

Quilty, J., & Adamowski, J. (2018). Addressing the incorrect usage of wavelet-based hydrological and water resources forecasting models for real-world applications with best practices and a

new forecasting framework. *Journal of Hydrology*, *563*, 336–353. https://doi.org/10.1016/j.jhydrol.2018.05.003

Quilty, J., & Adamowski, J. (2020). A stochastic wavelet-based data-driven framework for forecasting uncertain multiscale hydrological and water resources processes. *Environmental Modelling & Software*, *130*, 104718. https://doi.org/10.1016/j.envsoft.2020.104718

Quilty, J., & Adamowski, J. (2021). A maximal overlap discrete wavelet packet transform integrated approach for rainfall forecasting – A case study in the Awash River Basin (Ethiopia). *Environmental Modelling & Software*, *144*, 105119. https://doi.org/10.1016/j.envsoft.2021.105119

Quilty, J., Adamowski, J., Khalil, B., & Rathinasamy, M. (2016). Bootstrap rank-ordered conditional mutual information (broCMI): A nonlinear input variable selection method for water resources modeling. *Water Resources Research*, *52*(3), 2299–2326. https://doi.org/10.1002/2015WR016959

Quilty, J., Jahangir, M. S., You, J., Hughes, H., Hah, D., & Tzoganakis, I. (2023). Bayesian extreme learning machines for hydrological prediction uncertainty. *Journal of Hydrology*, *626*, 130138. https://doi.org/10.1016/j.jhydrol.2023.130138

R. M. Adnan, Z. Liang, A. Kuriqi, O. Kisi, A. Malik, & B. Li. (2020). Streamflow forecasting using heuristic machine learning methods. *2020 2nd International Conference on Computer and Information Sciences (ICCIS)*, 1–6. https://doi.org/10.1109/ICCIS49240.2020.9257658

Rahman, A. T. M. S., Hosono, T., Quilty, J. M., Das, J., & Basak, A. (2020). Multiscale groundwater level forecasting: Coupling new machine learning approaches with wavelet transforms. *Advances in Water Resources*, *141*, 103595. https://doi.org/10.1016/j.advwatres.2020.103595

Ramos, M.-H., Cassagnole, M., Zalachori, I., Thirel, G., Garçon, R., Gailhard, J., & Ouillon, T. (2021). *Understanding how hydrological forecast quality impacts the management of*

hydroelectric reservoirs (EGU21-14532). EGU21. Copernicus Meetings. https://doi.org/10.5194/egusphere-egu21-14532

Randrianasolo, A., Ramos, M. H., Thirel, G., Andréassian, V., & Martin, E. (2010). Comparing the scores of hydrological ensemble forecasts issued by two different hydrological models. *Atmospheric Science Letters*, *11*(2), 100–107. https://doi.org/10.1002/asl.259

Rathinasamy, M., Adamowski, J., & Khosa, R. (2013). Multiscale streamflow forecasting using a new Bayesian Model Average based ensemble multi-wavelet Volterra nonlinear method. *Journal of Hydrology*, *507*, 186–200. https://doi.org/10.1016/j.jhydrol.2013.09.025

Razvan Pascanu, Tomas Mikolov, & Yoshua Bengio. (2013). On the difficulty of training recurrent neural networks. In Sanjoy Dasgupta & David McAllester (Eds.), *Proceedings of the 30th International Conference on Machine Learning* (Vol. 28, Issue 3, pp. 1310–1318). PMLR. https://proceedings.mlr.press/v28/pascanu13.html

Regan, R. S., Markstrom, S. L., Hay, L. E., Viger, R. J., Norton, P. A., Driscoll, J. M., & LaFontaine, J. H. (2018). Description of the National Hydrologic Model for use with the Precipitation-Runoff Modeling System (PRMS). In *Techniques and Methods* (6-B9). U.S. Geological Survey. https://doi.org/10.3133/tm6B9

Ren, K., Fang, W., Qu, J., Zhang, X., & Shi, X. (2020). Comparison of eight filter-based feature selection methods for monthly streamflow forecasting – Three case studies on CAMELS data sets. *Journal of Hydrology*, *586*, 124897. https://doi.org/10.1016/j.jhydrol.2020.124897

Ren, K., Wang, X., Shi, X., Qu, J., & Fang, W. (2021). Examination and comparison of binary metaheuristic wrapper-based input variable selection for local and global climate information-driven one-step monthly streamflow forecasting. *Journal of Hydrology*, *597*, 126152. https://doi.org/10.1016/j.jhydrol.2021.126152

Ren, Q., Li, H., Li, M., Kong, T., & Guo, R. (2023). Bayesian incremental learning paradigm for online monitoring of dam behavior considering global uncertainty. *Applied Soft Computing*, *143*(C). https://doi.org/10.1016/j.asoc.2023.110411

Robertson, D. E., Shrestha, D. L., & Wang, Q. J. (2013). Post-processing rainfall forecasts from numerical weather prediction models for short-term streamflow forecasting. *Hydrology and Earth System Sciences*, *17*(9), 3587–3603. https://doi.org/10.5194/hess-17-3587-2013

Rodríguez, P., Bautista, M. Á., Gonzàlez, J., & Escalera, S. (2018). Beyond One-Hot Encoding: Lower Dimensional Target Embedding. *Image and Vision Computing*. https://doi.org/10.1016/j.imavis.2018.04.004

Rohli, R. V., & Vega, A. J. (2017). *Climatology*. Jones & Bartlett Learning.

Rolim, L. Z. R., de Souza Filho, F. de A., & Brown, C. (2023). A Multi-model Framework for Streamflow Forecasting Based on Stochastic Models: An Application to the State Of Ceará, Brazil. *Water Conservation Science and Engineering*, *8*(1), 7. https://doi.org/10.1007/s41101-023-00184-1

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, *323*(6088), Article 6088. https://doi.org/10.1038/323533a0

Sabzipour, B., Arsenault, R., Troin, M., Martel, J.-L., Brissette, F., Brunet, F., & Mai, J. (2023). Comparing a long short-term memory (LSTM) neural network with a physically-based hydrological model for streamflow forecasting over a Canadian catchment. *Journal of Hydrology*, *627*, 130380. https://doi.org/10.1016/j.jhydrol.2023.130380

Samadianfard, S., Asadi, E., Jarhan, S., Kazemi, H., Kheshtgar, S., Kisi, O., Sajjadi, S., & Manaf, A. A. (2018). Wavelet neural networks and gene expression programming models to predict short-term soil temperature at different depths. *Soil and Tillage Research*, *175*, 37–50. https://doi.org/10.1016/j.still.2017.08.012

Santos, C. A. G., Nascimento, G. R. do, de Farias, C. A. S., Silva, R. M. da, & Mishra, M. (2023). Short- and long-term streamflow forecasting using wavelet neural networks for complex watersheds: A case study in the Mahanadi River, India. *Ecological Informatics*, *73*, 101945. https://doi.org/10.1016/j.ecoinf.2022.101945

Scharffenberg, W., & Harris, J. (2012). *Hydrologic Engineering Center Hydrologic Modeling System, HEC-HMS: Interior Flood Modeling*. 1–3. https://doi.org/10.1061/40976(316)632

Seibert, J., & Bergström, S. (2022). A retrospective on hydrological catchment modelling based on half a century with the HBV model. *Hydrology and Earth System Sciences*, *26*(5), 1371–1388. https://doi.org/10.5194/hess-26-1371-2022

Sezen, C., Bezak, N., Bai, Y., & Šraj, M. (2019). Hydrological modelling of karst catchment using lumped conceptual and data mining models. *Journal of Hydrology*, *576*, 98–110. https://doi.org/10.1016/j.jhydrol.2019.06.036

Shaikh, W. A., Shah, S. F., Pandhiani, S. M., Solangi, M. A., Farooq, M., Ahmad, H., Kashuri, A., Jarasthitikulchai, N., & Sudsutad, W. (2022). A hybrid forecasting model based on the group method of data handling and wavelet decomposition for monthly rivers streamflow data sets. *Open Physics*, *20*(1), 1096–1111. https://doi.org/10.1515/phys-2022-0066

Shannon, C. E. (1948). A Mathematical Theory of Communication. *Bell System Technical Journal*, *27*(3), 379–423. https://doi.org/10.1002/j.1538-7305.1948.tb01338.x

Sharma, A. (2000a). Seasonal to interannual rainfall probabilistic forecasts for improved water supply management: Part 1 — A strategy for system predictor identification. *Journal of Hydrology*, *239*(1), 232–239. https://doi.org/10.1016/S0022-1694(00)00346-2

Sharma, A. (2000b). Seasonal to interannual rainfall probabilistic forecasts for improved water supply management: Part 3 — A nonparametric probabilistic forecast model. *Journal of Hydrology*, *239*(1), 249–258. https://doi.org/10.1016/S0022-1694(00)00348-6

Sharma, A., Luk, K. C., Cordery, I., & Lall, U. (2000). Seasonal to interannual rainfall probabilistic forecasts for improved water supply management: Part 2 — Predictor identification of quarterly rainfall using ocean-atmosphere information. *Journal of Hydrology*, *239*(1), 240–248. https://doi.org/10.1016/S0022-1694(00)00347-4

Sharma, A., & Mehrotra, R. (2014). An information theoretic alternative to model a natural system using observational information alone. *Water Resources Research*, *50*(1), 650–660. https://doi.org/10.1002/2013WR013845

Sharma, A., & Wasko, C. (2019). Chapter 5—Trends and Changes in Streamflow With Climate. In R. Teegavarapu (Ed.), *Trends and Changes in Hydroclimatic Variables* (pp. 275–304). Elsevier. https://doi.org/10.1016/B978-0-12-810985-4.00005-0

Sharma, P., & Machiwal, D. (2021). *Advances in Streamflow Forecasting: From Traditional to Modern Approaches*. Elsevier Science.

