DEGREE PROJECT IN TECHNOLOGY,
SECOND CYCLE, 30 CREDITS
STOCKHOLM, SWEDEN 2022

FLEX: Force Linear to Exponential

Improving Time Series Forecasting Models For Hydrological Level Using A Scalable Ensemble Machine Learning Approach

Koen van den Brink

KTH ROYAL INSTITUTE OF TECHNOLOGY
ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

Author

Koen van den Brink <koenvdb@kth.se>
Faculty of Electrical Engineering and Computer Science (Data Science)
KTH Royal Institute of Technology

Place for Project

Stockholm, Sweden
KTH

Company

Gecosistema
Italy

Examiner

Henrik Boström
Stockholm, Sweden
KTH Royal Institute of Technology

Supervisor

Nancy Xu

Stockholm, Sweden

KTH Royal Institute of Technology

Abstract

Time-series forecasting is an area of machine learning that can be applied to many real-life problems. It is used in areas such as water level forecasting, which aims to help people evacuate on time for floods. This thesis aims to contribute to the research area of time-series forecasting, by introducing a simple but novel ensemble model: Force Linear to Exponential (FLEX). A FLEX ensemble first forecasts points that are exponentially further into the forecasting horizon. After this, the gaps between forecasted points are produced from said forecasted points, as well as the entire data history. This simple model is able to outperform all base models considered in this thesis, even when having the same amount of parameters to tune.

Keywords

machine learning; time-series forecasting; water level forecasting; time-series transformers; recurrent neural networks

Sammanfattning

Tidsserieprognoser är ett område för maskininlärning som kan tillämpas på många verkliga problem. Det används i områden som vattenståndsprognoser, som syftar till att hjälpa människor att evakuera i tid för översvämningar. Denna uppsats syftar till att bidra till forskningsområdet tidsserieprognoser genom att introducera en enkel men ny ensemblemodell: Force Linear to Exponential (FLEX). En FLEX-ensemble prognostiserar först punkter som ligger exponentiellt längre in i prognoshorisonten. Efter detta produceras gapen mellan prognostiserade punkter från nämnda prognostiserade punkter, såväl som hela datahistoriken. Denna enkla modell kan överträffa alla basmodeller som behandlas i denna uppsats, även när den har samma mängd parametrar att ställa in.

Nyckelord

maskininlärning; tidsserieprognoser; vattenståndsprognoser; tidsserietransformatorer; återkopplade neurala nätverk

Acknowledgements

I would like to first thank my academic supervisor Nancy Xu, who helped me revise my thesis. Next, I would like to thank Stefano Bagli, the CEO of GecoSistema and also a supervisor, who gave me some pointers and quickly got me familiar with the state-of-the-art approaches. Both supervisors also gave me periodic feedback throughout the project and it was a pleasant experience working with both.

My examiner, Henrik Boström, also deserves great gratitude: for being involved during the thesis and answering my questions during feedback sessions and over email.

I would like to thank my peers at the University of Twente: Adjorn van Engelenhoven, Anna Mae van de Peut, and Íñigo Artolozaga for giving me feedback throughout making the thesis and giving me great input on my methods.

Lastly, I would like to thank the universities KTH, Royal Institute of Technology in Stockholm, and the University of Twente in Enschede. I have learned a lot through my master at these two universities, which enabled me to create this thesis, which I believe to be of contribution to computer science.

Acronyms

FLEX	Force Linear to Exponential
FCN	Fully Convolutional Network
gMLP	Gated MLP
GTT	GatedTabTransformer
GRU	Gated Recurrent Unit
IT	InceptionTime
LSTM	Long-Short-Term Memory
MAE	Mean Absolute Error
ML	Machine Learning
MSE	Mean Squared Error
NSE	Nash-Sutcliffe Coefficient of Efficiency
RNN	Recurrent Neural Network
SOTA	State-Of-The-Art
STF	SpaceTimeFormer
TCN	Temporal Convolutional Network
TFT	Temporal Fusion Transformer
TSF	Time Series Forecasting
TSM	Two-stage model
TST	Time-Series-Transformer
XCM	Explainable Convolutional Model

Contents

Chapter 1

Introduction

Time Series Forecasting (TSF) is a field of research within Machine Learning (ML). For problems within this field, ML models are expected to predict future data points from present and (usually) past data points. This is also more formally known as TSF. This form of ML has a fundamental issue: the engineer needs to balance accuracy and the forecast length. This issue is discussed in more detail in Section 1.2. Solutions have been proposed that aim to tackle this fundamental issue. However, there are still many options to consider. This thesis proposes an ensemble model that works together with existing TSF models, and is able to improve its effectiveness of forecasting time series. Using the aformentioned ensemble model results in both better accuracy and forecast length. The ensemble model will be specifically compared to models in the field of water level prediction (univariate and multivariate). For more information about this, refer to Section 1.5.

In this thesis, water level forecasting will be the main area of focus.[1] In this field, algorithms and models are proposed that attempt to forecast the level of water over time in water bodies. This thesis proposes an approach to create model ensembles that can perform as well as State-Of-The-Art (SOTA) models, while at the same time having fewer trainable parameters. Fewer parameters would have a few upsides. For example, the model would be smaller in memory. In addition, if two models are similar, the one with fewer parameters will usually train faster and have shorter inference times.

[1] Water level forecasting is also called *hydrological level forecasting* or even just *hydrological forecasting*. In this thesis, the term *water level forecasting* is used.

1.1 Background

Time series forecasting is a well-established field of research and has many good resources on existing algorithms and approaches [9, 14]. However, as ML models are becoming the norm in the field, new approaches are attempted and proposed continually [37].

The time frame in which a TSF model aims to predict future data is called the horizon. The horizon of a TSF model can be short-term or long-term, the choice for which is usually defined by the problem and field. However, TSF models have a fundamental issue: the results are more accurate if the model has to predict less data points [49, 51]. This means a model forecasting with a smaller horizon will usually perform better than the same model predicting a larger horizon. The reason for this is the following. If a model has the same size as another model, but has to forecast fewer points in the horizon, it will in general forecast those points better than the other model. A question could be rasied here: can this knowledge gain be used by a very small network to improve forecasts in general? In addition to this, if the forecaster is a recurrent neural network, having a larger horizon means the model will have to take more recurrent steps, which quickly results in vanishing gradients [27]. It has been shown that horizon size has an impact on the accuracy of forecasting models [49]. There have been proposed TSF models that forecast multiple different horizons at once [4]. These models have shown better results on smaller horizons, and therefore the authors proposed forecasting a few small horizons, instead of one large horizon. To add to this, there have been many proposed time series forecasting models that are trained on different 'resolutions' of the input data [5, 20, 53]. This enabled the models to more accurately forecast future trends at different scales, and made them both accurate with smaller (high resolution) horizons, and with larger (low resolution) horizons. The approach that is proposed in this thesis is highly inspired by these results: if horizons of lower resolution are easier to forecast, could this be used to make TSF models scale? Additionally, predicting fewer data points per time steps[2] also consistently yields better results [51]. This also means a model can forecast far into the future (as long as enough - historical - input data is provided) more accurately than another model that would forecast a much shorter horizon but with a higher data point density.

[2] Ergo, fewer total data points per forecast horizon will make TSF models perform better in general, as there are less points of prediction needed

In Figure 1.1.1 below, you can see different types of ways a horizon can be defined. From the information given above, we can transform a horizon in a few ways to decrease the number of data points (read: increase average model performance). Firstly, we can simply sample down the horizon, and forecast less. This is also called clipping. However, this comes at the cost of forecasting less far into the future. The second option is to reduce the resolution of the forecast horizon. This allows the model to forecast the same amount of time into the future as the base model. However, it comes at the cost of having more time between forecasted points. The third thing that could be done is something that has not been well tested in literature, which is to forecast at irregular intervals into the future. In this case, an example would be to forecast further into the future, depending on the data point index within the horizon. The latter approach is the fundamental idea behind the Force Linear to Exponential (FLEX) approach.

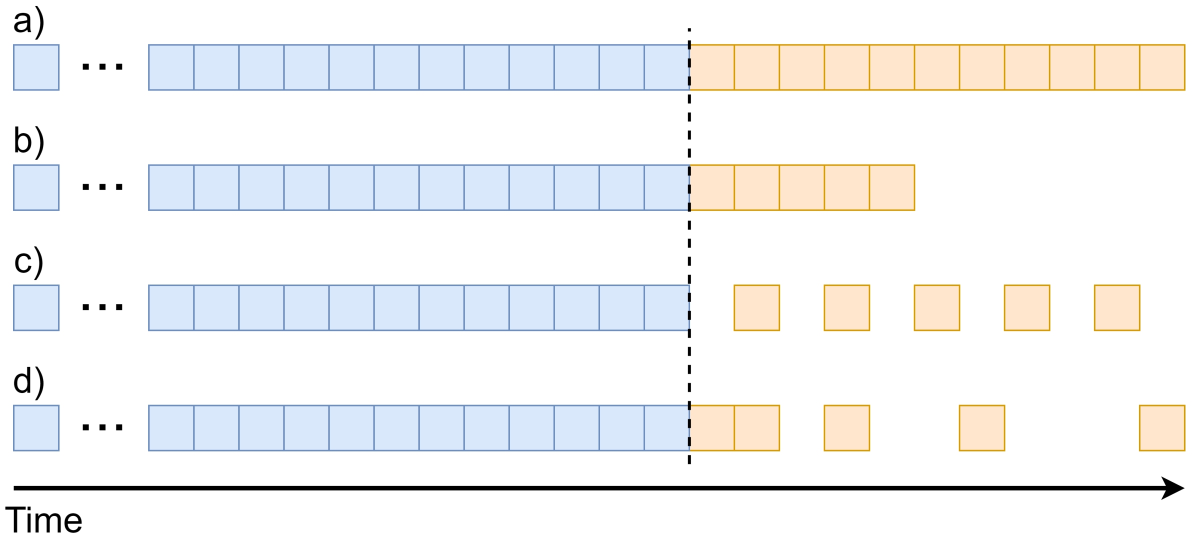

Figure 1.1.1: Different ways a horizon can be defined for a TSF model. Blue boxes are data points in history, which is the TSF model's input. Orange boxes are forecasted data points: the horizon. This figure includes a) A normal horizon of some given size; b) a downscaled horizon; c) a downsampled horizon; d) the proposed method.

