Ministère de l'Enseignement
Supérieur et de la Recherche
Scientifique
Université de Carthage
Ecole Polytechnique de Tunisie

وزارة التعليم العالي و البحث
العلمي
جامعة قرطاج
المدرسة التونسية للتقنيات

Module: Projet Transversal

Dam's inflow forecasting with temporal deep learning model using IoT Data

Elaborated by: Naim HOUES
3rd year engineering student -SISY
(Signaux et Systèmes)

Supervised by: Dr. Takoua Abdellatif
Dr. Olfa Besbes

Academic year
2020-2021

Contents

List of Figures

Chapter 1

Introduction

Within the framework of the module "Projet Transversal" in the training of the 3rd year of multidisciplinary engineering studies at Ecole Polytechnique de Tunisie, we chose a project the field of time series forecasting and anomaly detection "Dam's inflow forecasting with temporal deep learning model using IoT Data".

The aim goal of this project is to forecast the water level of water in a hydrometric station to detect extreme events and anomaly observation.

This project was conducted in the semester of fall 2020 under the supervision of Dr. Takoua Abdellatif and Dr. Olfa Besbes.

In this report, we will present methods and techniques used to achieve an accurate daily water level forecasting of a hydrometric station and how to detect anomalies and extreme events in a time series problem.

The study is broken into three parts. We will start by exploring and analyzing the dataset and the time series problem. In the second part, we will present the deep learning techniques used to forecast the daily water level and how to predict extreme events. And in the third part, we will focus on how to detect the anomalies using the forecast results and the Autoencoders.

Chapter 2

Exploratory Data Analysis

2.1 Introduction

In this section we will do Exploratory Data analysis and we will introduce different characteristic of time series and how we can model them to obtain accurate forecasts.

2.2 About the dataset

In this project we have used a historical hydrometric data downloaded from The Water Survey of Canada site Web.

The Water Survey of Canada (WSC) is the national authority responsible for the collection, interpretation and dissemination of standardized water resource data and information in Canada [1].

After analyzing the Canadian regions weather conditions. We have selected to work with historical data of hydrometric in British Columbia.

2.3 Time Series Analysis

2.3.1 Definition

A time series as the name suggests is a series of data points with respect to time. The data points are indicators of some activity that takes place in a given period of time. So we have the time on the x-axis and the magnitude on the y-axis. An important thing to keep in mind when dealing with the Time series is the consistency of the interval of time. You can define this interval to whatever suits your needs and offers the most valuable insight. So the time interval can range from a millisecond, second, minute, hour, day, week, month, year or even a decade or century. There are no limits[2].

In this project, We will use a series of daily observation of Water level in a hydro metric station from January 2000 to December 2019.

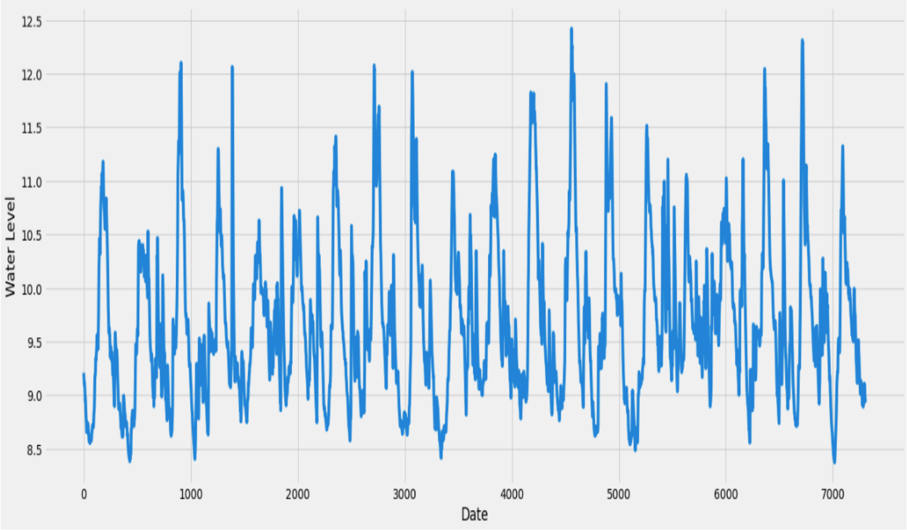


Figure 2.1: daily water level time series

2.3.2 Stationarity

A time series is said to be stationary, when it's statistical properties do not change over time. That is, it's mean, variance and autocorrelation are equally distributed over time.