Sharma, S., Raj Ghimire, G., & Siddique, R. (2022). Machine learning for postprocessing ensemble streamflow forecasts. *Journal of Hydroinformatics*, *25*(1), 126–139. https://doi.org/10.2166/hydro.2022.114

Shi, B., Wang, P., Jiang, J., & Liu, R. (2018). Applying high-frequency surrogate measurements and a wavelet-ANN model to provide early warnings of rapid surface water quality anomalies. *Science of The Total Environment*, *610–611*, 1390–1399. https://doi.org/10.1016/j.scitotenv.2017.08.232

Shrestha, D. L., Robertson, D. E., Wang, Q. J., Pagano, T. C., & Hapuarachchi, H. a. P. (2013). Evaluation of numerical weather prediction model precipitation forecasts for short-term streamflow forecasting purpose. *Hydrology and Earth System Sciences*, *17*(5), 1913–1931. https://doi.org/10.5194/hess-17-1913-2013

Sikorska-Senoner, A. E., & Quilty, J. M. (2021a). A novel ensemble-based conceptual-data-driven approach for improved streamflow simulations. *Environmental Modelling & Software*, *143*, 105094. https://www.sciencedirect.com/science/article/pii/S1364815221001377

Sikorska-Senoner, A. E., & Quilty, J. M. (2021b). A novel ensemble-based conceptual-data-driven approach for improved streamflow simulations. *Environmental Modelling & Software*, *143*, 105094.

Siler, N., & Durran, D. (2016). What Causes Weak Orographic Rain Shadows? Insights from Case Studies in the Cascades and Idealized Simulations. *Journal of the Atmospheric Sciences*, *73*(10), 4077–4099. https://doi.org/10.1175/JAS-D-15-0371.1

Siler, N., Roe, G., & Durran, D. (2013). On the Dynamical Causes of Variability in the Rain-Shadow Effect: A Case Study of the Washington Cascades. *Journal of Hydrometeorology*, *14*(1), 122–139. https://doi.org/10.1175/JHM-D-12-045.1

Sivakumar, B. (2017). Modern Nonlinear Time Series Methods—Chaos in Hydrology: Bridging Determinism and Stochasticity. In B. Sivakumar (Ed.), *Modern Nonlinear Time Series Methods—Chaos in Hydrology: Bridging Determinism and Stochasticity* (pp. 111–145). Springer Netherlands. https://doi.org/10.1007/978-90-481-2552-4_4

Sivakumar, B., & Berndtsson, R. (2010). *Advances in Data-based Approaches for Hydrologic Modeling and Forecasting*. World Scientific.

Snieder, E., Shakir, R., & Khan, U. T. (2020). A comprehensive comparison of four input variable selection methods for artificial neural network flow forecasting models. *Journal of Hydrology*, *583*, 124299. https://doi.org/10.1016/j.jhydrol.2019.124299

Somorowska, U. (2017). Climate-driven changes to streamflow patterns in a groundwater-dominated catchment. *Acta Geophysica*, *65*(4), 789–798. https://doi.org/10.1007/s11600-017-0054-5

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, *15*(56), 1929–1958. http://jmlr.org/papers/v15/srivastava14a.html

Stockham, A. J., Schultz, D. M., Fairman, J. G., & Draude, A. P. (2018). Quantifying the Rain-Shadow Effect: Results from the Peak District, British Isles. *Bulletin of the American Meteorological Society*, *99*(4), 777–790. https://doi.org/10.1175/BAMS-D-17-0256.1

Sun, N., Zhang, S., Peng, T., Zhang, N., Zhou, J., & Zhang, H. (2022). Multi-Variables-Driven Model Based on Random Forest and Gaussian Process Regression for Monthly Streamflow Forecasting. *Water*, *14*(11). https://doi.org/10.3390/w14111828

Sutskever, I., Vinyals, O., & Le, Q. V. (2014). *Sequence to Sequence Learning with Neural Networks* (arXiv:1409.3215). arXiv. https://doi.org/10.48550/arXiv.1409.3215

Syed, Z., Mahmood, P., Haider, S., Ahmad, S., Jadoon, K. Z., Farooq, R., Syed, S., & Ahmad, K. (2023). Short–long-term streamflow forecasting using a coupled wavelet transform–artificial neural network (WT–ANN) model at the Gilgit River Basin, Pakistan. *Journal of Hydroinformatics*, *25*(3), 881–894. https://doi.org/10.2166/hydro.2023.161

Szczepanek, R. (2022). Daily Streamflow Forecasting in Mountainous Catchment Using XGBoost, LightGBM and CatBoost. *Hydrology*, *9*(12), Article 12. https://doi.org/10.3390/hydrology9120226

Tarasova, L., Gnann, S., Yang, S., Hartmann, A. & Wagener, T. Catchment characterization: Current descriptors, knowledge gaps and future opportunities. *Earth-Science Reviews* **252**, 104739 (2024).

Teng, J., Jakeman, A. J., Vaze, J., Croke, B. F. W., Dutta, D., & Kim, S. (2017). Flood inundation modelling: A review of methods, recent advances and uncertainty analysis. *Environmental Modelling & Software*, *90*, 201–216. https://doi.org/10.1016/j.envsoft.2017.01.006

Tongal, H., & Booij, M. J. (2018). Simulation and forecasting of streamflows using machine learning models coupled with base flow separation. *Journal of Hydrology*, *564*, 266–282. https://doi.org/10.1016/j.jhydrol.2018.07.004

Tripathy, K. P., & Mishra, A. K. (2024). Deep learning in hydrology and water resources disciplines: Concepts, methods, applications, and research directions. *Journal of Hydrology*, *628*, 130458. https://doi.org/10.1016/j.jhydrol.2023.130458

Troin, M., Arsenault, R., Wood, A. W., Brissette, F., & Martel, J.-L. (2021). Generating Ensemble Streamflow Forecasts: A Review of Methods and Approaches Over the Past 40 Years. *Water Resources Research*, *57*(7), e2020WR028392. https://doi.org/10.1029/2020WR028392

Tyralis, H., Papacharalampous, G., & Langousis, A. (2021). Chapter 2—Streamflow forecasting at large time scales using statistical models. In P. Sharma & D. Machiwal (Eds.), *Advances in Streamflow Forecasting* (pp. 51–86). Elsevier. https://doi.org/10.1016/B978-0-12-820673-7.00004-4

US EPA, O. (2014, May 21). *Storm Water Management Model (SWMM)* [Data and Tools]. https://www.epa.gov/water-research/storm-water-management-model-swmm

USDA ARS Grassland Soil and Water Research Laboratory & Texas A&M AgriLife Research. (2018). *SWAT - Soil and Water Assessment Tool* [Dataset]. USDA Agricultural Research Service; Texas A&M AgriLife Research. https://data.nal.usda.gov/dataset/swat-soil-and-water-assessment-tool

Valipour, M. (2015). Long-term runoff study using SARIMA and ARIMA models in the United States. *Meteorological Applications*, *22*(3), 592–598. https://doi.org/10.1002/met.1491

Valipour, M., Banihabib, M. E., & Behbahani, S. M. R. (2013). Comparison of the ARMA, ARIMA, and the autoregressive artificial neural network models in forecasting the monthly inflow of

Dez dam reservoir. *Journal of Hydrology*, *476*, 433–441. https://doi.org/10.1016/j.jhydrol.2012.11.017

VanDerwarker, A. M., & Marcoux, J. B. (2018). *Principal Component Analysis*. https://doi.org/10.1002/9781119188230.saseas0477

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). *Attention Is All You Need* (arXiv:1706.03762). arXiv. https://doi.org/10.48550/arXiv.1706.03762

Vicente-Serrano, S. M., Peña-Gallardo, M., Hannaford, J., Murphy, C., Lorenzo-Lacruz, J., Dominguez-Castro, F., López-Moreno, J. I., Beguería, S., Noguera, I., Harrigan, S., & Vidal, J.-P. (2019). Climate, Irrigation, and Land Cover Change Explain Streamflow Trends in Countries Bordering the Northeast Atlantic. *Geophysical Research Letters*, *46*(19), 10821–10833. https://doi.org/10.1029/2019GL084084

Vogeti, R. K., Mishra, B. R., & Raju, K. S. (2022). Machine learning algorithms for streamflow forecasting of Lower Godavari Basin. *H2Open Journal*, *5*(4), 670–685. https://doi.org/10.2166/h2oj.2022.240

Wang, H., Stephenson, S. R., & Qu, S. (2020). Quantifying the relationship between streamflow and climate change in a small basin under future scenarios. *Ecological Indicators*, *113*, 106251. https://doi.org/10.1016/j.ecolind.2020.106251

Wang, K., Band, S. S., Ameri, R., Biyari, M., Hai, T., Hsu, C.-C., Hadjouni, M., Elmannai, H., Chau, K.-W., & Mosavi, A. (2022). Performance improvement of machine learning models via wavelet theory in estimating monthly river streamflow. *Engineering Applications of Computational Fluid Mechanics*, *16*(1), 1833–1848. https://doi.org/10.1080/19942060.2022.2119281

Wavelet Toolbox (Version 2022a). (2022). [Computer software]. The MathWorks Inc.