Note that the proposed method above is only forecasting exponentially indexed data points on the horizon. There needs to be another (separate) model which takes these forecasted points and constructs the data points in between, to provide a full forecast. Such a model (first forecasting a part of the horizon and then constructing remaining points) is called a Two-stage model (TSM).

There are many SOTA models within TSF. Many models are optimised (and therefoer:

SOTA) in a specific area of TSF. In this thesis, when referring to ”SOTA TSF” models, what is meant is the list of the following models:

- Long-Short Term Memory Models (LSTMs)
- Gated Recurrent Unit Models (GRUs)
- Temporal Convolutional Networks (TCNs)
- Time Series Transformers (TSTs)
- InceptionTime models (ITs)
- Explainable Convolutional Networks (XCMs)

These models are explained in Chapter 2.

1.2 Problem

As stated in Section 1.1, it is difficult to make existing models generalize to perform well in both high-density, near future and low-density, far-future scenarios. This is mentioned in ”Forecasting: Principles & practice”. [43] In some TSF scenarios, this is important for a model to forecast. For example, with water level prediction [33], energy prediction [1] and financial prediction (stock price, sales, etc.) [44]. In these areas, it is important to forecast at high accuracy and high resolution in the near future, as well as forecast with good accuracy with low resolution in the far future.

An option that would come to mind is to train a large ensemble of models, which are each optimised for a different horizon (specifically: a horizon with an altered resolution). However, such an ensemble would quickly become very large for far-future horizons. In addition to this, such an ensemble would still have trouble accurately describing at a high resolution for far-future data points.

In short, the literature outlines the following two main problems:

- For long-term horizons, models need to either become very large, or need to forecast at a lower resolution.
- Short-term horizons are predicted well by SOTA models, but obviously often do not provide enough forecasting points to give valuable insights.

1.3 Purpose

The main purpose of this thesis is to answer the following research question:

> **Research question**
>
> Can a two-stage model - which first forecasts exponentially indexed data points; and secondly predicts the points in between the forecasted data points - perform as well as state-of-the-art time series forecasting models? The state-of-the-art time series forecasting models are: Long-Short Term Memory Models; Gated Recurrent Unit Models; Temporal Convolutional Networks; Time Series Transformers; InceptionTime models; and Explainable Convolutional Networks.

An additional purpose of this thesis is to outline a proposed TSF model and discuss its value in applications in the real world. The proposed model's results are compared to the SOTA within the field and similar models on the same problem.

Lastly, this thesis discusses future improvements on the model, and whether the model looks to be a promising advancement for TSF.

1.4 Objectives

To answer the research question given in Section 1.3, a list of objectives needs to be defined. This thesis has the following:

- First, define a general TSF ensemble, with a new model/ML structure.
- Second, train the defined ensemble and a list of existing SOTA TSF models on the same dataset.
- Following this, test all trained models and provide the results.
- Next, publish the code of the ensemble (make code open-source), so future work can build upon it.
- Lastly, provide a good base to build future work.

1.4.1 Benefits, Ethics and Sustainability

For every degree project, there are possible upsides and downsides. These can be during the execution of the project, but can also arise much later, as developed technology has impact on real-world systems.

This degree project will contribute towards the TSF research field. Therefore, this research field will benefit from the project. However, there are examples of greatly beneficial ML technologies being used for mass surveillance and infringements on human rights at large scale [21]. Furthermore, any form of ML technology can be (or become) biased and can therefore even impair people with activities in their daily lives [10].

The technological advancements of TSF can be used to forecast any series of data that evolves over time. Every use case in this large field will have ethical implications and issues. For example:

- If a bank were to be able to accurately forecast a person's spending for the next ten years, would it be ethical to use this to influence their decision on handing said person a mortgage?
- If forecasting the stock market would be possible to an extreme accuracy, but doing so required billions of dollars worth in equipment, couldn't this result in an extreme wealth gap?
- If TSF would be used to forecast flows of people through an area, wouldn't it be unforgivable when this technology is used by some external entity that wants to maximize loss of live in some sort of attack? Think of war crimes, but also terrorist attacks.

These are all issues that have to keep being examined and weighed against the possible benefits. However, in this thesis TSF is applied to water level forecasting, which aims to benefit flood prediction systems and aid countries with their disaster evacuations and economic repairs.

Floods are the most common natural disaster in the world. In addition, it has caused the most deaths of any natural disaster and (especially in developing countries) one of the most economically damaging natural disasters [24, 54]. When expressing the damage in terms of GDP of a country in which the disaster takes place, floods are by far the most economically damaging natural disaster in the world[3]. Most research into the impact of these natural disasters does not even include environmental and societal impact, which is also considered to be more considerable for floods in comparison to other natural disasters [2].

Predicting floods earlier and more accurately, paired with a structural management approach can significantly contribute to reducing economic damages and loss of life [30]. This thesis aims to contribute to this. Since flooding is such a large issue in the world, this likely will weigh up against the possible downsides of improving the technology.

1.5 Research methodology

During this project, a combination of a deductive and abductive approach will be used [25]. The proposed system will be compared to similar models using experiments and quantitative analyses of the results of those experiments. However, the model will also specifically be applied to the problem of water level forecasting, which is considered as a small case study during this project.

To answer the research question, the following steps will be taken:

- First, data is gathered from water basins in Italy. This data is cleaned and prepared for use with the proposed ML model.
- The proposed ML model is implemented and trained using the gathered and prepared data set.
- Existing models are trained using the gather and prepared data set.
- All trained models are evaluated on the same testing set. This is a sub-set from the prepared data set.

1.6 Delimitations

There are factors that could not be investigated in the time frame of this thesis. To describe the proposed approach within all contexts including all factors would take

[3] For clarification, floods cause approximately as much damage as climatological, geophysical and meteorological disasters. However, floods happen more frequently in developing countries and therefore destroy a larger percentage of economies.

a large number of experiments, which would not be properly ran and finished in time.

This thesis focuses on time series forecasting in the field of water level forecasting. This can affect results, as time series forecasting models can have significantly different results on different datasets [3, 56]. To be more specific: although the model proposed in this thesis might be applicable in many different fields of TSF, application to no field other than water level forecasting will be discussed.

In addition to this, the water level data and precipitation data that is used for training is gathered from an open portal for Italian water body data. As water level is influenced by many different factors, which can even be different depending on the location, this might also have a significant impact on the model's generalization capabilities [17, 29, 41, 57]. This thesis will not discuss applicability or accuracy outside this geographic region.

Lastly, only a subset of temporal models will be investigated. Models of many different types are examined. However, there are fundamental models that are not examined. The models that are examined are outlined in Section 3.2. Some models that are not included are gated temporal models such as GatedTabTransformers (GTTs) and Gated MLPs (gMLPs) [11, 38]. Besides this, some notable (newer) temporal transformer-based models like SpaceTimeFormers (STFs) and Google AI's relatively new Temporal Fusion Transformers (TFTs) [22, 36]. Lastly, Fully Convolutional Networks (FCNs) were also not investigated, as there was no clear indication from literature that they would perform well within a FLEX system [32].

1.7 Outline

Chapter 2 will discuss the related literature in more detail (in contrast on Section 1.1. This section will also mention the specific related works that this thesis's proposed system will be compared to.

After this, Chapter 3 will discuss the theory behind the used methodologies. These choice of methodologies will be explained. Lastly, the setup of the experiments will be explained. In addition, this chapter will discuss the work that was performed during the degree project. More specifically, the chapter will discuss the gathering of data; the processing of this data; how the proposed model was set up and how experiments were run.

Chapter 4 outlines the proposed model and its implementation.

Following this, Chapter 5 will display the results of the mentioned experiments. After this, the chapter discusses and analyses the results and prepares for the conclusions that can be made from the results.

Lastly, Chapter 6 conclusions drawn from the aforementioned discussion chapter. In addition to this, possible directions for future work are discussed.

Chapter 2

Theoretical background

Neural network ML was originally done using feed-forward neural networks. However, these networks did not perform very well on temporal data, as they lacked temporal context. To solve this, many different types of ML models were introduced. This chapter outlines some of the models that were introduced, and explains how they help shape the solution proposed in this thesis.

2.1 Recurrent Neural Networks

Recurrent Neural Networks (RNNs) were the first real temporal ML model that was introduced which performed significantly better on temporal data than other model types. These models would perform operations in order and use previous output of itself to calculate future outputs [31, 45].