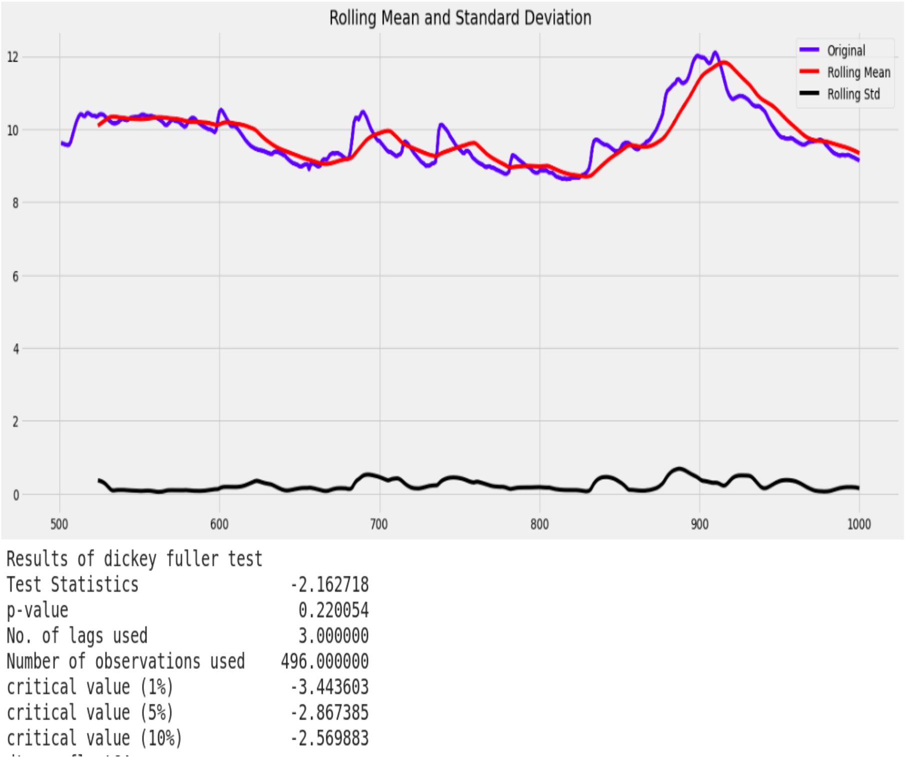
Stationarity is seen as a prerequisite of time series models. The reason for this is, when we try to make predictions on a time series, the statistical properties of the time series, that is the mean, variance and correlation should not be different than the ones currently observed. If the time series was non-stationary, making models and predictions on these properties would not give us an accurate result, as these properties would have changed[3].

So, let us check if the Water Level series is stationary or not. to do this i will use ADF (Augmented Dickey-Fuller) Test :

- Null Hypothesis: The series has a unit root

- Alternate Hypothesis: The series has no unit root.

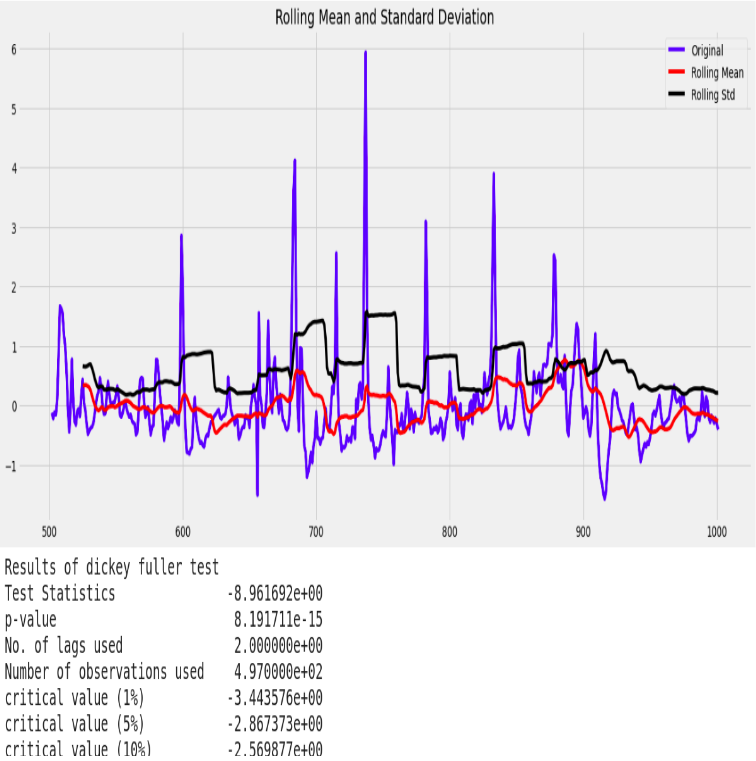
If we fail to reject the null hypothesis, we can say that the series is non-stationary.



```
Results of dickey fuller test
Test Statistics                 -2.162718
p-value                          0.220054
No. of lags used                 3.000000
Number of observations used    496.000000
critical value (1%)             -3.443603
critical value (5%)             -2.867385
critical value (10%)            -2.569883
```

Figure 2.2: stationarity test of the daily water level time series

As we can see in the above figure the p-value is higher than 5 percent. So, we can not reject the null hypothesis and we can conclude that our time series is non-stationary. we have computed the daily percentage of change of the water level and we have found that the the percentage of change time series is stationary. results are in the below figure.



```
Results of dickey fuller test
Test Statistics                -8.961692e+00
p-value                         8.191711e-15
No. of lags used                2.000000e+00
Number of observations used     4.970000e+02
critical value (1%)            -3.443576e+00
critical value (5%)            -2.867373e+00
critical value (10%)           -2.569877e+00
```