Wei, Y., Hashim, H., Chong, K. L., Huang, Y. F., Ahmed, A. N., & El-Shafie, A. (2023). Investigation of Meta-heuristics Algorithms in ANN Streamflow Forecasting. *KSCE Journal of Civil Engineering*, *27*(5), 2297–2312. https://doi.org/10.1007/s12205-023-0821-6

Wei, Y., Hashim, H., Lai, S., Chong, K. L., Huang, Y., Najah, A.-M., Sherif, M., & El-Shafie, A. (2024). Comparative Analysis of Artificial Intelligence Methods for Streamflow Forecasting. *IEEE Access*, *PP*, 1–1. https://doi.org/10.1109/ACCESS.2024.3351754

Whiteman, C. D. (2000). *Mountain Meteorology: Fundamentals and Applications*. Oxford University Press.

Wi, S. (2022). Assessing the Physical Realism of Deep Learning Hydrologic Model Projections Under Climate Change. *Water Resources Research*. https://doi.org/10.1029/2022wr032123

Xiang, Z., & Demir, İ. (2020). Distributed Long-Term Hourly Streamflow Predictions Using Deep Learning – A Case Study for State of Iowa. *Environmental Modelling & Software*. https://doi.org/10.1016/j.envsoft.2020.104761

Xie, M., Wang, B., Zhu, S., Ma, G., Yang, Z., Liu, B., & Jia, Y. (2022). Daily streamflow forecasting using hybrid long short-term memory model. *Journal of Physics: Conference Series*, *2271*(1), 012019. https://doi.org/10.1088/1742-6596/2271/1/012019

Y. Xiao & K. Wang. (2021). Research on Streamflow Forecast Based on EEMD and Long Short-Term Memory. *2021 International Conference on Artificial Intelligence and Electromechanical Automation (AIEA)*, 328–333. https://doi.org/10.1109/AIEA53260.2021.00076

Yang, S., Yang, D., Chen, J., Santisirisomboon, J., Lu, W., & Zhao, B. (2020). A physical process and machine learning combined hydrological model for daily streamflow simulations of large watersheds with limited observation data. *Journal of Hydrology*, *590*, 125206. https://doi.org/10.1016/j.jhydrol.2020.125206

Yaseen, Z. M., El-shafie, A., Jaafar, O., Afan, H. A., & Sayl, K. N. (2015). Artificial intelligence based models for stream-flow forecasting: 2000–2015. *Journal of Hydrology*, *530*, 829–844. https://doi.org/10.1016/j.jhydrol.2015.10.038

Yaseen, Z. M., Jaafar, O., Deo, R. C., Kisi, O., Adamowski, J., Quilty, J., & El-Shafie, A. (2016). Stream-flow forecasting using extreme learning machines: A case study in a semi-arid region in Iraq. *Journal of Hydrology*, *542*, 603–614. https://doi.org/10.1016/j.jhydrol.2016.09.035

Yaseen, Z. M., Sulaiman, S. O., Deo, R. C., & Chau, K.-W. (2019). An enhanced extreme learning machine model for river flow forecasting: State-of-the-art, practical applications in water resource engineering area and future research direction. *Journal of Hydrology*, *569*, 387–408. https://doi.org/10.1016/j.jhydrol.2018.11.069

Yilmaz, M., Tosunoğlu, F., Kaplan, N. H., Üneş, F., & Hanay, Y. S. (2022). Predicting monthly streamflow using artificial neural networks and wavelet neural networks models. *Modeling Earth Systems and Environment*, *8*(4), 5547–5563. https://doi.org/10.1007/s40808-022-01403-9

You, J. (2022). *hydroIVS* [Computer software]. https://github.com/johnswyou/hydroIVS

You, J. (2023). *Python Wavelet Data Driven Forecasting Framework* [Computer software]. https://github.com/johnswyou/pywddff

Yu, H., Wen, X., Feng, Q., Deo, R. C., Si, J., & Wu, M. (2018). Comparative Study of Hybrid-Wavelet Artificial Intelligence Models for Monthly Groundwater Depth Forecasting in Extreme Arid Regions, Northwest China. *Water Resources Management*, *32*(1), 301–323. https://doi.org/10.1007/s11269-017-1811-6

Yu, X., Wang, Y., Wu, L., Chen, G., Wang, L., & Qin, H. (2020). Comparison of support vector regression and extreme gradient boosting for decomposition-based data-driven 10-day

streamflow forecasting. *Journal of Hydrology*, *582*, 124293. https://doi.org/10.1016/j.jhydrol.2019.124293

Yuan, L., & Forshay, K. J. (2022). Evaluating Monthly Flow Prediction Based on SWAT and Support Vector Regression Coupled with Discrete Wavelet Transform. *Water*, *14*(17). https://doi.org/10.3390/w14172649

Zakhrouf, M., Bouchelkia, H., Stamboul, M., Kim, S., & Singh, V. P. (2020). Implementation on the evolutionary machine learning approaches for streamflow forecasting: Case study in the Seybous River Algeria. *Journal of Korea Water Resources Association*, *53*(6), 395–408. https://doi.org/10.3741/JKWRA.2020.53.6.395

Zeng, A., Chen, M., Zhang, L., & Xu, Q. (2022). *Are Transformers Effective for Time Series Forecasting?* (arXiv:2205.13504). arXiv. https://doi.org/10.48550/arXiv.2205.13504

Zounemat-Kermani, M., Batelaan, O., Fadaee, M., & Hinkelmann, R. (2021). Ensemble machine learning paradigms in hydrology: A review. *Journal of Hydrology*, *598*, 126266. https://doi.org/10.1016/j.jhydrol.2021.126266

Appendix A

Select Highly Cited Wavelet Transform-based Streamflow Forecasting Papers (Scopus)

Authors	Title	Year
Adamowski J.F.	Development of a short-term river flood forecasting method for snowmelt driven floods based on wavelet and cross-wavelet analysis	2008
Adamowski J.F.	River flow forecasting using wavelet and cross-wavelet transform models	2008
Kisi O.; Cimen M.	A wavelet-support vector machine conjunction model for monthly streamflow forecasting	2011
Zhang Q.; Xu C.-Y.; Jiang T.; Wu Y.	Possible influence of ENSO on annual maximum streamflow of the Yangtze River, China	2007
Maheswaran R.; Khosa R.	Comparative study of different wavelets for hydrologic forecasting	2012
Kişi Ö.	Neural networks and wavelet conjunction model for intermittent streamflow forecasting	2009
Shiri J.; Kisi O.	Short-term and long-term streamflow forecasting using a wavelet and neuro-fuzzy conjunction model	2010
Guo J.; Zhou J.; Qin H.; Zou Q.; Li Q.	Monthly streamflow forecasting based on improved support vector machine model	2011
Karthikeyan L.; Nagesh Kumar D.	Predictability of nonstationary time series using wavelet and EMD based ARMA models	2013
Danandeh Mehr A.; Kahya E.; Olyaie E.	Streamflow prediction using linear genetic programming in comparison with a neuro-wavelet technique	2013

Prasad R.; Deo R.C.; Li Y.; Maraseni T.	Input selection and performance optimization of ANN-based streamflow forecasts in the drought-prone Murray Darling Basin region using IIS and MODWT algorithm	2017
Kasiviswanathan K.S.; He J.; Sudheer K.P.; Tay J.-H.	Potential application of wavelet neural network ensemble to forecast streamflow for flood management	2016
Ni L.; Wang D.; Singh V.P.; Wu J.; Wang Y.; Tao Y.; Zhang J.	Streamflow and rainfall forecasting by two long short-term memory-based models	2020
Zhang X.; Peng Y.; Zhang C.; Wang B.	Are hybrid models integrated with data preprocessing techniques suitable for monthly streamflow forecasting? Some experiment evidences	2015
Kişi Ö.	Stream flow forecasting using neuro-wavelet technique	2008
Zuo G.; Luo J.; Wang N.; Lian Y.; He X.	Decomposition ensemble model based on variational mode decomposition and long short-term memory for streamflow forecasting	2020
Kisi O.	Wavelet regression model for short-term streamflow forecasting	2010
Guimarães Santos C.A.; Silva G.B.L.	Daily streamflow forecasting using a wavelet transform and artificial neural network hybrid models	2014
Fang W.; Huang S.; Ren K.; Huang Q.; Huang G.; Cheng G.; Li K.	Examining the applicability of different sampling techniques in the development of decomposition-based streamflow forecasting models	2019

Nanda T.; Sahoo B.; Beria H.; Chatterjee C.	A wavelet-based non-linear autoregressive with exogenous inputs (WNARX) dynamic neural network model for real-time flood forecasting using satellite-based rainfall products	2016
Zhu S.; Zhou J.; Ye L.; Meng C.	Streamflow estimation by support vector machine coupled with different methods of time series decomposition in the upper reaches of Yangtze River, China	2016
Rathinasamy M.; Adamowski J.; Khosa R.	Multiscale streamflow forecasting using a new Bayesian Model Average based ensemble multi-wavelet Volterra nonlinear method	2013
Peng T.; Zhou J.; Zhang C.; Fu W.	Streamflow forecasting using empirical wavelet transform and artificial neural networks	2017
Adamowski J.F.; Karen Sun	Development of a coupled wavelet transform and neural network method for flow forecasting of non-perennial rivers in semi-arid watersheds	2010

Appendix B

Filtered Catchments

See Newman et al. (2015).