2.2 Long-Short Term Memory Models

Long-Short-Term Memory (LSTM) models are a very basic type of RNN [28]. They have seen very wide-spread use and also are SOTA in water level forecasting [34]. Gated Recurrent Units (GRUs) aim to improve on LSTMs [12]. GRUs have been shown to sometimes perform better on data sets with fewer data points [13, 23].

2.3 Temporal Convolutional Networks

There are Temporal Convolutional Network (TCN) models, which use exponentially dilated convolutional layers to train faster on temporal data. Certain operations for later time steps in the input data do not need to be done after earlier time steps have been completed (like with RNNs), so training is usually much faster (as training can be parallelized better) [6].

2.4 Time Series Transformers

Time-Series-Transformer (TST) models are attention-based temporal networks [58]. TSTs are inspired by Google Deepmind's transformers, which popularized fully attention-based models [50]. TSTs are really helpful within this thesis, as they are not recurrent, and therefore can have temporal outputs and handle temporal inputs, but also make use of extra input data. They are a key ingredient in the FLEX approach.

2.5 InceptionTime

An InceptionTime (IT) model is a model that uses inception blocks [19]. These blocks are recurrent blocks that attempt to output the entire output after every block. Because of residual connections between blocks, and because every block is trying to output the exact same, IT models usually learn very fast.

2.6 Explainable Convolutional Networks

Explainable Convolutional Models (XCMs) are simply convolutional networks (N dimensional) that have smaller sub-networks which make it more explainable [18]. It has not been SOTA in many areas, but has seen wide use due to the explainability feature.

2.7 One-Cycle training policy

In this thesis, Smith's 1-cycle (alternatively: one-cycle) training policy is used as the main training approach. Because of this, it is important to know how this policy works [46].

The 1-cycle policy that is used in this thesis specifically has the following three steps:[1]

- We progressively increase our learning rate from `lr_max` / `div_factor` to `lr_max` and at the same time we progressively decrease our momentum from `mom_max` to `mom_min`.
- We do the exact opposite: we progressively decrease our learning rate from `lr_max` to `lr_max` / `div_factor` and at the same time we progressively increase our momentum from `mom_min` to `mom_max`.
- We further decrease our learning rate from `lr_max` / `div_factor` to `lr_max` / (`div_factor` × `100`) and we keep momentum steady at `mom_max`.

In the steps above, `lr_max`, `div_factor`, `mom_min`, and `mom_max` are parameters given when training. The exact values for these parameters that were used in this paper are outlined in Section 3.2.3.

Using the 1-cycle policy, models can be trained much faster. This improvement in convergence time is so large, that the 1-cycle policy training can be considered ”super-convergence” [46, 47].

2.8 Statistical testing

From two sets of metrics (one set from two different models, for example), it is possible to calculate whether the two sets are statistically significantly different. Usually, this is done by providing some value for the parameter p, which states how certain one can be that two series are significantly different. A value of $p < 0.05$ is considered as the maximum value that signifies statistical significance. Usually, statistical tests are done with a given value for p. However, it is also possible to calculate p from two series [39]

[1] These steps are taken from: `https://fastai1.fast.ai/callbacks.one_cycle.html`

using a t-test.[2] In this case, we specifically use a *paired* t-test.

To calculate p, the first step is to calculate the pooled standard deviation σ:

$$\sigma = \sqrt{\frac{(n_1 - 1)\sigma_1^2 + (n_2 - 1)\sigma_2^2}{n_1 + n_2 - 2}} \tag{2.1}$$

where σ_1 and σ_2 are the standard deviations of the two samples with sample sizes n_1 and n_2.

After this, the standard error se is calculated:

$$se(\overline{x_1} - \overline{x_2}) = \sigma\sqrt{\frac{1}{n_1} + \frac{1}{n_2}} \tag{2.2}$$

Where $\overline{x_i}$ is the mean of the data series x_i, and n_i is the sample size of the data series x_i. The p-value can then be calculated with a t-test, where the distribution for t is as follows:

$$t = \frac{\overline{x_1} - \overline{x_2}}{se(\overline{x_1} - \overline{x_2})} \tag{2.3}$$

From this distribution, the p-value is the area of the t distribution with $n_1 + n_2 - 2$ degrees of freedom, that falls outside $\pm t$.

2.9 Evaluation metrics

The main evaluation metric that will be used is the Mean Absolute Error (MAE). This is defined as:

$$M_{\text{MAE}} = \frac{1}{N}\sum_{i=1}^{N} |y_i - \hat{y}_i| \tag{2.4}$$

In Equation 2.4, y_i is any given actual output, and $\hat{y}_i$ is the predicted output for the same input. In the equation above, and the equations below, N is the number of predictions

[2] Please note that performing a simple t-test such as described in this section may lead to a type-I error as described by Bengio et. al. (2003) [7]. This means that the resulting p-value can be a strong indicator of statistical significance, but should not be considered fully truthy without more tests being done. The exact assumptions that are made for this thesis will be discussed in Section 3.2.5.

done.

However, for the analysis of TSF models, it is interesting to see the error over the horizon: depending on how far into the future the model is predicting, what is the MAE for all predictions x time steps into the future? In this thesis, a metric will be used that will be called MAE-Over-Horizon. This is the MAE over horizon as the model is predicting points further into the future and is defined as follows:

$$M_{\text{MAEOH}} = \{M_{\text{MAEOH},h} | h \in H\} \tag{2.5}$$

In Equation 2.5, H is the list of all horizon indices. For this set, $H \subset \mathbb{Z}$. $M_{\text{MAEOH},h}$ is the MAE of all predictions h time steps into the horizon. It will be defined as:

$$M_{\text{MAEOH},h} = \frac{1}{N} \sum_{i=1}^{N} \left| Y_{h,i} - \hat{Y}_{h,i} \right| \tag{2.6}$$

In Equation 2.6, $Y_{h,i}$ is the actual output at h time steps in the horizon for some input, and $\hat{Y}_{h,i}$ is the predicted output h time steps in the horizon for the same input.

Lastly, the Nash-Sutcliffe Coefficient of Efficiency (NSE) was used as a metric to quantitatively evaluate models. The computation of the metric is similar to the Mean Squared Error (MSE) metric. However, it is normalized depending on how the entropy in the dataset. The NSE is computed as follows:

$$NSE = 1 - \frac{\sum_{i=1}^{N} \left(y_i - \hat{y}_i\right)^2}{\sum_{i=1}^{N} \left(\hat{y}_i - \overline{\hat{y}_i}\right)^2} \tag{2.7}$$

Chapter 3

Methods

The methodologies presented in this thesis are based upon the portal of research methods proposed by Håkansson in 2013 [25].

3.1 Choice of research method

In the choice of methodology for this thesis, replicability is important, as any reader should be able to validate the results and claims presented in this thesis. The method of Design Science will be used. Design Science is a methodology spanning many specific methodologies [52]. However, the fundamental objective is to create abstract knowledge and systems which can be used by professionals in industries to create solutions. As this thesis is done in collaboration with a geological data-driven consultancy, such a methodology is highly valuable, as the knowledge is validated and presented in a way it can be easily applied in a solution.

The proposed model, FLEX, is compared against other models. These baseline models will be implemented during this thesis in the same software library. All models are then in a set of experiments. Metrics are gathered from each model and based on these metrics, the results will be compiled and analysed. The baseline models that will be analyzed are listed in Section 1.1. However, to reiterate, the baseline models that will be implemented are: LSTM, GRU, TCN, TST, IT, XCM.

This thesis will have a short qualitative analysis of the results, as this is also highly valuable. Because this thesis proposes a new model, it is important to understand what outputs the model can generate: it becomes important to create an understanding of the newly proposed model. This is why this is also done in addition to the quantitative method, which investigates how the model performs overall and on average.

As stated before, replicability and credibility of the results of the proposed model is very important. Therefore, focus was relieved from performing a case study of the model within a field. Such a case study was performed: the model was trained and tested within a specific region of Italy. However, the case study does not significantly add upon the knowledge gained within the thesis, and therefore was not used within the analysis of the results. Of course, the fact that the model was applied in a case study does have an influence on the delimitation of this work.

3.2 Application of research method

3.2.1 Area of study

For this thesis, a specific area of research within the field of TSF was chosen. The area was chosen to have highly available data; a have lot of existing SOTA research; and to be in line with the mission fo the company this thesis was done at. The chosen area is water level forecasting.

Water level forecasting is a type of time series forecasting which aims to accurately predict water level in any water body. The SOTA model for this is heavily dependent for the type of water body in which the water level is forecasted. For example, coastal prediction and lake prediction have two very different models for the SOTA, even though both are water bodies [35, 55]. The current SOTA approach for water level forecasting in rivers is Google Deepmind's Hydronets [40]. These models first use ML to first describe a state of the water body, and then use fluid simulation to forecast water level and potential floods.

3.2.2 Data gathering and processing

First of, the data was gathered from the Dext3r portal from Arpae [15]. The data types that were requested were water level and cumulative precipitation over an hour. This means that every data point in the dataset has a timestamp, the water level at that timestamp and the total amount of rainfall in the hour preceding the timestamp. The resolution of data points was one point per hour, and the total dataset was just over seven years of data, which brings the total amount of data points to just over 61,300 points. These points were gathered from two locations: Parma River, and Casalecchio Canal in Italy. These were chosen because existing research also often looks at these areas. This was split into training, testing and validation sets. The training set took up 60% of the data, the validation set took up 20% and the testing set took up the remaining 20%. K-fold cross validation was used for training and validation (with $k = 4$). The testing set was kept constant. The data splits for different folds is shown in Figure 3.2.1, below.