Figure 2.3: stationarity test of the daily percentage of change of the water level time series

2.3.3 Autocorrelation

Autocorrelation can be seen as the measure of internal correlation in a time series. It is a way of measuring and explaining the internal association between observations. You could have a very strong and positive association, that the time series at one point is going to be the same as a point in some time in the future. Or it could be a very strong and negative association, that is the time series at one point is going to be completely different at a point in the future. Autocorrelation is always measured between +1 and -1. With +1 indicating a strong positive association, -1 a strong negative association and 0 indicating no association[3].

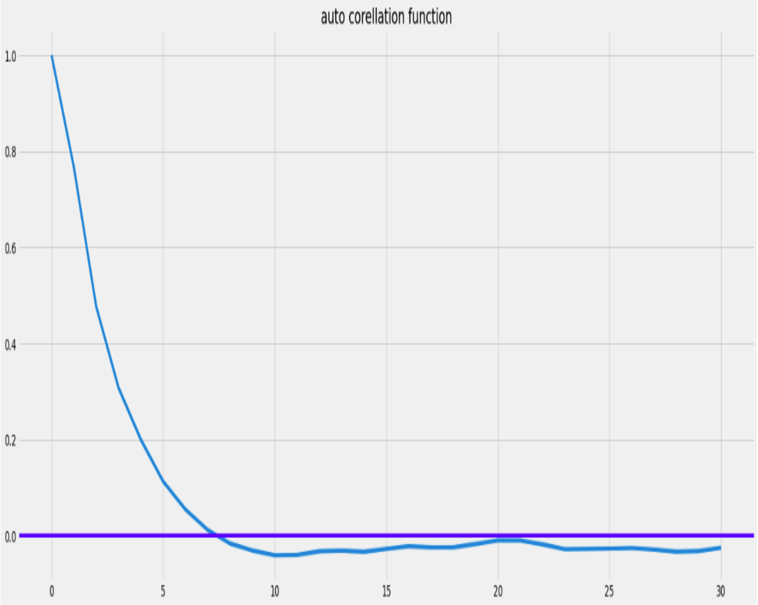


Figure 2.4: Autocorrelation of daily percentage of change of the water level time series

2.4 Climate Data

We have added to the hydrometric station historical dataset the historical climatic data.

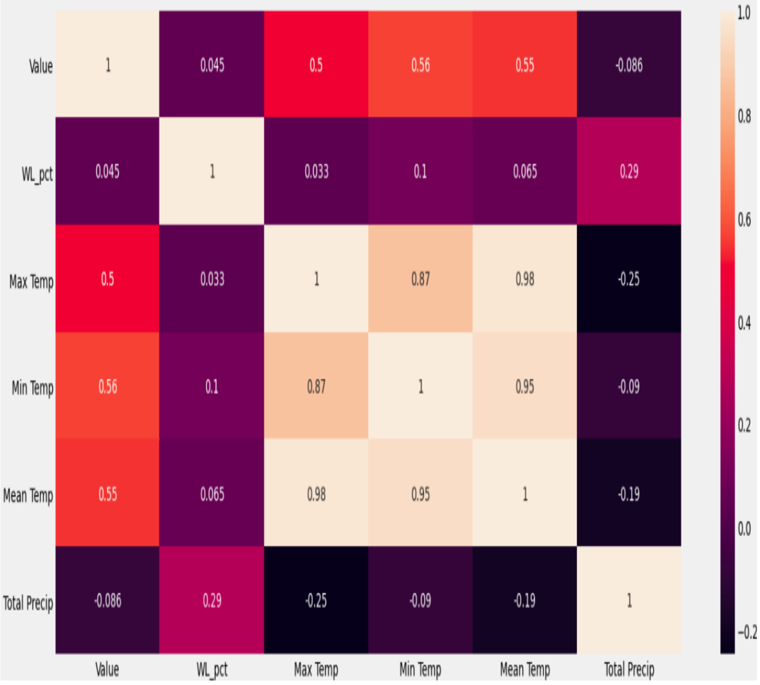


Figure 2.5: correlation matrix

We can see from the correlation matrix that the Water level (Value) is correlated with the temperature value but isn't correlated with percipation. In the other hand, We can note that the percentage of change is correlated with percipation value.

2.5 Conclusion

In this Section, We have explored the different characteristic of a our time series data. And, we have also analyzed the correlation between the water level and the climatic conditions.

Chapter 3

Forecasting and Extreme events

3.1 Introduction

In this section, we will present the different deep learning architecture used to obtain an accurate forecasting. we will also define the Extreme Event loss function used to help the model to predict the extreme events.