One Day Ahead Forecast Horizon

Index	Gauge ID	Gauge Name
0	1134500	MOOSE RIVER AT VICTORY, VT
1	1486000	MANOKIN BRANCH NEAR PRINCESS ANNE, MD
2	1542810	WALDY RUN NEAR EMPORIUM, PA
3	2215100	TUCSAWHATCHEE CREEK NEAR HAWKINSVILLE, GA
4	2216180	TURNPIKE CREEK NEAR MCRAE, GA
5	2299950	MANATEE RIVER NEAR MYAKKA HEAD FL
6	2349900	TURKEY CREEK AT BYROMVILLE, GA
7	2465493	ELLIOTTS CREEK AT MOUNDVILLE AL
8	4043050	TRAP ROCK RIVER NEAR LAKE LINDEN, MI
9	4105700	AUGUSTA CREEK NEAR AUGUSTA, MI
10	4296000	BLACK RIVER AT COVENTRY, VT
11	6221400	DINWOODY CREEK ABOVE LAKES, NEAR BURRIS, WYO.
12	6224000	BULL LAKE CREEK ABOVE BULL LAKE, WY
13	6278300	SHELL CREEK ABOVE SHELL CREEK RESERVOIR, WY

14	6280300	SOUTH FORK SHOSHONE RIVER NEAR VALLEY, WY
15	6311000	NORTH FORK POWDER RIVER NEAR HAZELTON, WY
16	6614800	MICHIGAN RIVER NEAR CAMERON PASS, CO.
17	6623800	ENCAMPMENT RIVER AB HOG PARK CR, NR ENCAMPMENT...
18	6746095	JOE WRIGHT CREEK ABOVE JOE WRIGHT RESERVOIR, CO.
19	7083000	HALFMOON CREEK NEAR MALTA, CO.
20	7208500	RAYADO CREEK NEAR CIMARRON, NM
21	8267500	RIO HONDO NEAR VALDEZ, NM
22	8269000	RIO PUEBLO DE TAOS NEAR TAOS, NM
23	8271000	RIO LUCERO NEAR ARROYO SECO, NM
24	9034900	BOBTAIL CREEK NEAR JONES PASS, CO.
25	9035800	DARLING CREEK NEAR LEAL, CO.
26	9035900	SOUTH FORK OF WILLIAMS FORK NEAR LEAL, CO.
27	9047700	KEYSTONE GULCH NEAR DILLON, CO.
28	9065500	GORE CREEK AT UPPER STATION, NEAR MINTURN, CO.
29	9066000	BLACK GORE CREEK NEAR MINTURN, CO.
30	9066200	BOOTH CREEK NEAR MINTURN, CO.
31	9066300	MIDDLE CREEK NEAR MINTURN, CO.

32	9081600	CRYSTAL RIVER AB AVALANCHE C, NEAR REDSTONE, CO.
33	9107000	TAYLOR RIVER AT TAYLOR PARK, CO.
34	9210500	FONTENELLE C NR HERSCHLER RANCH, NR FONTENELLE...
35	9223000	HAMS FORK BELOW POLE CREEK, NEAR FRONTIER, WY
36	9352900	VALLECITO CREEK NEAR BAYFIELD, CO.
37	10172800	SOUTH WILLOW CREEK NEAR GRANTSVILLE, UT
38	10205030	SALINA CREEK NEAR EMERY, UT
39	10242000	COAL CREEK NEAR CEDAR CITY, UT
40	10249300	S TWIN R NR ROUND MOUNTAIN, NV
41	10316500	LAMOILLE C NR LAMOILLE, NV
42	10336645	GENERAL C NR MEEKS BAY CA
43	10336660	BLACKWOOD C NR TAHOE CITY CA
44	11230500	BEAR C NR LAKE THOMAS A EDISON CA
45	11237500	PITMAN C BL TAMARACK C CA
46	11264500	MERCED R A HAPPY ISLES BRIDGE NR YOSEMITE CA
47	11266500	MERCED R A POHONO BRIDGE NR YOSEMITE CA
48	11475560	ELDER C NR BRANSCOMB CA
49	11476600	BULL C NR WEOTT CA
50	11480390	MAD R AB RUTH RES NR FOREST GLEN CA
51	11481200	LITTLE R NR TRINIDAD CA

52	12010000	NASELLE RIVER NEAR NASELLE, WA
53	12013500	WILLAPA RIVER NEAR WILLAPA, WA
54	12020000	CHEHALIS RIVER NEAR DOTY, WA
55	12025700	SKOOKUMCHUCK RIVER NEAR VAIL, WA
56	12073500	HUGE CREEK NEAR WAUNA, WA
57	12114500	CEDAR RIVER BELOW BEAR CREEK NEAR CEDAR FALLS, WA
58	12115500	REX RIVER NEAR CEDAR FALLS, WA
59	12117000	TAYLOR CREEK NEAR SELLECK, WA
60	12145500	RAGING RIVER NEAR FALL CITY, WA
61	12147500	NORTH FORK TOLT RIVER NEAR CARNATION, WA
62	12175500	THUNDER CREEK NEAR NEWHALEM, WA
63	12375900	SOUTH CROW CREEK NEAR RONAN MT
64	12377150	MISSION CR AB RESERVOIR NR ST IGNATIUS MT
65	12381400	SOUTH FORK JOCKO RIVER NEAR ARLEE MT
66	12383500	BIG KNIFE CREEK NEAR ARLEE MT
67	12388400	REVAIS CR BL WEST FORK NR DIXON MT
68	12447390	ANDREWS CREEK NEAR MAZAMA, WA
69	12488500	AMERICAN RIVER NEAR NILE, WA
70	13011500	PACIFIC CREEK AT MORAN WY
71	13011900	BUFFALO FORK AB LAVA CREEK NR MORAN WY
72	13018300	CACHE CREEK NEAR JACKSON, WY
73	13023000	GREYS RIVER AB RESERVOIR NR ALPINE WY

74	13083000	TRAPPER CREEK NR OAKLEY ID
75	13161500	BRUNEAU RIVER AT ROWLAND NV
76	13235000	SF PAYETTE RIVER AT LOWMAN ID
77	13240000	LAKE FORK PAYETTE RIVER AB JUMBO CR NR MCCALL ID
78	13313000	JOHNSON CREEK AT YELLOW PINE ID
79	14138800	BLAZED ALDER CREEK NEAR RHODODENDRON, OREG.
80	14138870	FIR CREEK NEAR BRIGHTWOOD, OR
81	14138900	NORTH FORK BULL RUN RIVER NEAR MULTNOMAH FALLS...
82	14139800	SOUTH FORK BULL RUN RIVER NEAR BULL RUN, OR
83	14141500	LITTLE SANDY RIVER NEAR BULL RUN, OREG.
84	14187000	WILEY CREEK NEAR FOSTER, OR
85	14236200	TILTON RIVER AB BEAR CANYON CREEK NEAR CINEBAR...
86	14303200	TUCCA CREEK NEAR BLAINE, OR.
87	14306340	EAST FORK LOBSTER CREEK NEAR ALSEA, OR.
88	14309500	WEST FORK COW CREEK NEAR GLENDALE, OR
89	14362250	STAR GULCH NEAR RUCH, OR.

Three Days Ahead Forecast Horizon

Index	Gauge ID	Gauge Name
0	2298123	PRAIRIE CREEK NEAR FORT OGDEN FL
1	4057800	MIDDLE BRANCH ESCANABA RIVER AT HUMBOLDT, MI
2	6221400	DINWOODY CREEK ABOVE LAKES, NEAR BURRIS, WYO.
3	6224000	BULL LAKE CREEK ABOVE BULL LAKE, WY
4	6278300	SHELL CREEK ABOVE SHELL CREEK RESERVOIR, WY
5	6280300	SOUTH FORK SHOSHONE RIVER NEAR VALLEY, WY
6	6291500	LODGE GRASS CR AB WILLOW C DIV NR WYOLA MT
7	6311000	NORTH FORK POWDER RIVER NEAR HAZELTON, WY
8	6614800	MICHIGAN RIVER NEAR CAMERON PASS, CO.
9	6632400	ROCK CREEK AB KING CANYON CANAL, NR ARLINGTON, WY
10	6746095	JOE WRIGHT CREEK ABOVE JOE WRIGHT RESERVOIR, CO.
11	7083000	HALFMOON CREEK NEAR MALTA, CO.