For every training entry in the dataset calculating the MAE and NSE, a history size of 672 was taken, which is exactly 4 weeks of data every hour. A horizon size of 168 was taken, which is exactly one week of data every hour.

For investigating the individual horizons, a history size of 336 (two weeks, every hour) was taken, and a horizon size of 96 (4 days, every hour) was taken.

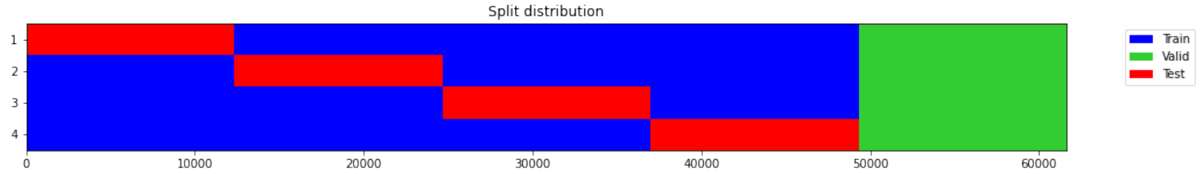

Figure 3.2.1: Splits of training / validation / test sets for different folds. Every row is a different fold. Colors show what indices of the data is dedicated to what (training, validation or testing). Testing data is always the same data, while training and validation data for training is shuffled.

3.2.3 Model training

After pre-processing the data, and defining the ML model, the model has to be trained. Firstly, a pre-training method will be used where the model will first be trained with a loss function with a exponentially indexed mask. Literature suggest that pre-training can end up not adding any significant improvements to results. Therefore, a different training approach will be employed. The model will be trained in one pass, but the gradients are stopped on the connections from the base model to the reconstruction model. This means specific indexes of the loss vector for every prediction will be used to train the base model, and the remaining indexes will be used to train the reconstruction model. Because of the gradient cut, the base model is not adjusted with the error on remaining indexes. This also means that the base model learns solely the exponentially indexed points, and not some hidden representation of the forecast.

The parameters that are used for the one-cycle training (explained in Section 2.7) are: `div_factor` $= 25$, `mom_min` $= 0.85$, and `mom_max` $= 0.95$. The value for `lr_max` is inferred automatically by using a learning rate finding algorithm first described by Leslie Smith [48].

3.2.4 Experiment setup

As mentioned in Section 1.5, there will be a set of experiments set up to verify the proposed system. These experiments are set up for the assurance of replicability. In addition to this, some results from the experiments will be inspected qualitatively. These inspections and discussion thereof are to discover the validity of the quantitative results and to provide insight into how the proposed system would influence performance of models in the real world.

Experiment Setup

Test the proposed model with different arch-types and compare it with standalone models of that arch-type. The comparisons are done based on MAE and NSE. The two models should have a similar size (parameter count), or the standalone model should be much larger. The arch-types that will be considered are:

- A TCN model [6].
- A TST model [58].
- An IT model [19].
- An XCM [18].
- A LSTM model [28].
- A GRU model [12].

These models will be evaluated quantitatively on a water level dataset. Their MAE will be calculated. In addition to this, the average MAE is also averaged over many runs. The NSE is calculated and averaged the same way. The results will be discussed qualitatively by also providing average error over horizon per forecast. The loss will evolve in different ways over the horizon size. The loss will increase for all models, but this increase might be at different speeds and accelerations. Then, this will be verified quantitatively on all forecasts (average derivative of loss function). The exact way the MAE is calculated for average forecasts, is explained in Section 2.9.

3.2.5 Statistical analysis assumptions

A t-test (as described in Section 2.8) makes a few assumptions. Some assumptions, in the case of this thesis, are more trustworthy than others. Below, a list with the assumptions (bolded) and a short discussion on why this assumption was made and its credibility [8].

1. **The scale of measurement applied to the data collected follows a continuous or ordinal scale, such as the scores for an IQ test**: This is the case for this thesis.

2. **The data is collected from a representative, randomly selected portion of the total population**: This is not the case for this thesis. On the one hand, the data that is collected are trained models with a random initialization of parameters. However, there is a problem that makes this assumption untrustworthy: a lot of measurements will follow from cross-validation, which means that the results are not unbiased or independent. This means the outcomes of the results are only indicators and the statistical test should be treated as a heuristic in this thesis.

3. **The data, when plotted, results in a normal distribution**: This is not necessarily the case for all metrics, but existing work has shown that machine learning models' evaluation does follow a normal distribution if the metric is positively correlated with its accuracy: for example the MAE used for this thesis [16, 26].

4. **A reasonably large sample size is used. A larger sample size means the distribution of results should approach a normal bell-shaped curve**: This assumption could easily be refuted. The sample size is only 20, which also follow from k-fold cross validation. Therefore, a larger sample should be considered to give more convincing and credible results.

5. **Homogeneous, or equal, variance exists when the standard deviations of samples are approximately equal**: This assumption is only needed for *unpaired* t-tests. However, as stated in Section 2.8, this thesis makes use of a *paired* t-test. Therefore, the two distributions' variances do not need to be the same (which likely also is not the case).

Chapter 4

Proposed model

The ensemble model that is proposed in this thesis is one of the outcomes of the thesis. This chapter explains how this model works.

4.1 General architecture

The model was created using the PyTorch library [42]. The model has two general components:

- **The base model** is the base architecture that outputs the initial forecast. This forecast, whereas it would usually be linear, is an exponential forecast. This is what the name ”FLEX” thanks its name to.
- **The reconstruction model** is a light-weight model which is responsible for outputting the final forecast. It takes the initial input data, and the exponential forecast of the base model.

4.2 Model rationale

The exponential forecast points assist the reconstruction model to have relatively low-loss forecast points to base the final forecast on. This of course will only work if the base model trying to forecast a very small forecast has a better loss than the light-weight reconstruction model. This is not always the case, in which case the forecasting error at the exponential indexes will jump up relative to the overall loss curve over an average forecast. These cases are shown in Chapter 5 and discussed in Chapter 6.

The idea behind the architecture is that the two individual modules in the overall model are solving simpler problems. This allows them to be much smaller. So small in fact, that the sum of parameter counts of the two models can be much smaller than the architecture it is comparing against. In addition to this, even though it is smaller and usually reaches a similar performance, it also can reach an even better performance even though the model it is comparing against (of the same base architecture) is twice as large. These results are shown and discussed in Chapter 5.

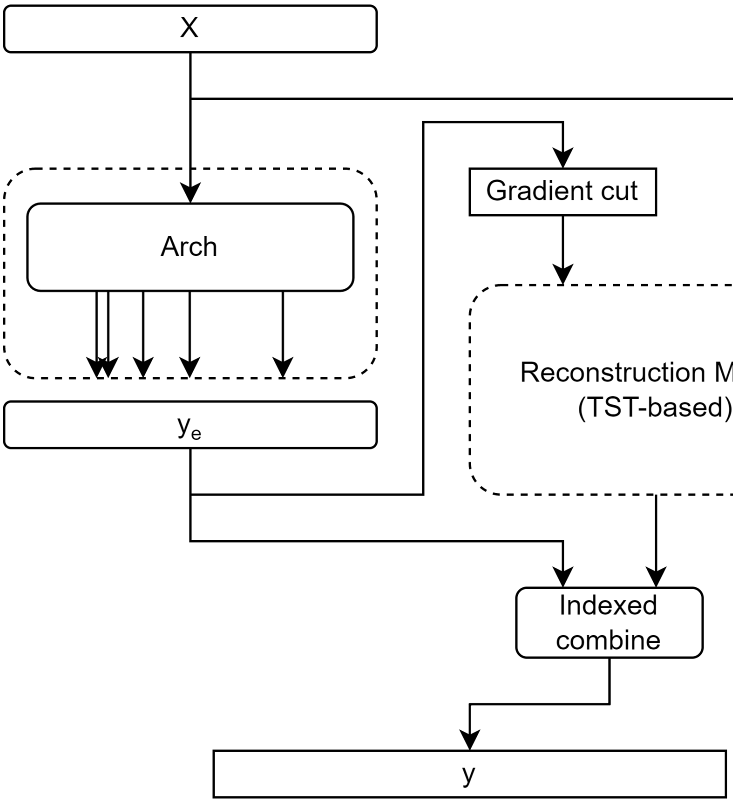

Figure 4.2.1: Architecture of the proposed model. The series of lines from the base architecture (”Arch” in the figure) are the exponentially indexed outputs. These outputs are fed to the reconstruction model after a gradient cut.

4.3 Indexing definitions

The indexes of the forecast points of the base model are determined by a hyper-parameter called the *base exponent* (referred to as χ in this thesis). The relationship between the indexes to the base exponent are defined in Equation 4.1:

$$I = \left\{ i | i = \chi^k - 1 \quad \forall\, k \in \left[\!\left[0..\left\lceil \log_\chi \left(|H| \right) \right\rceil \right]\!\right] \right\} \tag{4.1}$$

In Equation 4.1, I is the set of all indexes that the base model will forecast and pass to the reconstruction model, for some base exponent $\chi > 1$. H is the horizon that is being predicted by the model, and therefore $|H|$ is the number of time steps the model is forecasting into the future. The formula above has no conditions, but is mathematically actually equivalent to the (arguably simpler) formula below:

$$I = \left\{ i | i = \chi^k - 1, \text{ if } \chi^k < |H| - 1 \quad \forall \, k \in \left[\!\left[0..\, |H| \right]\!\right] \right\} \tag{4.2}$$

Chapter 5

Experimental results

5.1 Errors over horizon

First, the M_{MAEOH} was calculated for all models mentioned in Section 3.2. The results are shown in the figures below. The figure always contains two graphs: an orange and a blue graph. The orange graph shows the base architecture and the blue line shows the FLEX approach using that same base architecture for the base model (explained in Chapter 4).