3.2 Models

In this project, we have used tow type of Deep-Learning models to forecasts the water level on a hydro-metric station. the first model is Stacked-LSTM and the second is LSTM-Auto-encoder.

3.2.1 Stacked LSTM

The original LSTM model is comprised of a single hidden LSTM layer followed by a standard feedforward output layer. The Stacked LSTM is an extension to this model that has multiple hidden LSTM layers where each layer contains multiple memory cells. Stacking LSTM hidden layers makes the model deeper, more accurately earning the description as a deep learning technique. Additional hidden layers can be added to a Multi-layer neural network to make it deeper. The additional hidden layers are understood to recombine the learned representation from prior layers and create new representations at high levels of abstraction[4].

In this project, we have used batch normalization layers to normalize the numerical features and embedding layer to encode the categorical features followed by a stacked tow LSTM layers with different hidden sizes and fully connected layer as output layer.

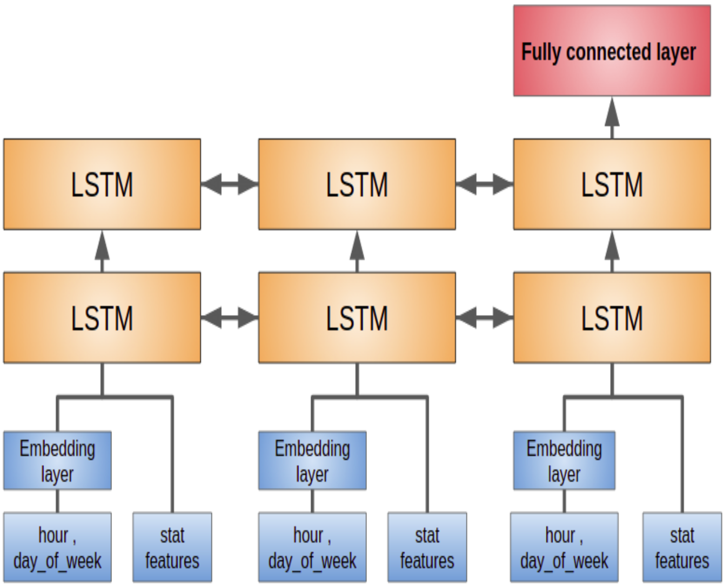


Figure 3.1: Stacked LSTM Architecture.

3.2.2 LSTM Auto-Encoder

The second model is based on tow part [5] :

- the first part is the Auto-encoder that takes as input the same features of the LSTM-Stacked model ant learn to reconstruct the input with this way it internally learns the best way to represent the input in lower dimensions. this part is composed of an encoder and a decoder, the encoder is responsible for learning how to represent the input into lower dimensions and the decoder learns how to rebuild the smaller representations into the input again.

- the second part is a model with the same architecture of the Stacked-LSTM that takes as input the learned representation of the original input features from the AUTO-Encoder.

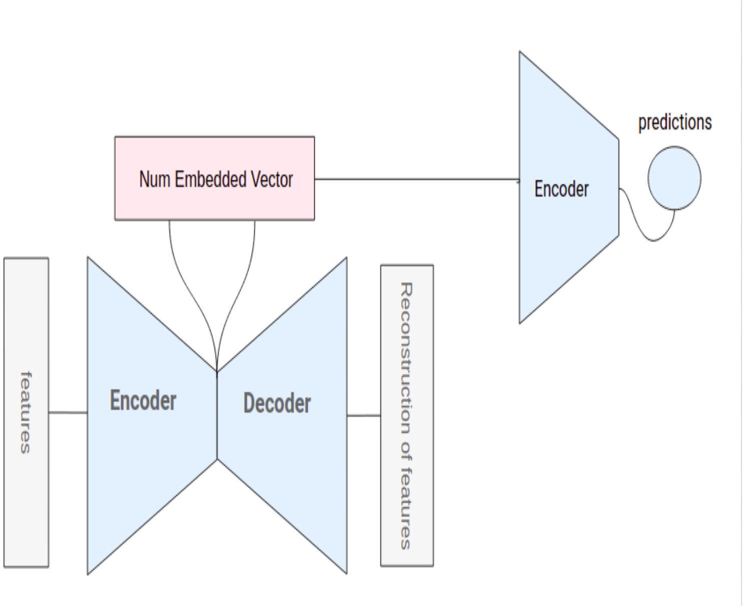


Figure 3.2: LSTM Auto-Encoder Architecture.

3.3 Extreme Events

3.3.1 Definition

The definition of an extreme event on time series problem depends on the task. In our task we can define an extreme event is a big variation on the water level of the hydro-metric station. below is the distribution of the daily percentage of variation of the water level[6].