12	7208500	RAYADO CREEK NEAR CIMARRON, NM
13	8267500	RIO HONDO NEAR VALDEZ, NM
14	8271000	RIO LUCERO NEAR ARROYO SECO, NM
15	9034900	BOBTAIL CREEK NEAR JONES PASS, CO.
16	9035800	DARLING CREEK NEAR LEAL, CO.
17	9035900	SOUTH FORK OF WILLIAMS FORK NEAR LEAL, CO.
18	9047700	KEYSTONE GULCH NEAR DILLON, CO.
19	9065500	GORE CREEK AT UPPER STATION, NEAR MINTURN, CO.
20	9066000	BLACK GORE CREEK NEAR MINTURN, CO.
21	9066200	BOOTH CREEK NEAR MINTURN, CO.
22	9066300	MIDDLE CREEK NEAR MINTURN, CO.
23	9081600	CRYSTAL RIVER AB AVALANCHE C, NEAR REDSTONE, CO.
24	9107000	TAYLOR RIVER AT TAYLOR PARK, CO.
25	9210500	FONTENELLE C NR HERSCHLER RANCH, NR FONTENELLE...
26	9223000	HAMS FORK BELOW POLE CREEK, NEAR FRONTIER, WY
27	9352900	VALLECITO CREEK NEAR BAYFIELD, CO.
28	10172800	SOUTH WILLOW CREEK NEAR GRANTSVILLE, UT
29	10205030	SALINA CREEK NEAR EMERY, UT
30	10242000	COAL CREEK NEAR CEDAR CITY, UT

31	10249300	S TWIN R NR ROUND MOUNTAIN, NV
32	10316500	LAMOILLE C NR LAMOILLE, NV
33	10329500	MARTIN C NR PARADISE VALLEY, NV
34	10336645	GENERAL C NR MEEKS BAY CA
35	10336660	BLACKWOOD C NR TAHOE CITY CA
36	10343500	SAGEHEN C NR TRUCKEE CA
37	11230500	BEAR C NR LAKE THOMAS A EDISON CA
38	11475560	ELDER C NR BRANSCOMB CA
39	11480390	MAD R AB RUTH RES NR FOREST GLEN CA
40	12010000	NASELLE RIVER NEAR NASELLE, WA
41	12025700	SKOOKUMCHUCK RIVER NEAR VAIL, WA
42	12175500	THUNDER CREEK NEAR NEWHALEM, WA
43	12375900	SOUTH CROW CREEK NEAR RONAN MT
44	12377150	MISSION CR AB RESERVOIR NR ST IGNATIUS MT
45	12381400	SOUTH FORK JOCKO RIVER NEAR ARLEE MT
46	12383500	BIG KNIFE CREEK NEAR ARLEE MT
47	12388400	REVAIS CR BL WEST FORK NR DIXON MT
48	12390700	PROSPECT CREEK AT THOMPSON FALLS MT
49	12447390	ANDREWS CREEK NEAR MAZAMA, WA
50	12488500	AMERICAN RIVER NEAR NILE, WA
51	13011500	PACIFIC CREEK AT MORAN WY
52	13011900	BUFFALO FORK AB LAVA CREEK NR MORAN WY
53	13018300	CACHE CREEK NEAR JACKSON, WY

54	13083000	TRAPPER CREEK NR OAKLEY ID
55	13161500	BRUNEAU RIVER AT ROWLAND NV
56	13240000	LAKE FORK PAYETTE RIVER AB JUMBO CR NR MCCALL ID
57	13313000	JOHNSON CREEK AT YELLOW PINE ID
58	14020000	UMATILLA RIVER ABOVE MEACHAM CREEK, NR GIBBON, OR
59	14166500	LONG TOM RIVER NEAR NOTI, OREG

Appendix C

Koppen Climate Classification Summary Table

See Pidwirny (2021).

Koppen Code	Description	Detailed Description
Af	Tropical rainforest	Tropical rainforest: No dry season, all months have average precipitation of at least 60 mm
Am	Tropical monsoon	Tropical monsoon: Short dry season with the driest month having less than 60 mm of precipitation and less than 1/25 of the total annual precipitation
Aw	Tropical savanna	Tropical savanna: Distinct dry season with the driest month having less than 60 mm of precipitation
BWh	Desert (hot)	Desert (hot): Arid climate with very high temperatures, annual precipitation less than 250 mm
BWk	Desert (cold)	Desert (cold): Arid climate with lower temperatures, annual precipitation less than 250 mm
BSh	Steppe (hot)	Steppe (hot): Semi-arid climate with high temperatures, annual precipitation between 250 and 500 mm
BSk	Steppe (cold)	Steppe (cold): Semi-arid climate with lower temperatures, annual precipitation between 250 and 500 mm
Csa	Mediterranean hot summer	Mediterranean hot summer: Hot, dry summers and mild, wet winters, average temperature of the warmest month above 22°C

Csb	Mediterranean warm summer	Mediterranean warm summer: Warm, dry summers and mild, wet winters, average temperature of the warmest month below 22°C
Csc	Mediterranean cold summer	Mediterranean cold summer: Mild, wet winters and cool, dry summers, average temperature of the warmest month below 22°C
Cwa	Temperate monsoon hot summer	Temperate monsoon hot summer: Hot summers and mild winters with distinct wet and dry seasons
Cwb	Temperate monsoon warm summer	Temperate monsoon warm summer: Warm summers and mild winters with distinct wet and dry seasons
Cwc	Temperate monsoon cold summer	Temperate monsoon cold summer: Cool summers and mild winters with distinct wet and dry seasons
Cfa	Humid subtropical hot summer	Humid subtropical hot summer: Hot, humid summers and mild winters with no dry season
Cfb	Humid subtropical warm summer	Humid subtropical warm summer: Warm, humid summers and mild winters with no dry season
Cfc	Humid subtropical cold summer	Humid subtropical cold summer: Cool, humid summers and mild winters with no dry season
Dsa	Continental hot summer	Continental hot summer: Hot summers and cold winters with dry summers
Dsb	Continental warm summer	Continental warm summer: Warm summers and cold winters with dry summers
Dsc	Continental cold summer	Continental cold summer: Cool summers and cold winters with dry summers

Dwa	Monsoon-influenced hot summer	Monsoon-influenced hot summer: Hot summers and cold winters with dry winters
Dwb	Monsoon-influenced warm summer	Monsoon-influenced warm summer: Warm summers and cold winters with dry winters
Dwc	Monsoon-influenced cold summer	Monsoon-influenced cold summer: Cool summers and cold winters with dry winters
Dfa	Humid continental hot summer	Humid continental hot summer: Hot summers and cold winters with no dry season
Dfb	Humid continental warm summer	Humid continental warm summer: Warm summers and cold winters with no dry season
Dfc	Humid continental cold summer	Humid continental cold summer: Cool summers and cold winters with no dry season
Dfd	Humid continental severe winter	Humid continental severe winter: Extremely cold winters and no dry season
Dsd	Continental severe winter, dry summer	Continental severe winter, dry summer: Extremely cold winters and dry summers
Dwd	Continental severe winter, dry winter	Continental severe winter, dry winter: Extremely cold winters and dry winters
ET	Tundra	Tundra: Cold climate with at least one month having an average temperature above 0°C but no month above 10°C
EF	Ice cap	Ice cap: Very cold climate with no month having an average temperature above 0°C

Appendix D

Hydrographs

One Day Ahead Forecast Horizon

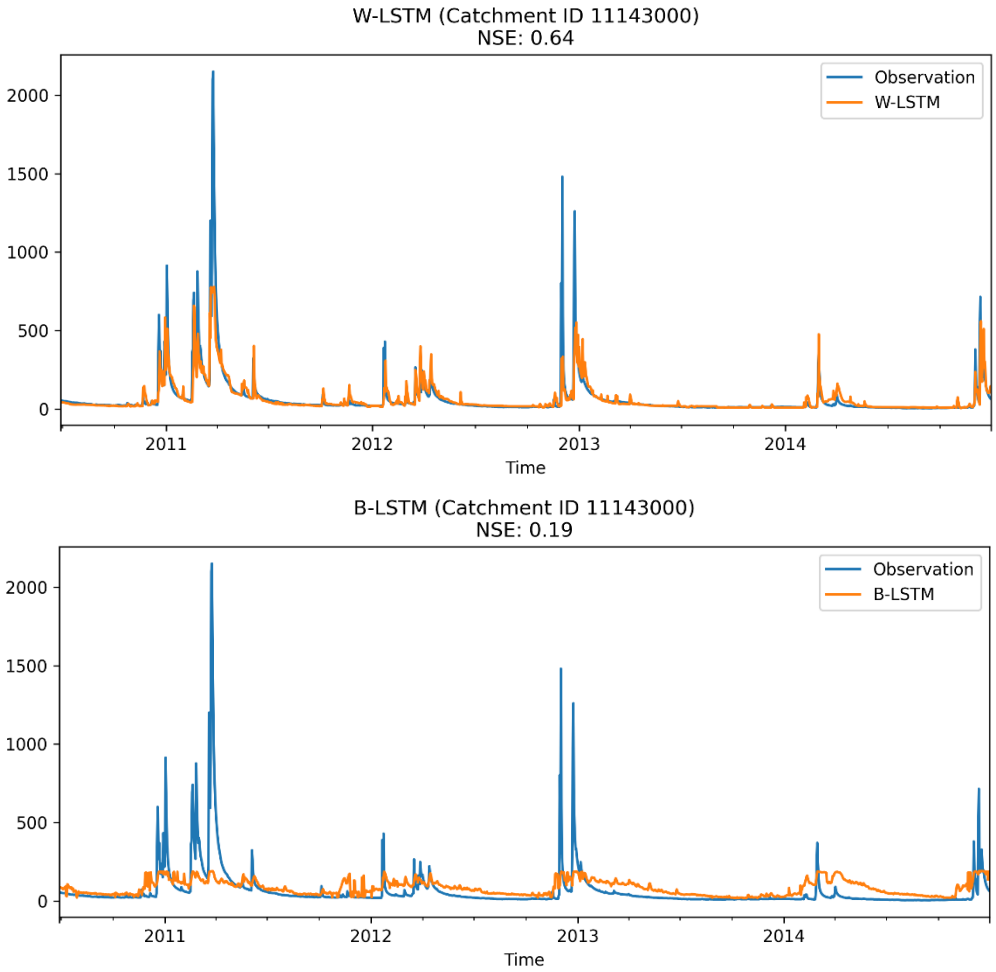


Figure 31. Comparison of optimal W-LSTM (top) to B-LSTM (bottom) for one day ahead streamflow forecasting (mm/day) of Big Sur River near Big Sur, California (United States). The blue line shows observed streamflow, while the orange line shows the forecast.