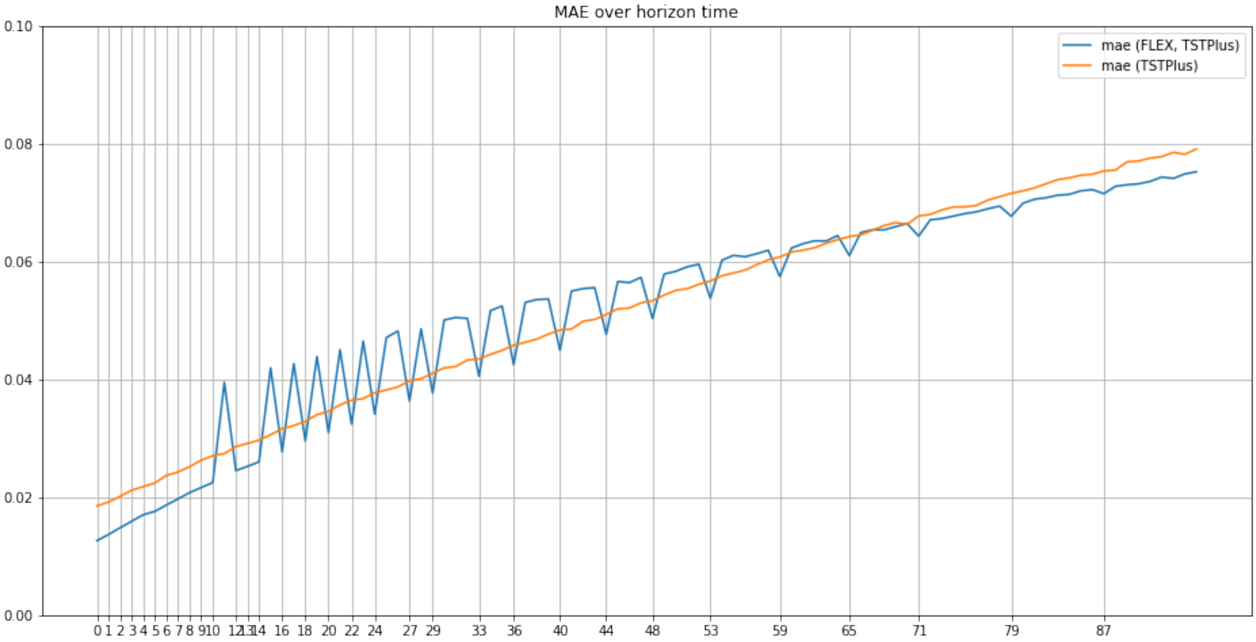

Figure 5.1.1: Results over horizon time of TST arch-type

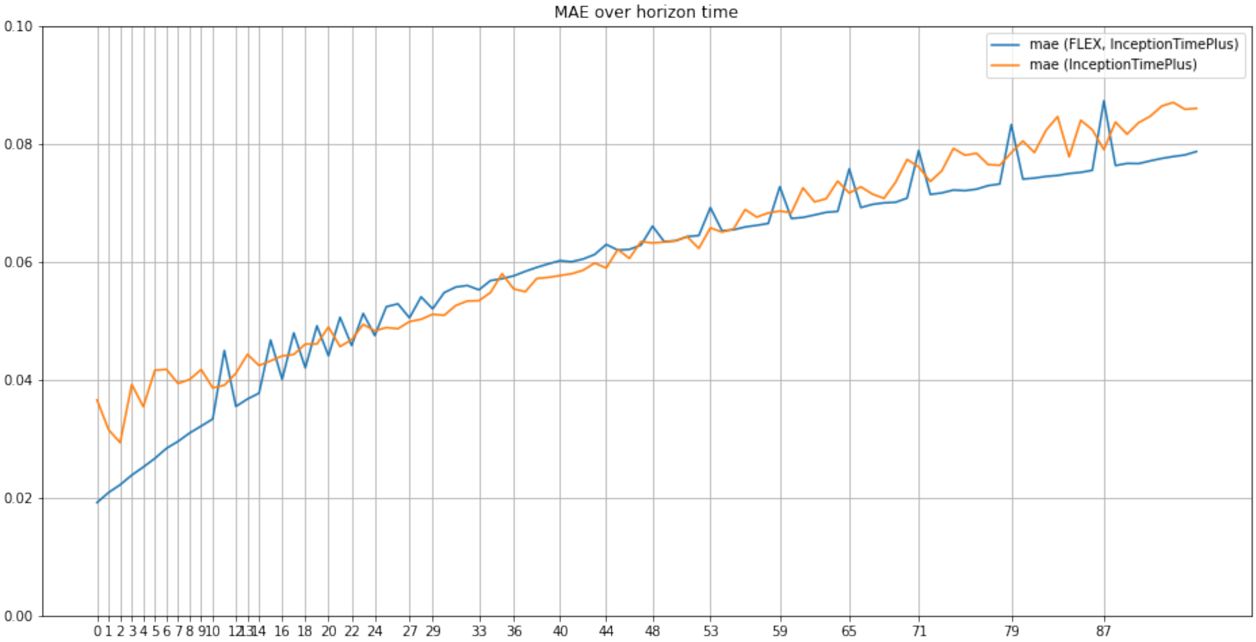

Figure 5.1.2: Results over horizon time of IT arch-type

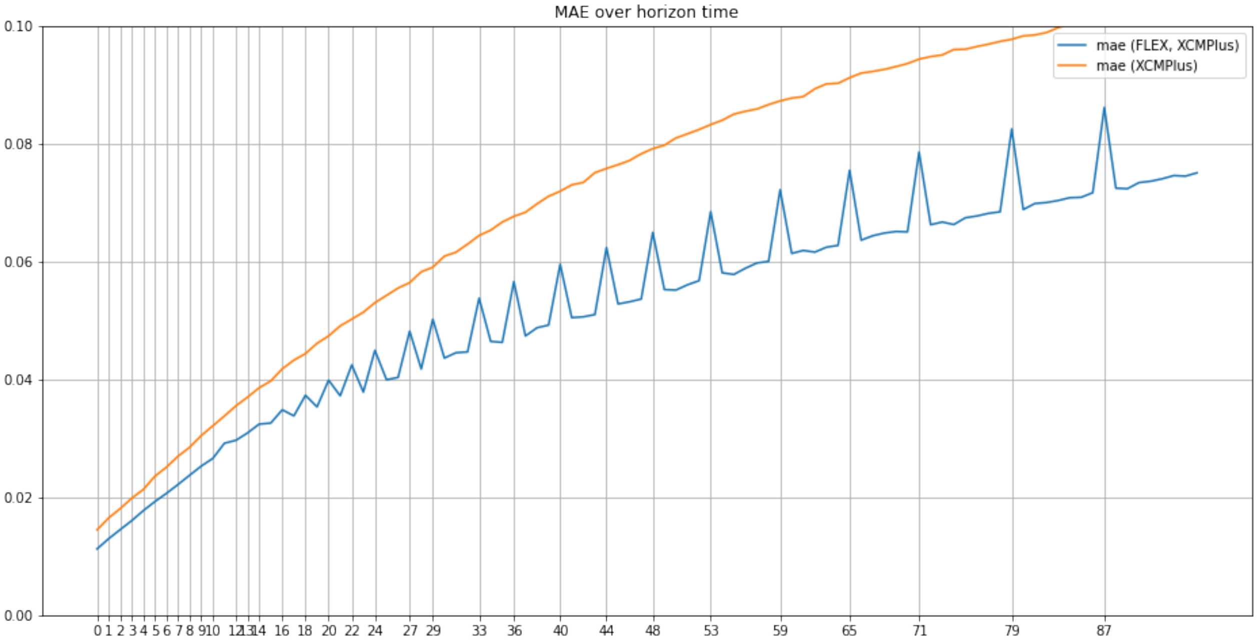

Figure 5.1.3: Results over horizon time of XCM arch-type

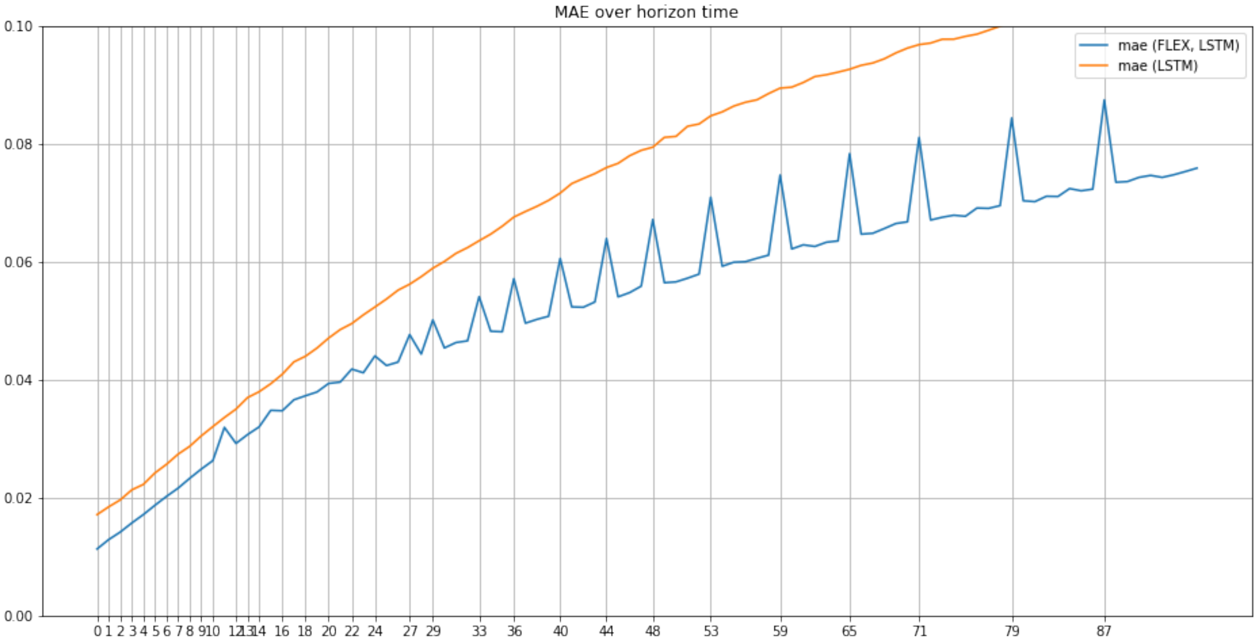

Figure 5.1.4: Results over horizon time of LSTM arch-type

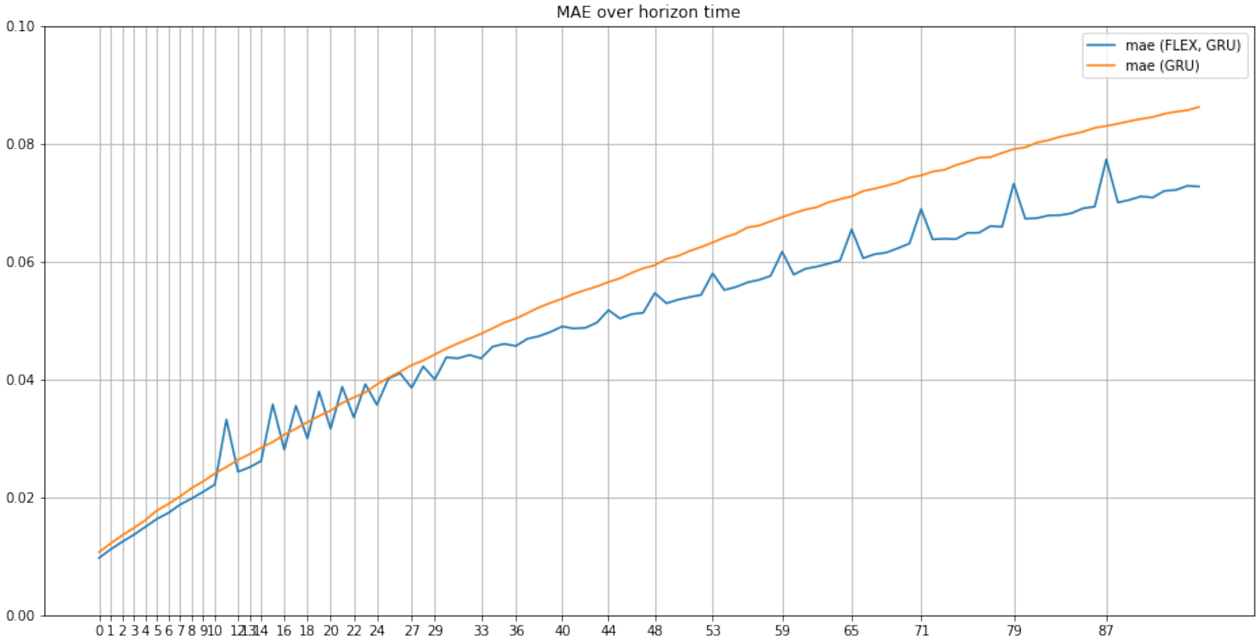

Figure 5.1.5: Results over horizon time of GRU arch-type

5.2 Compiled results

After this, a table was computed with aggregate data over all runs and all data points. The results can be seen in Table 5.2.1, below.

Model name	Parameters	$\mu \pm \sigma$	Minimum
TCN	235K	0.0574 ± 0.0011	0.0551
FLEX (TCN, $\chi = 1.1$)	119K	0.0531 ± 0.0014	0.0520
TST[2]	197K	0.0534 ± 0.0017	0.0506
TST	98K	0.0527 ± 0.0005	0.0514
FLEX (TST, $\chi = 1.1$)[2]	99K	0.0521 ± 0.0009	0.0503
IT[3]	10.6M	0.0610 ± 0.0010	0.0597
FLEX (IT, $\chi = 1.1$)[3]	6.7M	0.0570 ± 0.0032	0.0541
XCM[4]	6.6M	0.0561 ± 0.0014	0.0528
FLEX (XCM, $\chi = 1.1$)[4]	3.3M	0.0549 ± 0.0021	0.0514
LSTM[5]	4.9M	0.0736 ± 0.0046	0.0667
FLEX (LSTM, $\chi = 1.1$)[5]	2.5M	0.0540 ± 0.0007	0.0530
GRU[6]	4.9M	0.0634 ± 0.0066	0.0581
FLEX (GRU, $\chi = 1.1$)[6]	2.5M	**0.0514** ± 0.0012	**0.0495**

Table 5.2.1: Compiled results from all experiments. All values are aggregates over 20 model evaluations (5 runs × 4 folds). Showing (in order) model name, total parameter count, mean (μ), standard deviation of MAE across models and folds (σ), and minimum value. Best scores are bolded.

[2] Shown in figure 5.1.1.

[3] Shown in figure 5.1.2.

[4] Shown in figure 5.1.3.

[5] Shown in figure 5.1.4.

[6] Shown in figure 5.1.5.

As can be seen in the table above, it can be seen that the best performing model is FLEX (GRU, $\chi = 1.1$). In addition, please note that all base architectures are approximately twice the size (two times the parameter count) of the FLEX-based model with the same architecture. This has one exception: the TST architecture was tested twice with two parameter totals. This is because the reconstruction model (explained in Chapter 4) is a TST. Therefore, FLEX models outperforming the base model could have been explained because TSTs lend themselves well to this problem. Therefore, some extra

TST models were tested, to show the true impact the FLEX approach specifically has on the performance.

5.3 Statistical tests

The values for p were computed as explained in Section 2.8 and are as follows:

Model 1	Params	Model 2	Params	p
TCN	235K	FLEX (TCN, $\chi = 1.1$)	119K	< 0.0001
TST	197K	FLEX (TST, $\chi = 1.1$)	99K	0.0045
TST	98K	FLEX (TST, $\chi = 1.1$)	99K	0.0130
IT	10.6M	FLEX (IT, $\chi = 1.1$)	6.7M	< 0.0001