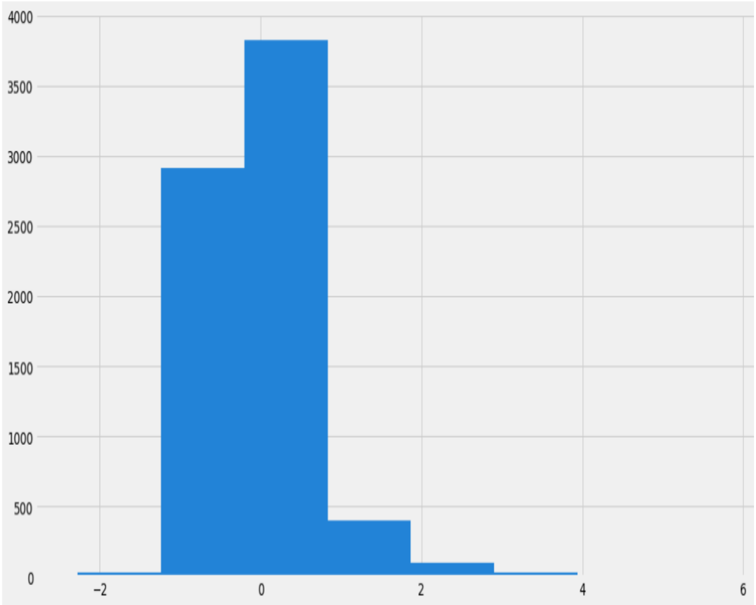


Figure 3.3: daily percentage of variation of the water level.

From the above distribution we can simply assume that an extreme event is an observation with daily variation of water level higher than 1.5 percent.

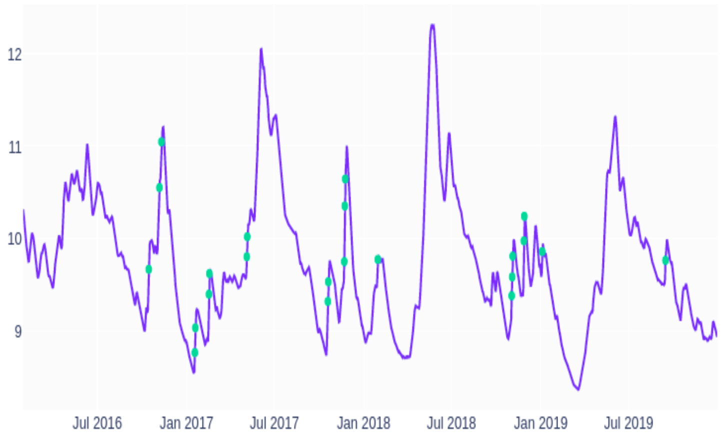


Figure 3.4: extreme events

3.3.2 Extreme Event Loss

Dealing with extreme event prediction very hard task for the model. So, to push the models to predict those events We have added a term to the loss function to penalize the model if he missed prediction of an extreme event. the mathematical format of the new loss function is.

$$loss(y, \hat{y}) = MAE(y, \hat{y}) + \alpha \, MAE(z, \hat{z}) \tag{3.1}$$

Where:

- MAE : mean absolute error
- y : Ground truth water level variation
- $\hat{y}$: predicted water level variation
- z : Ground truth water level variation of missed extreme event
- $\hat{z}$: predicted truth water level variation of missed extreme event

3.4 Evaluation

We have trained the models with NVidia K80 GPUs. with a batch size of 32 and using a early stopping mechanism to avoid the over-fitting. we used the implemented Adam optimizer by Pytorch.

3.4.1 Forecasting

We have trained the models on the first time using Mean squared error as loss function. and below is table that summarize the results.

	Stacked-LSTM	LSTM-AutoEncoder
MSE	0.00090	0.00084
RMSE	0.0300	0.0290
MAE	0.0173	0.0168
R2	0.9983	0.9984

Figure 3.5: models results

We can see that the results of the LSTM autoencoder model are better than LSTM stacked.

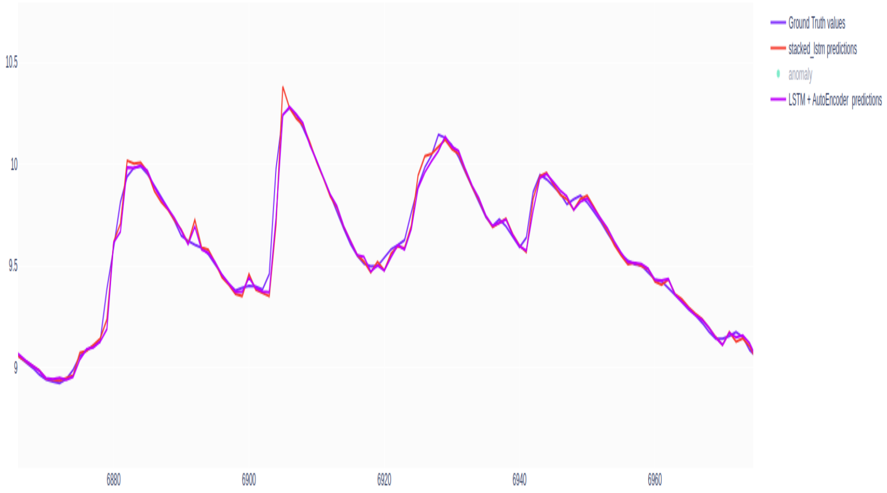


Figure 3.6: Forecast Results of the different models

3.4.2 Extreme Events

In a first time we have checked the ability of our model (LSTM AutoEncoder) to predict the extreme events and we found the below results.