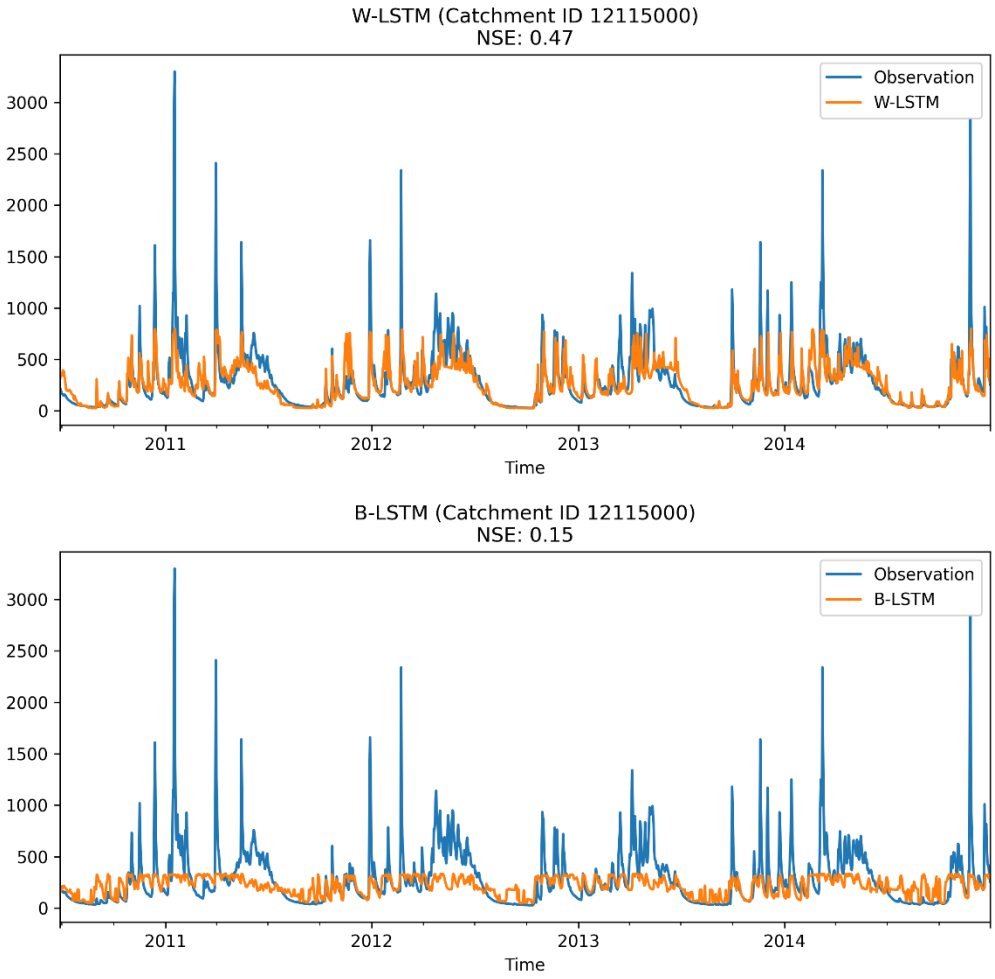


Figure 32. Comparison of optimal W-LSTM (top) to B-LSTM (bottom) for one day ahead streamflow forecasting (mm/day) of Cedar River near Cedar Falls, Washington (United States). The blue line shows observed streamflow, while the orange line shows the forecast.

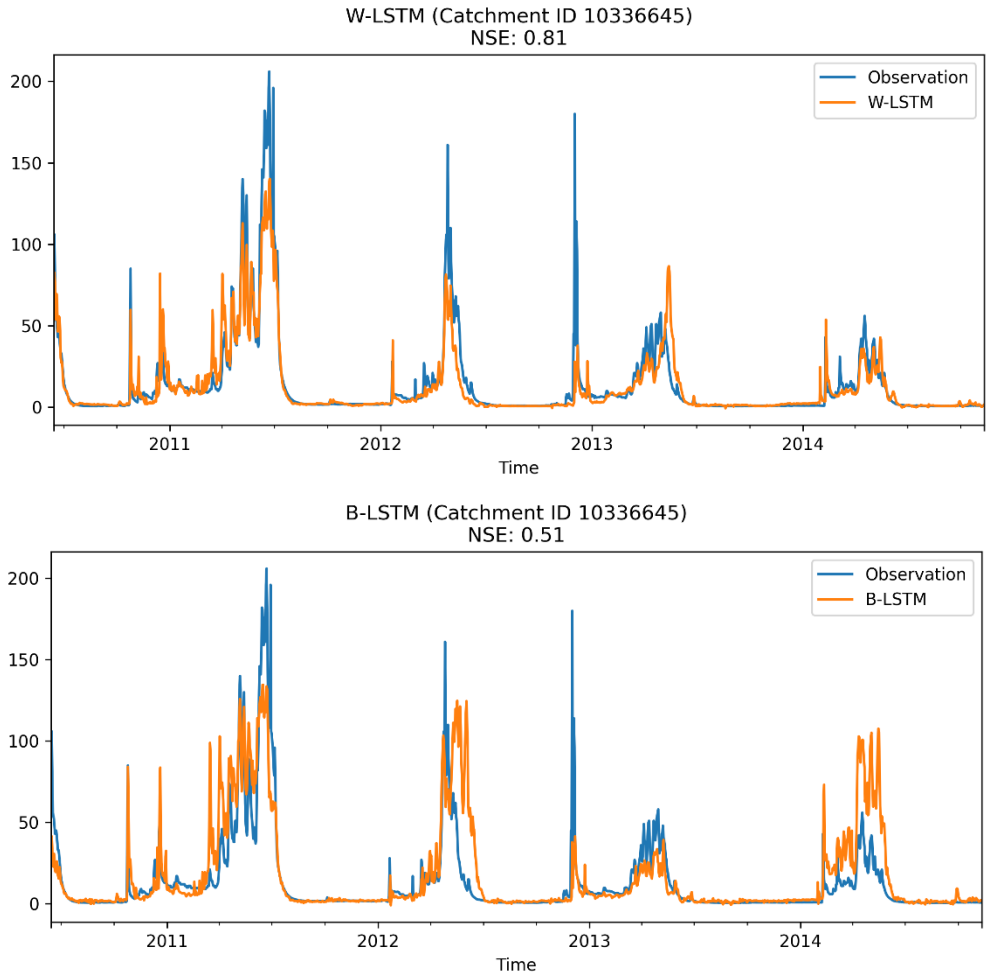


Figure 33. Comparison of optimal W-LSTM (top) to B-LSTM (bottom) for one day ahead streamflow forecasting (mm/day) of General Creek near Meeks Bay, California (United States). The blue line shows observed streamflow, while the orange line shows the forecast.