XCM	6.6M	FLEX (XCM, $\chi = 1.1$)	3.3M	0.0400
LSTM	4.9M	FLEX (LSTM, $\chi = 1.1$)	2.5M	< 0.0001
GRU	4.9M	FLEX (GRU, $\chi = 1.1$)	2.5M	< 0.0001
FLEX (TST, $\chi = 1.1$)	99K	FLEX (GRU, $\chi = 1.1$)	2.5M	0.0436
XCM	6.6M	FLEX (GRU, $\chi = 1.1$)	2.5M	< 0.0001

Table 5.3.1: Table showing the statistical significance of the difference in results of *Model 1* and *Model 2*. In this table, *Model 2* is the better model (read: lower mean in results), and the p-value is for the hypothesis test $H_0 : \overline{x_2} < \overline{x_1}$, where $\overline{x_i}$ is the mean of the results of *Model* i.

5.4 Analysis

As can be seen in Table 5.2.1, the GRU model performs the best on the water level forecasting problem. Something else that can be seen is that the FLEX models always outperform the base models of the same architecture. However, the FLEX models always have a submodel which is a TST. Therefore, the improved performance for most of the models in the table can be explained with the fact that a small TSTs might just be better at learning this problem than most of the other models.

To verify whether the FLEX approach actually improves the model, it needs to be compared to the TST architecture directly. In this case, the FLEX approach uses two smaller TSTs and is compared to one larger TST. Even in this case, FLEX is still slightly better. However, the performance of the models is within 1% of each other. This is why the statistical tests were done. Between the FLEX TST model and the best (98K parameters) baseline TST model, the p-value is 0.0130. This still suggests the models' results differences are statistically significant.

Lastly, it is uncertain whether these results will carry over to other fields of TSF, for example the field of finance, briefly introduced in section 1.2. For now these results remain proven only in the context of water level forecasting.

Chapter 6

Concluding remarks

From the analysis, some conclusions can be made. The conclusions are outlined in Section 6.1. Ideas and recommendations to better explore issues brought up in the discussion are found in Section 6.2.

6.1 Conclusions

The results and discussion reveal some conclusions that can be made. These conclusions are summed up below.

- The FLEX model consistently outperforms its base counterpart, whether the counterpart is the same size or even much larger, in both the metrics MAE and NSE. This means FLEX is able to improve TSF models by simply extending an existing model.

- Although improving results, FLEX is not always able to extend any model such that it performs better than any other model. The TSTs, for example, are only ever beaten by the FLEX model for TST and for GRU.