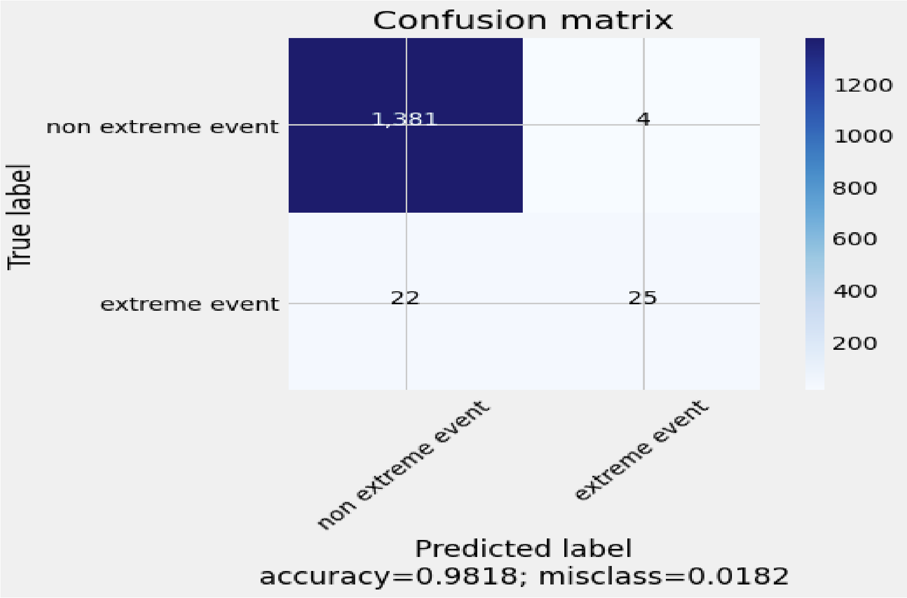


Figure 3.7: confusion matrix of predicted extreme events vs the Ground Truth

We Can see that the actual model detected 50 percent from the total extreme events with higher precision. but the problem here is that we want to predict as maximum as possible of those extreme events. So, to solve this problem we tried to retrain the model using the extreme event loss function and below are the obtained results.

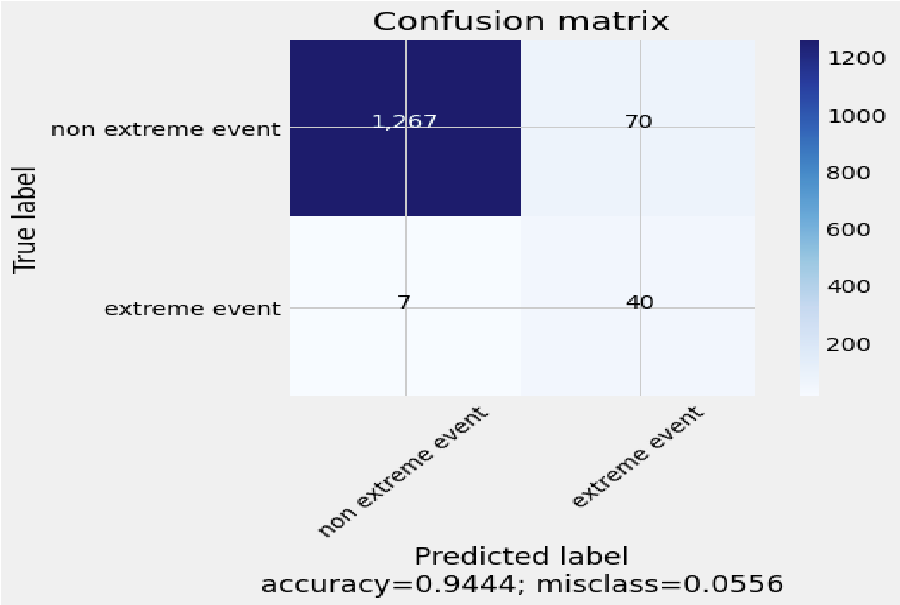


Figure 3.8: confusion matrix of predicted extreme events vs the Ground Truth with the trained model with Extreme event loss

We Can see that the retrained model succeed to detect 90 percent of the extreme events with lower precision we can try to increase the alpha parameter on the loss function to increase this percentage. In the other hand, we have observed that the forecast quality of the model trained with MSE is better than the model trained with Extreme Event Loss. So, we can use the first model (trained wit mean squared error) in the normal cases. and we can use the second model (trained with EVL) to detect the extreme events.