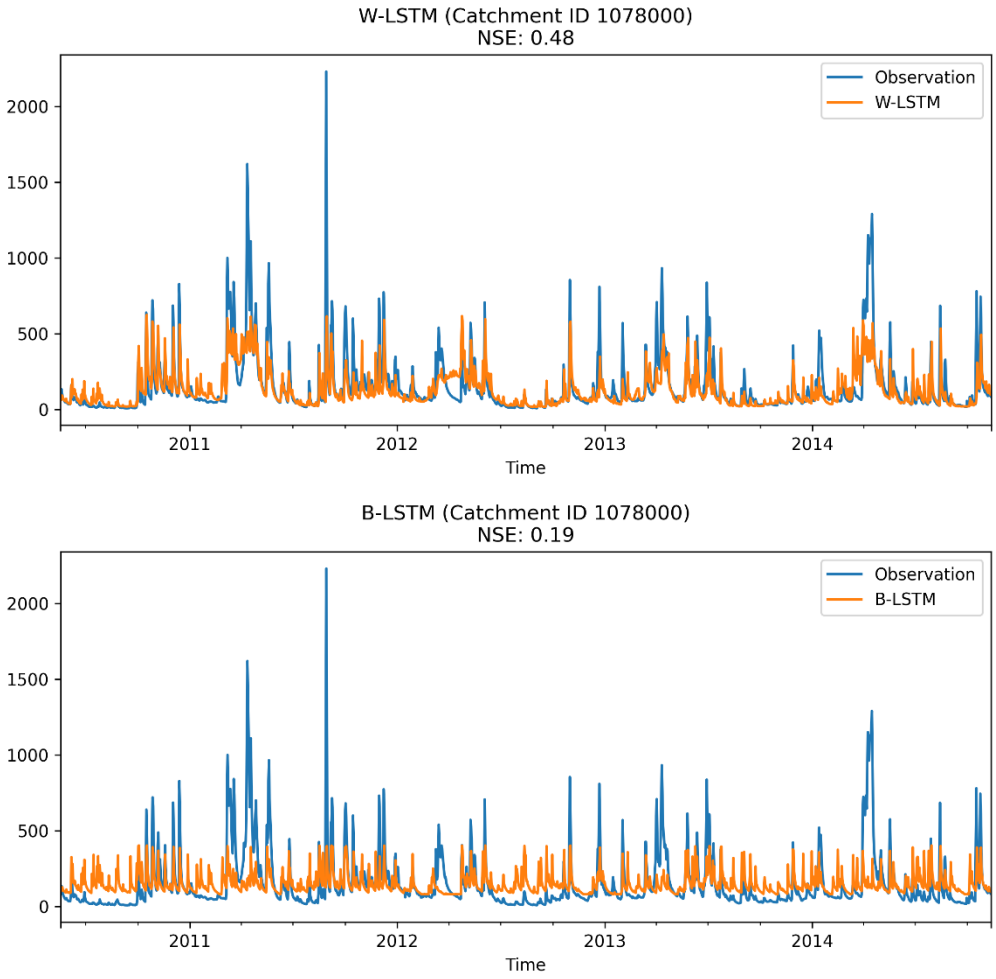


Figure 34. Comparison of optimal W-LSTM (top) to B-LSTM (bottom) for one day ahead streamflow forecasting (mm/day) of Assabet River near Maynard, Massachusetts (United States). The blue line shows observed streamflow, while the orange line shows the forecast.

3 Day Forecast Horizon

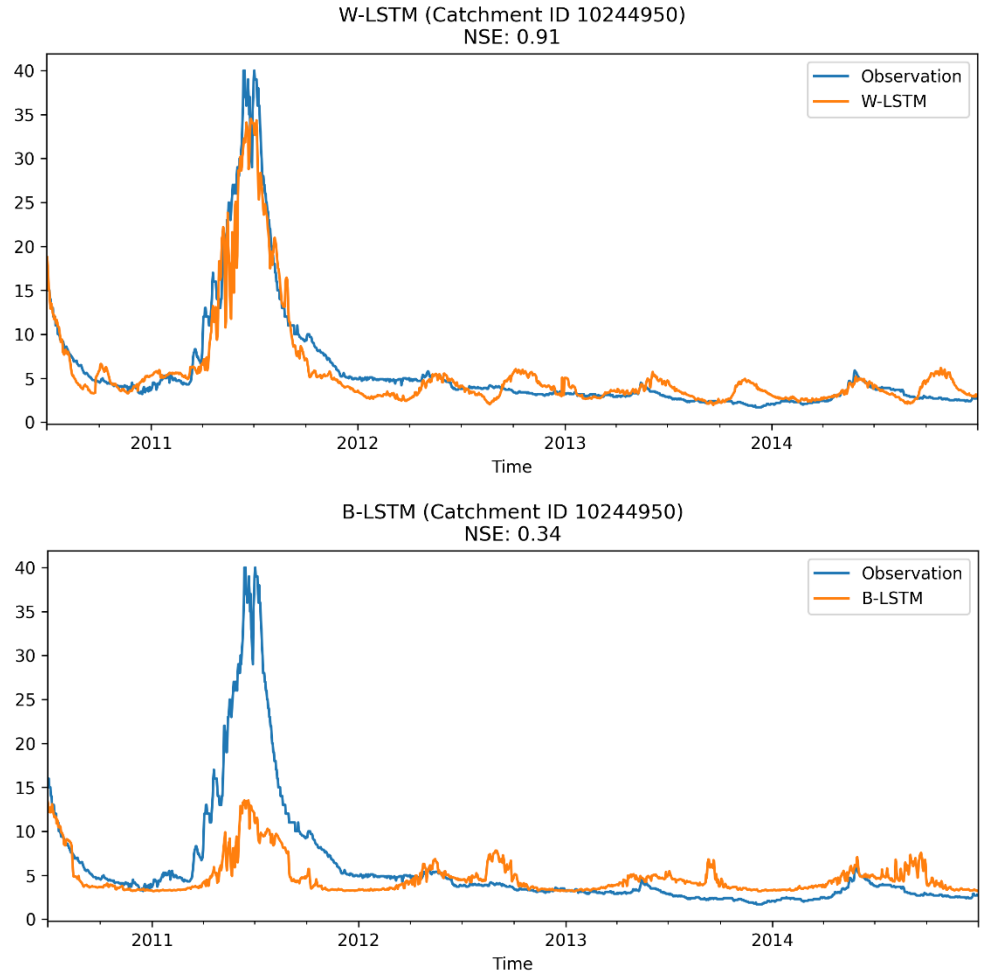


Figure 35. Comparison of optimal W-LSTM (top) to B-LSTM (bottom) for three days ahead streamflow forecasting (mm/day) of Steptoe Creek near Ely, Nevada (United States). The blue line shows observed streamflow, while the orange line shows the forecast.

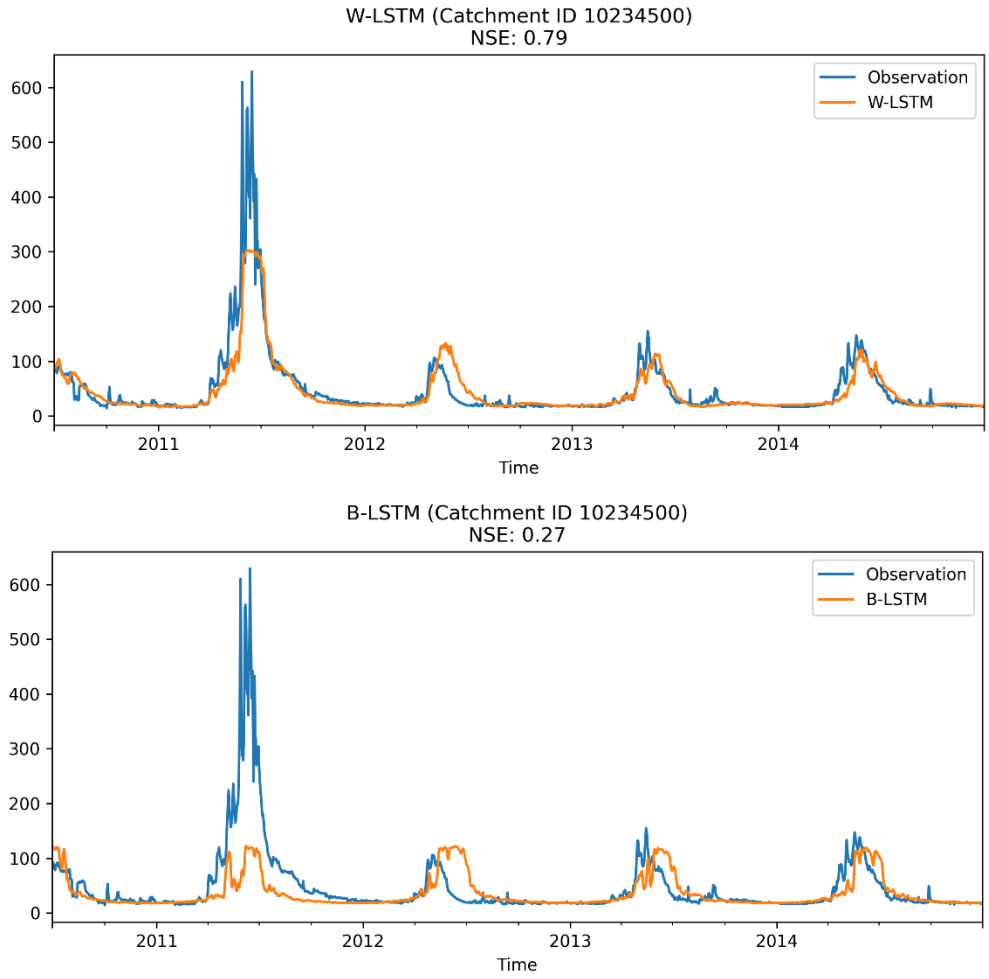


Figure 36. Comparison of optimal W-LSTM (top) to B-LSTM (bottom) for three days ahead streamflow forecasting (mm/day) of Beaver River near Beaver, Utah (United States). The blue line shows observed streamflow, while the orange line shows the forecast.