- The FLEX model using a base architecture of GRU is statistically significantly ($p = 0.0436$ at worst) the best model. However, the statistical significance is slim, assuming a threshold of $p = 0.05$.[1]

- An exponentially indexed model (as described in Section **??** performs better on

[1] Kindly note that (as described in Section 3.2.5, there is an incorrect assumption in this thesis about the bias in the sampling of the results. Therefore, this conclusion should be seen as likely, but not proven statistically.

its indexes than a model that is not indexed would on those indexes. This means TSF models can become better, and the limiting factor currently is horizon size.

To reiterative the research question: *” Can a two-stage model - which first forecasts exponentially indexed data points; and secondly predicts the points in between the forecasted data points - perform as well as state-of-the-art time series forecasting models? The state-of-the-art time series forecasting models are: Long-Short Term Memory Models; Gated Recurrent Unit Models; Temporal Convolutional Networks; Time Series Transformers; InceptionTime models; and Explainable Convolutional Networks. ”*. The answer to this question is: yes, most likely. First, the context in which it would outperform other SOTA models is important, and has not been fully described yet. However, in water level forecasting specifically, the research question can be easily be answered. In addition, many factors that influence the exact context in which this claim holds true have not all been investigated. Therefore, more research would have to be done into this approach to prove its value to the TSF field.

Existing literature has shown that shorter horizons lead to better model performance, even for large models. This thesis has shown that it is indeed possible to also forecast far into the future, using a horizon with steps that are exponentially far apart from each other. After this, a reconstruction model (for which TSTs are perfectly suited) can use this information to provide a better final forecast.

6.2 Future work

The results of the experiments are promising. However, there are many areas in which the FLEX approach has to be verified.

Firstly, the experiments should be run and the results verified on a dataset from a different field. The financial field (share price forecasting, sales forecasting, customer forecasting) would be ideal for this, as it makes up for a large part of the TSF field.

Secondly, the FLEX approach should be run for many different exponents. It is possible the optimal exponent is linked to the horizon size, in which case this correlation should be investigated and preferably described. As a continuation, it would be important to see if a hyperparameter-optimized FLEX model outperforms a hyperparameter-optimized base architecture model.

Lastly, all the above future work and given experiments in this thesis should be run without using k-fold cross validation to retrieve the results of the statistical tests (t-tests). This will result to a sound metric that can have FLEX be properly evaluated.

6.3 Final Words

In this thesis, a simple ensemble was introduced that had a novel approach to TSF. The model was able to outperform all considered baseline models. Therefore, the proposed approach proves to be useful in at least the given research area of water level forecasting. Although there remains some future work to be done on the exact effectiveness of the approach (in what areas it is useful, etc.), this could be a good step into a new family of TSF models.

Bibliography

[1] Afrasiabi, Mousa, Mohammadi, Mohammad, Rastegar, Mohammad, and Kargarian, Amin. “Multi-agent microgrid energy management based on deep learning forecaster”. In: *Elsevier, Energy* 186 (2019), p. 115873. ISSN: 0360-5442. DOI: `https://doi.org/10.1016/j.energy.2019.115873`. URL: `https://www.sciencedirect.com/science/article/pii/S0360544219315452`.

[2] Allaire, Maura. “Socio-economic impacts of flooding: A review of the empirical literature”. In: *Water Security* 3 (2018), pp. 18–26. ISSN: 2468-3124. DOI: `https://doi.org/10.1016/j.wasec.2018.09.002`. URL: `https://www.sciencedirect.com/science/article/pii/S2468312418300063`.

[3] Athiyarath, Srihari, Paul, Mousumi, and Krishnaswamy, Srivatsa. “A comparative study and analysis of time series forecasting techniques”. In: *SN Computer Science* 1.3 (2020), pp. 1–7.

[4] Azam, Furqan and Younis, Shahzad. “Multi-Horizon Electricity Load and Price Forecasting Using an Interpretable Multi-Head Self-Attention and EEMD-Based Framework”. In: *IEEE Access* PP (June 2021), pp. 1–1. DOI: `10.1109/ACCESS.2021.3086039`.

[5] Bahrpeyma, Fouad, Mccarren, Andrew, and Roantree, Mark. “Multi-Resolution Forecast Aggregation for Time Series in Agri Datasets”. In: Dec. 2017.

[6] Bai, Shaojie, Kolter, J. Zico, and Koltun, Vladlen. “An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling”. In: *CoRR* abs/1803.01271 (2018). arXiv: `1803.01271`. URL: `http://arxiv.org/abs/1803.01271`.

[7] Bengio, Yoshua and Grandvalet, Yves. “No unbiased estimator of the variance of k-fold cross-validation”. In: *Advances in Neural Information Processing Systems* 16 (2003).

[8] Cals, Jochen and Winkens, Bjorn. *The Student t-test is a beer test.* nl. Netherlands, Aug. 2018.

[9] Chatfield, Chris. *Time-series forecasting*. Chapman and Hall/CRC, 2000.

[10] Chokshi, Niraj. “Facial Recognition’s Many Controversies, From Stadium Surveillance to Racist Software’”. In: *The New York Times* (2019).

[11] Cholakov, Radostin and Kolev, Todor. “The GatedTabTransformer. An enhanced deep learning architecture for tabular modeling”. In: *CoRR* abs/2201.00199 (2022). arXiv: 2201.00199. URL: https://arxiv.org/abs/2201.00199.

[12] Chung, Junyoung, Gulcehre, Caglar, Cho, KyungHyun, and Bengio, Yoshua. *Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling*. 2014. DOI: 10.48550/ARXIV.1412.3555. URL: https://arxiv.org/abs/1412.3555.

[13] Chung, Junyoung, Gulcehre, Caglar, Cho, KyungHyun, and Bengio, Yoshua. *Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling*. 2014. DOI: 10.48550/ARXIV.1412.3555. URL: https://arxiv.org/abs/1412.3555.

[14] De Gooijer, Jan G and Hyndman, Rob J. “25 years of time series forecasting”. In: *International journal of forecasting* 22.3 (2006), pp. 443–473.

[15] *DEXT3R data portal by Arpae*. https://simc.arpae.it/dext3r/. Accessed: 2022-04-16.

[16] Dietterich, Thomas G. “Approximate statistical tests for comparing supervised classification learning algorithms”. In: *Neural computation* 10.7 (1998), pp. 1895–1923.

[17] Fan, Fernando Mainardi, Paiva, Rodrigo Cauduro Dias de, and Collischonn, Walter. “Hydrological forecasting practices in Brazil”. In: *Flood Forecasting*. Elsevier, 2016, pp. 41–66.

[18] Fauvel, Kevin, Lin, Tao, Masson, Véronique, Fromont, Élisa, and Termier, Alexandre. “XCM: An Explainable Convolutional Neural Network for Multivariate Time Series Classification”. In: *Mathematics* 9.23 (Dec. 2021), pp. 1–20. DOI: 10.3390/math9233137. URL: https://hal.inria.fr/hal-03469487.

[19] Fawaz, Hassan Ismail, Lucas, Benjamin, Forestier, Germain, Pelletier, Charlotte, Schmidt, Daniel F., Weber, Jonathan, Webb, Geoffrey I., Idoumghar, Lhassane, Muller, Pierre-Alain, and Petitjean, François. "InceptionTime: Finding AlexNet for Time Series Classification". In: *CoRR* abs/1909.04939 (2019). arXiv: `1909.04939`. URL: `http://arxiv.org/abs/1909.04939`.

[20] Ferreira, Marco A. R., Higdon, David M., Lee, Herbert K. H., and West, Mike. "Multi-scale and hidden resolution time series models". In: *Bayesian Analysis* 1.4 (2006), pp. 947–967. DOI: `10.1214/06-BA131`. URL: `https://doi.org/10.1214/06-BA131`.

[21] Fitzgerald, Grace. "Chinese Authoritarianism and the Systematic Suppression of the Uighur Ethnic Minority". In: (2020).

[22] Grigsby, Jake, Wang, Zhe, and Qi, Yanjun. "Long-Range Transformers for Dynamic Spatiotemporal Forecasting". In: *arXiv preprint arXiv:2109.12218* (2021).

[23] Gruber, Nicole and Jockisch, Alfred. "Are GRU cells more specific and LSTM cells more sensitive in motive classification of text?" In: *Frontiers in artificial intelligence* 3 (2020), p. 40.

[24] Guha-Sapir, Debby, Vos, Femke, Below, Regina, and Ponserre, Sylvain. "Annual disaster statistical review 2011: the numbers and trends". In: (2012).

[25] Håkansson, Anne. "Portal of research methods and methodologies for research projects and degree projects". In: Hamid R. Arabnia Azita Bahrami Victor A. Clincy Leonidas Deligiannidis George Jandieri (ed.), *Proceedings of the International Conference on Frontiers in Education: Computer Science and Computer Engineering FECS*. CSREA Press USA. Las Vegas USA, 2013, pp. 67–73.

[26] Hamarashid, Hozan Khalid. "Utilizing statistical tests for comparing machine learning algorithms". In: *Kurd J Appl Res* 6.1 (2021), pp. 69–74.

[27] Hochreiter, Sepp. "The Vanishing Gradient Problem During Learning Recurrent Neural Nets and Problem Solutions". In: *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 06.02 (1998), pp. 107–116. DOI: `10.1142/S0218488598000094`. eprint: `https://doi.org/10.1142/S0218488598000094`. URL: `https://doi.org/10.1142/S0218488598000094`.

[28] Hochreiter, Sepp and Schmidhuber, Jurgen. “Long Short-Term Memory”. In: *Neural Computation* 9.8 (1997), pp. 1735–1780.