3.5 Conclusion

In this section, we have presented and analyzed the results of the deep learning models used to forecast the daily water level. we have found that the LSTM Autoencoder perform better than the LSTM stacked in this task. We have defined also the Extreme event loss we have found that the model trained with this loss function perform better in the detection of the extreme events.

Chapter 4

Anomaly detection

4.1 Introduction

In this Section, we will present some deep learning techniques used to detect the anomaly on a time series problem.

4.2 Anomaly detection with AutoEncoder

In this section we will try to detect anomalies in the historical data of our hydro metric station using an LSTM Auto-Encoder.

Autoencoders are an unsupervised learning technique, although they are trained using supervised learning methods. The goal is to minimize reconstruction error based on a loss function, such as the mean absolute error.

The steps we will follow to detect anomalies in Hydrometric data using an LSTM auto encoder:

- Train an LSTM autoencoder on Hydro metric data from 2000–01–01 to 2016–01–01. We assume that there were no anomalies and they were normal.
- Using the LSTM autoencoder to reconstruct the error on the test data from 2016–01–01 to 2019-12-31.
- If the reconstruction error for the test data is above the threshold, we label the data point as an anomaly.

We have trained the AutoEncoder and we computed the error of the reconstruction. Below is the histogram of the model output error.

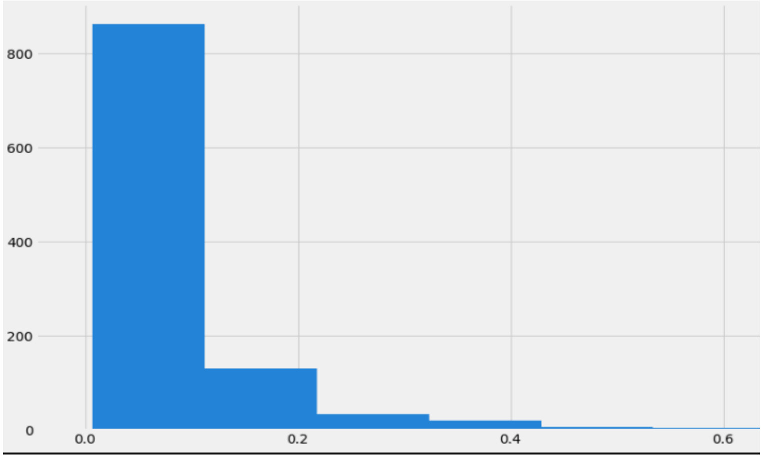


Figure 4.1: Reconstruction error histogram

We can assume from this histogram that we can classifier any prediction with error higher than 0.4 as anomaly observation.

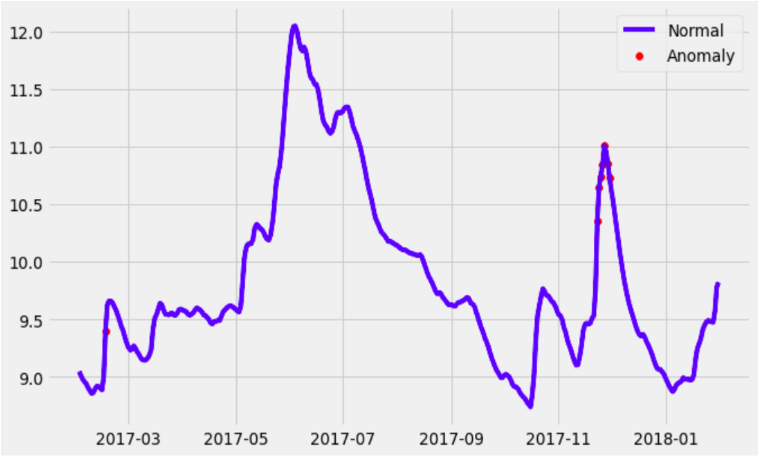


Figure 4.2: Actual data and Anomaly points

4.3 Anomaly detection with Time Series Forecasting

In this section, We will see about detecting anomalies with time series forecasting. Time series forecasting helps us in preparing us for future needs by estimating them with the current data. Once we have the forecast we can use that data to detect anomalies on comparing them with actual. In the previous chapter we have built tow deep learning models to forecast next day water level in the hydro metric station. In this approach we will compare the predictions of LSTM Auto-Encoder forecast and the ground truth values to detect the anomalies.

Steps We do to detect anomalies:

- Compute the error term(actual- predicted).

- Compute the rolling mean and rolling standard deviation(window is a week).
- Classify data with an error of 1.5,1.75 and 2 standard deviations as limits for low,medium and high anomalies.