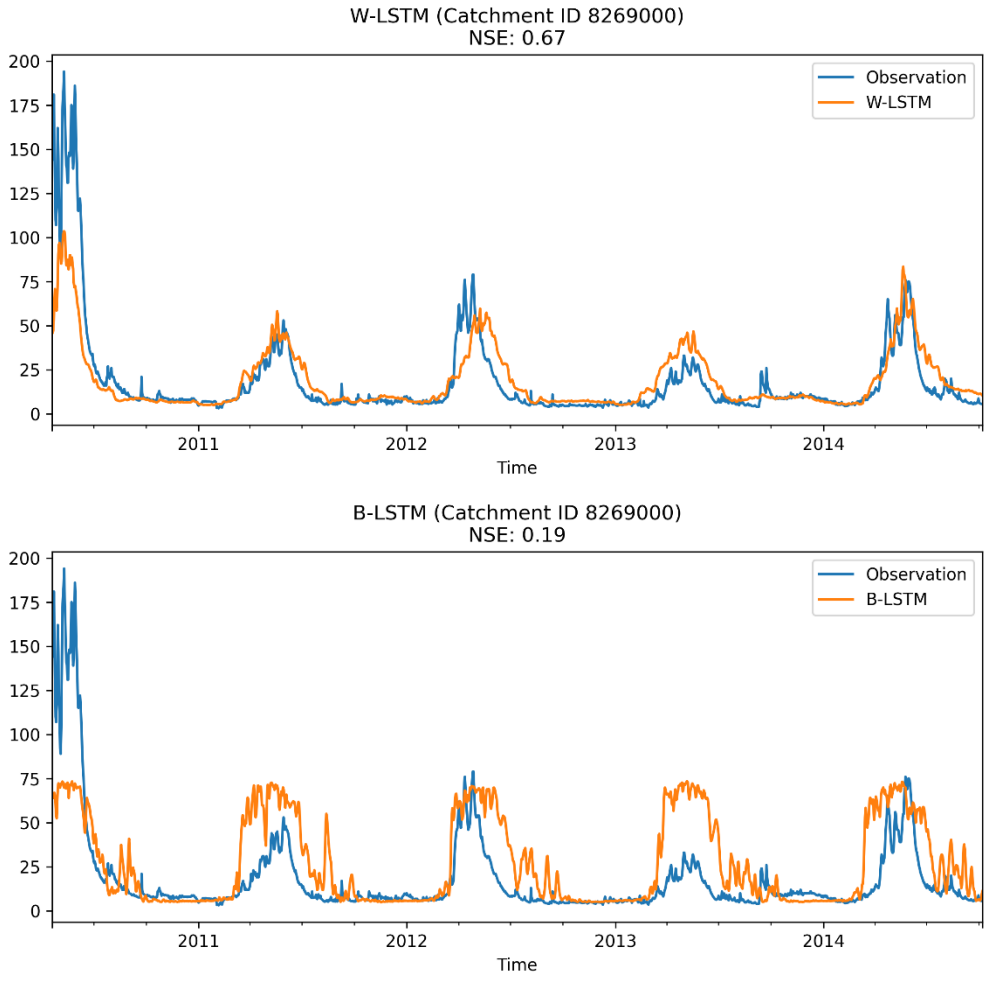


Figure 37. Comparison of optimal W-LSTM (top) to B-LSTM (bottom) for three days ahead streamflow forecasting (mm/day) of Salmon River near Shoup, Idaho (United States). The blue line shows observed streamflow, while the orange line shows the forecast.

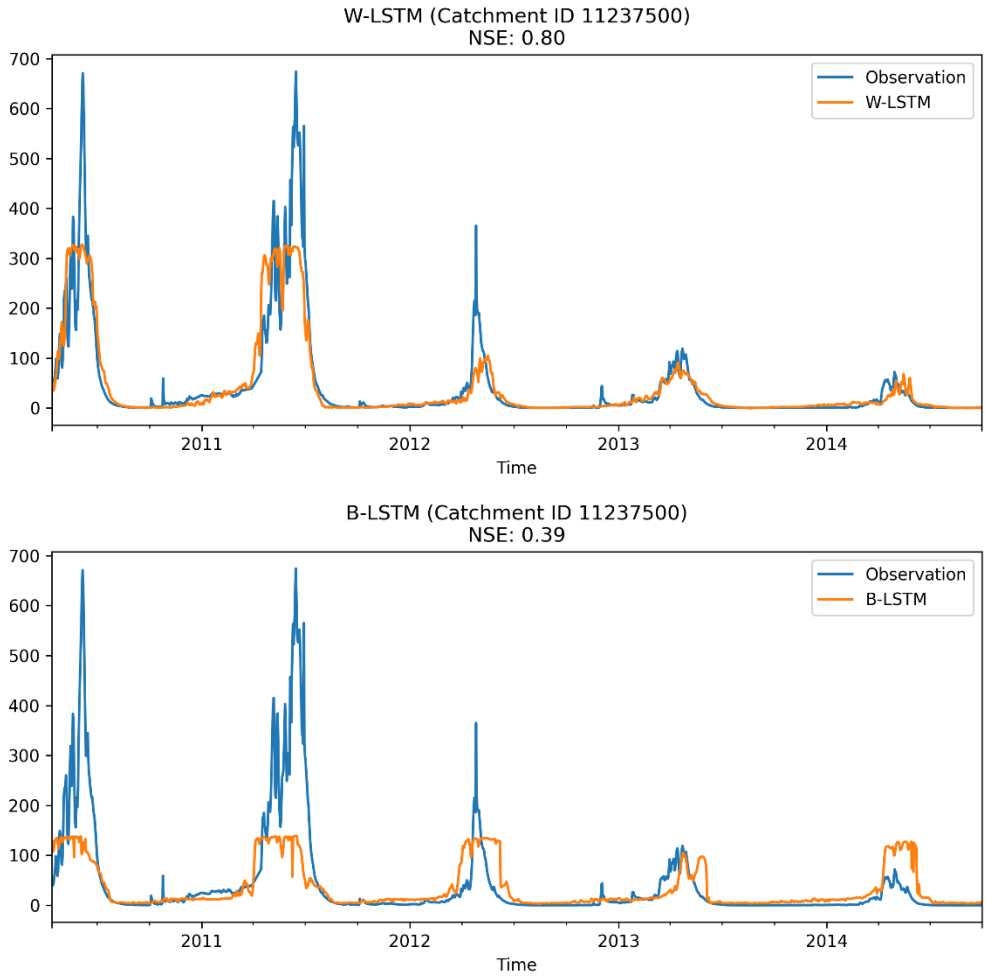


Figure 38. Comparison of optimal W-LSTM (top) to B-LSTM (bottom) for three days ahead streamflow forecasting (mm/day) of Merced River at Happy Isles Bridge in Yosemite National Park, California (United States). The blue line shows observed streamflow, while the orange line shows the forecast.

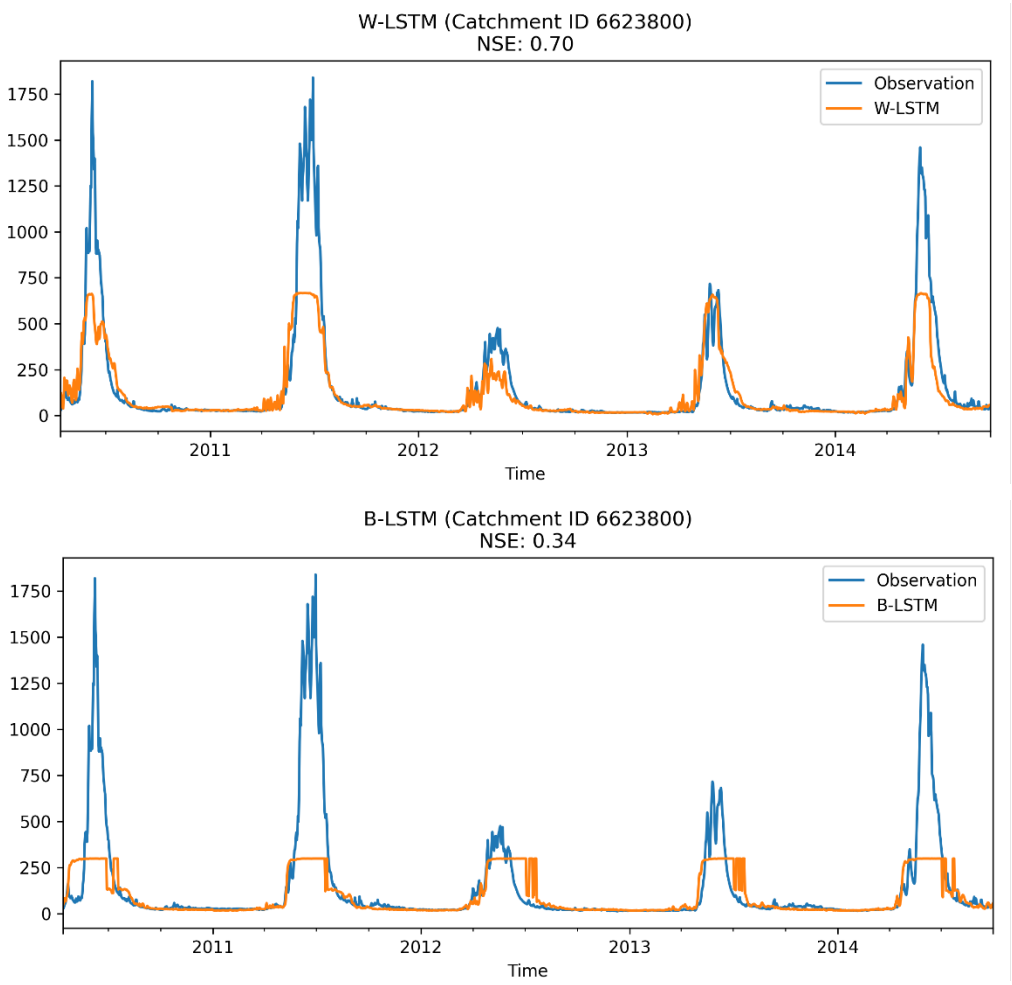


Figure 39. Comparison of optimal W-LSTM (top) to B-LSTM (bottom) for three days ahead streamflow forecasting (mm/day) of Encampment River, upstream of Hog Park Creek, near the town of Encampment in Carbon County, Wyoming (United States). The blue line shows observed streamflow, while the orange line shows the forecast.