[29] Hudlow, Michael D. “Technological developments in real-time operational hydrologic forecasting in the United States”. In: *Journal of Hydrology* 102.1-4 (1988), pp. 69–92.

[30] Jha, Abhas, Lamond, Jessica, Bloch, Robin, Bhattacharya, Namrata, Lopez, Ana, Papachristodoulou, Nikolaos, Bird, Alan, Proverbs, David, Davies, John, and Barker, Robert. *Five feet high and rising : cities and flooding in the 21st century*. Policy Research Working Paper Series 5648. The World Bank, May 2011. URL: https://ideas.repec.org/p/wbk/wbrwps/5648.html.

[31] Jordan, M I. “Serial order: a parallel distributed processing approach”. In: (May 1986). URL: https://www.osti.gov/biblio/6910294.

[32] Karim, Fazle, Majumdar, Somshubra, Darabi, Houshang, and Chen, Shun. “LSTM Fully Convolutional Networks for Time Series Classification”. In: *IEEE Access* 6 (2018), pp. 1662–1669. DOI: 10.1109/access.2017.2779939. URL: https://doi.org/10.1109%2Faccess.2017.2779939.

[33] Kim, Donghyun, Han, Heechan, Wang, Wonjoon, and Kim, Hung Soo. “Improvement of Deep Learning Models for River Water Level Prediction Using Complex Network Method”. In: *Water* 14.3 (2022). ISSN: 2073-4441. DOI: 10.3390/w14030466. URL: https://www.mdpi.com/2073-4441/14/3/466.

[34] Kim, Donghyun, Lee, Joonseok, Kim, Jongsung, Lee, Myungjin, Wang, Wonjoon, and Kim, Hung Soo. “Comparative analysis of long short-term memory and storage function model for flood water level forecasting of Bokha stream in NamHan River, Korea”. In: *Journal of Hydrology* 606 (2022), p. 127415. ISSN: 0022-1694. DOI: https://doi.org/10.1016/j.jhydrol.2021.127415. URL: https://www.sciencedirect.com/science/article/pii/S0022169421014657.

[35] Lee, Wei-Koon and Resdi, Tuan Asmaa Binti Tuan. “Neural Network Approach to Coastal High and Low Water Level Prediction”. In: *InCIEC 2013*. Ed. by Rohana Hassan, Marina Yusoff, Zulhabri Ismail, Norliyati Mohd Amin, and Mohd Arshad Fadzil. Singapore: Springer Singapore, 2014, pp. 275–286. ISBN: 978-981-4585-02-6.

[36] Lim, Bryan, Arik, Sercan O, Loeff, Nicolas, and Pfister, Tomas. “Temporal fusion transformers for interpretable multi-horizon time series forecasting”. In: *arXiv preprint arXiv:1912.09363* (2019).

[37] Lim, Bryan and Zohren, Stefan. “Time-series forecasting with deep learning: a survey”. In: *Philosophical Transactions of the Royal Society A* 379.2194 (2021), p. 20200209.

[38] Liu, Hanxiao, Dai, Zihang, So, David R., and Le, Quoc V. *Pay Attention to MLPs*. 2021. DOI: `10.48550/ARXIV.2105.08050`. URL: `https://arxiv.org/abs/2105.08050`.

[39] MedCalc Software Ltd. *Comparison of means calculator*. Apr. 2022. URL: `https://www.medcalc.org/calc/comparison_of_means.php`.

[40] Moshe, Zach, Metzger, Asher, Elidan, Gal, Kratzert, Frederik, Nevo, Sella, and El-Yaniv, Ran. *HydroNets: Leveraging River Structure for Hydrologic Modeling*. 2020. arXiv: `2007.00595 [cs.LG]`.

[41] Nemec, Jaromir. *Hydrological forecasting: design and operation of hydrological forecasting systems*. Vol. 5. Springer Science & Business Media, 2012.

[42] Paszke, Adam, Gross, Sam, Massa, Francisco, Lerer, Adam, Bradbury, James, Chanan, Gregory, Killeen, Trevor, Lin, Zeming, Gimelshein, Natalia, Antiga, Luca, Desmaison, Alban, Kopf, Andreas, Yang, Edward, DeVito, Zachary, Raison, Martin, Tejani, Alykhan, Chilamkurthy, Sasank, Steiner, Benoit, Fang, Lu, Bai, Junjie, and Chintala, Soumith. “PyTorch: An Imperative Style, High-Performance Deep Learning Library”. In: *Advances in Neural Information Processing Systems 32*. Ed. by H. Wallach, H. Larochelle, A. Beygelzimer, F. d’Alché-Buc, E. Fox, and R. Garnett. Curran Associates, Inc., 2019, pp. 8024–8035. URL: `http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf`.

[43] R.J., Hyndman and G, Athanasopoulos. *Forecasting: principles and practice, 2nd edition*. Melbourne, Australia: OTexts. URL: `OTexts.com/fpp2`.

[44] Raudys, Sarunas and Zliobaite, Indre. “The Multi-Agent System for Prediction of Financial Time Series”. In: June 2006, pp. 653–662. ISBN: 978-3-540-35748-3. DOI: `10.1007/11785231_68`.

[45] Ruineihart, David E, Hint., Geoffrey E, and Williams, Ronald J. "Learning Internal Representations By Error propagation". In: *ICS Report* 8506 (1985).

[46] Smith, Leslie N. "A disciplined approach to neural network hyper-parameters: Part 1 - learning rate, batch size, momentum, and weight decay". In: *arXiv preprint arXiv:1803.09820* (2018).

[47] Smith, Leslie N and Topin, Nicholay. "Super-convergence: Very fast training of neural networks using large learning rates". In: *Artificial intelligence and machine learning for multi-domain operations applications*. Vol. 11006. International Society for Optics and Photonics. 2019, p. 1100612.

[48] Smith, Leslie N. *Cyclical Learning Rates for Training Neural Networks*. 2015. DOI: 10.48550/ARXIV.1506.01186. URL: https://arxiv.org/abs/1506.01186.

[49] Smith, Stanley K. and Sincich, Terry. "An Empirical Analysis of the Effect of Length of Forecast Horizon on Population Forecast Errors". In: *Demography* 28.2 (1991), pp. 261–274. ISSN: 00703370, 15337790. URL: http://www.jstor.org/stable/2061279.

[50] Vaswani, Ashish, Shazeer, Noam, Parmar, Niki, Uszkoreit, Jakob, Jones, Llion, Gomez, Aidan N., Kaiser, Lukasz, and Polosukhin, Illia. *Attention Is All You Need*. 2017. DOI: 10.48550/ARXIV.1706.03762. URL: https://arxiv.org/abs/1706.03762.

[51] Wang, Tai-Yue and Huang, Chien-Yu. "Improving forecasting performance by employing the Taguchi method". In: *European Journal of Operational Research* 176.2 (2007), pp. 1052–1065. ISSN: 0377-2217. DOI: https://doi.org/10.1016/j.ejor.2005.08.020. URL: https://www.sciencedirect.com/science/article/pii/S0377221705007150.

[52] Wieringa, Roel J. *Design science methodology for information systems and software engineering*. Springer, 2014.

[53] Wu, Xian, Shi, Baoxu, Dong, Yuxiao, Huang, Chao, Faust, Louis, and Chawla, Nitesh V. "RESTFul: Resolution-Aware Forecasting of Behavioral Time Series Data". In: *CIKM* 18 (2018). DOI: 10.1145/3269206.3271794.

[54] Wu, Xianhua and Guo, Ji. “A New Economic Loss Assessment System for Urban Severe Rainfall and Flooding Disasters Based on Big Data Fusion”. In: *Economic Impacts and Emergency Management of Disasters in China*. Singapore: Springer Singapore, 2021, pp. 259–287. ISBN: 978-981-16-1319-7. DOI: `10.1007/978-981-16-1319-7_9`. URL: `https://doi.org/10.1007/978-981-16-1319-7_9`.

[55] Yadav, Basant and Eliza, K. “A hybrid wavelet-support vector machine model for prediction of Lake water level fluctuations using hydro-meteorological data”. In: *Measurement* 103 (2017), pp. 294–301. ISSN: 0263-2241. DOI: `https://doi.org/10.1016/j.measurement.2017.03.003`. URL: `https://www.sciencedirect.com/science/article/pii/S0263224117301513`.

[56] Ye, Rui and Dai, Qun. “Implementing transfer learning across different datasets for time series forecasting”. In: *Pattern Recognition* 109 (2021), p. 107617.

[57] Zappa, Massimiliano, Rotach, Mathias W, Arpagaus, Marco, Dorninger, Manfred, Hegg, Christoph, Montani, Andrea, Ranzi, Roberto, Ament, Felix, Germann, Urs, Grossi, Giovanna, et al. “MAP D-PHASE: real-time demonstration of hydrological ensemble prediction systems”. In: *Atmospheric Science Letters* 9.2 (2008), pp. 80–87.

[58] Zerveas, George, Jayaraman, Srideepika, Patel, Dhaval, Bhamidipaty, Anuradha, and Eickhoff, Carsten. “A Transformer-Based Framework for Multivariate Time Series Representation Learning”. In: *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. KDD ’21. Virtual Event, Singapore: Association for Computing Machinery, 2021, pp. 2114–2124. ISBN: 9781450383325. DOI: `10.1145/3447548.3467401`. URL: `https://doi.org/10.1145/3447548.3467401`.