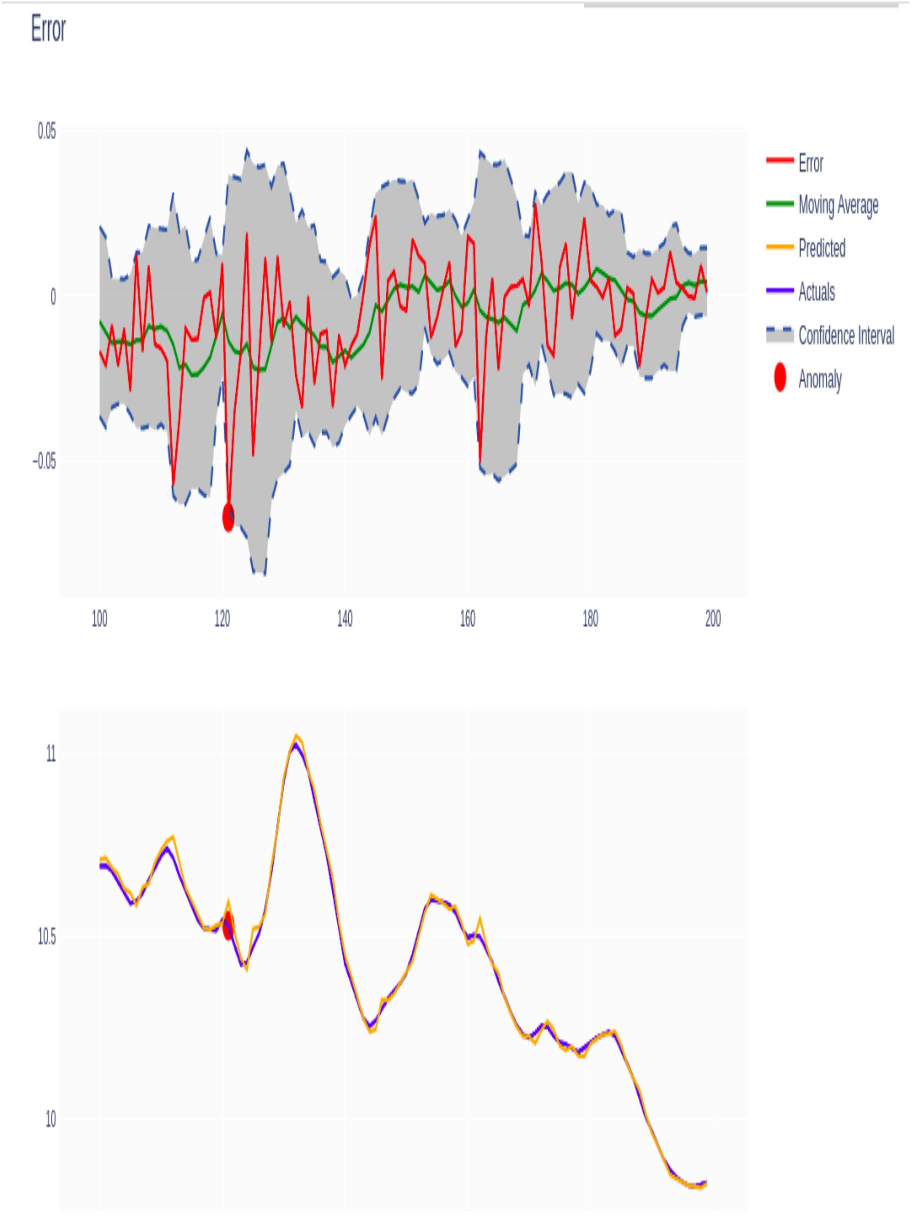


Figure 4.3: anomaly detection

We can see on the above Figure That the first plot has the error term with the upper and lower limit boundary specified and the second plot has actuals and predicted values with anomalies highlighted. By using a rolling mean and standard deviation here we are able to avoid continuous false anomalies during scenarios like extreme events.

4.4 Conclusion

In this section, we have presented tow techniques used to detect the anomaly in a time series the first technique is based on the Autoencoders and the second technique is base on time series forecasting.

Chapter 5

Conclusion

To sum up, our work is decomposed into three parts. In the first part, we started by an exploratory data analysis we have understood the characteristic of a our time series data and the correlation with the climatic data. In the second part, we have implemented tow deep learning models (LSTM Stacked and LSTM Autoencoder) to forecast the daily water level. then, we have explored the effect of the custom loss function on the detection of the extreme events. And in the last part, We have tried different deep learning techniques to detect the anomalies in our time series data.

Bibliography

1. *Water Survey of Canada* https://www.canada.ca/en/environment-climate-change/services/water-overview/quantity/monitoring/survey.html.
2. *What Is a Time Series?* https://www.investopedia.com/terms/t/timeseries.asp.
3. *How-To Guide on Exploratory Data Analysis for Time Series Data* https://medium.com/analytics-vidhya/how-to-guide-on-exploratory-data-analysis-for-time-series-data-34250ff1d04f.
4. *Stacked Long Short-Term Memory Networks* https://machinelearningmastery.com/stacked-long-short-term-memory-networks/.
5. *Extreme Event Forecasting with LSTM Autoencoders* https://towardsdatascience.com/extreme-event-forecasting-with-lstm-autoencoders-297492485037.
6. A framework for end-to-end deep learning-based anomaly detection intransportation networksNeema.