Birla Institute of Technology and Science,

Pilani

Project Report

On

Time Series and Data analysis of Water Data

Submitted in fulfilment of the course

Study Oriented Project – CSF266

Submitted by

NANDAN BHARATKUMAR PARIKH

2019A7PS0097P

Under the esteemed guidance and supervision of

Dr Pratik Narang

Faculty, Department of Computer Science & Information Systems

Birla Institute of Technology & Science, PILANI, 333031

Rajasthan, INDIA

Acknowledgement

I am highly indebted to my project guide and supervisor, Dr Pratik Narang, for initiating me into such a fruitful endeavour. His constant guidance and encouragement motivated me to go the extra mile throughout this project and sparked a profound interest in the topic. I am immensely grateful to him. His valuable inputs were vital to the success of this project.

Sincerely,

Nandan Bharatkumar Parikh

2019A7PS0097P

Introduction

Water Pollution is a significant threat in most rivers near big cities nowadays. We use data about pollutants in the watershed of the Mississippi River. For our task, we focused on the data for the following contaminants:

- Total suspended solids
- Total phosphorus
- Nitrate plus nitrite nitrogen
- Total Kjeldahl nitrogen
- Dissolved orthophosphate

For each of these pollutants, we do a multivariate time series analysis with the flow value of the river. Daily data of 2008-2019 available with very few missing values.

Data Preprocessing

Before running our forecasting models on the dataset, some essential preprocessing steps will affect our choice of models later on, so it is vital to go through these steps.

Missing Values

Several real-world datasets can have missing values for a variety of reasons. NaNs, blanks, and other placeholder used to represent them. Using a dataset for our case, many missing values to train a machine learning model can significantly impact the model's output. Some algorithms, like scikit-learn estimators, assume that all values are numerical and have meaning. One approach to this problem is to remove the observations with missing data. However, there is a danger that a data point containing helpful information will be lost. Imputing the missing values, i.e. infer those missing values from the available data, would be an ideal way.

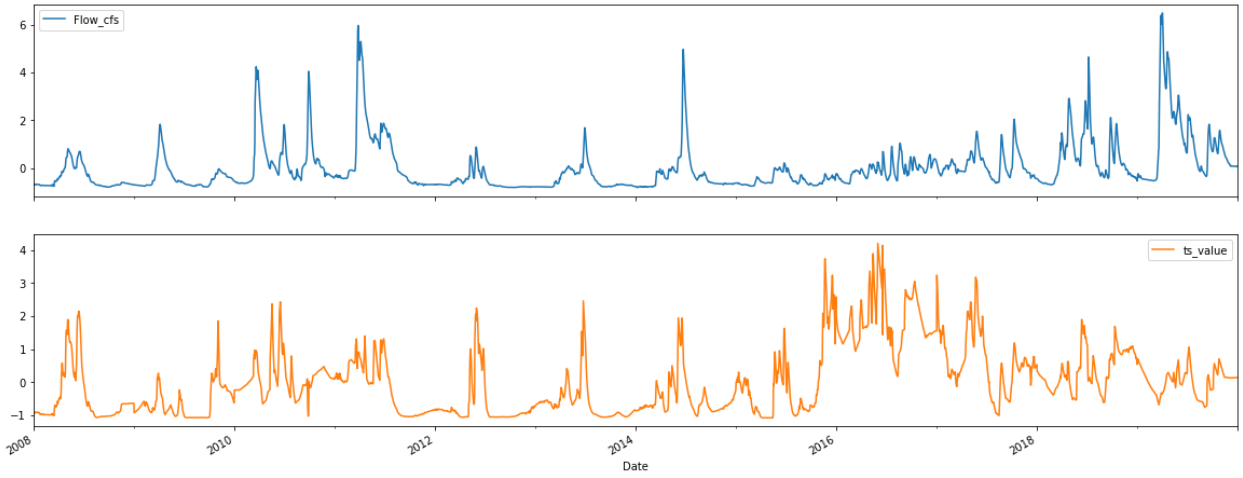
For our case, we saw through results that quadratic interpolation and kNN imputation both produced slightly better results than the other methods for the set of algorithms we had selected. For simplicity and negligible difference in outcomes, we used quadratic interpolation.

Feature Scaling

The magnitude, range, and units of the features are all different. This is a big challenge because a few machine learning algorithms are sensitive to these characteristics. Feature scaling is a technique for normalising a set of independent variables or data components. The process of rescaling values to [0,1] is known as normalisation. By scaling a feature X with its minimum and maximum values, the following formula transforms it.

$$X' = (X - min(X))/(max(X) - min(X))$$

Elements of the time series



The above figure shows the scaled values of the Flow and Total Suspended solids with respect to the temporal axis. On thorough inspection of this representation, we can make the following conclusions:

- **Trend:** Both the time series don't exhibit a continuous increasing or decreasing trend. Hence we conclude that the time series doesn't show any trend.
- **Seasonality:** Although our data does show fluctuations in values at intervals of time, the intervals are not regular enough to call the series seasonal. Hence we conclude that the time series has no seasonal component.
- **Cyclic:** We can see that the data exhibits a cyclical pattern. The values of both Flow and Total Suspended solids keeps fluctuating. Hence we can conclude that the data shows a cyclical pattern.

- **Irregular:** This component takes into account the randomness in the values. Our model aims to explain all the parts except the irregularity factor since we can not explain it. Our data shows a lot of irregularities with peak values and random level shifts. We can conclude that our data is highly irregular.

Stationarity of data

A stationary time series, in the most basic sense, is one whose attributes are independent of the time at which it is seen. Time series containing trends or seasonality, on the other hand, are not stationary since the trend and seasonality will alter the value of the time series at different times. We use the Augmented Dickey-Fuller (ADF) Test to check for stationarity.

Augmented Dickey-Fuller Test: It is a statistical test used to test whether a given time series is stationary or not. The more negative the ADF test statistic, the stronger is the rejection of the null hypothesis that there is a unit root at some level of confidence, i.e., data is stationary. More specifically this test helps to determine whether there is a trend in time series or not.

For our time series, we were able to reject the null hypothesis for all the variables and thus conclude that the time series is **stationary**.

Brief of the models used

Linear Regression

Linear regression models the relationship between the input and output as a linear function of the form $y = B_0+B_1x_1+B_2x_2+B_3x_3+ +B_nx_n$ (for inputs $x_1, x_2,...,x_n$). When there is more than one input, the function is called a hyperplane. Linear regression finds a relationship between the dependent and independent variables. The strength of relationship between the variables can be measured using the correlation coefficient (a value that ranges from -1 to 1). Higher the correlation, the absolute correlation value is closer to 1 and lower the correlation, the absolute correlation value is closer to 0. The sign determines the increasing or decreasing trend of the dependent variable. The coefficient of the linear regression line is found by minimizing the residual sum of squares (RSS) i.e., by minimising $Cost(B) = \Sigma(y - \hat{y})2$ where

$\hat{y} = B_0+B_1x_1+B_2x_2+B_3x_3+...+B_nx_n$.

Random Forests

Random forest builds multiple decision trees and merges their predictions to get a more accurate and stable prediction. The decision trees are built the same way as the previous algorithm. In a random forest, each tree learns from a sample of the training observations. Since the samples are drawn with substitution, or bootstrapping, several samples can be used several times in a single tree. The objective is to train each tree on different samples, even though each tree has a lot of variance with respect to a specific collection of training data, the average variance of the forest would be smaller, but not at the expense of bias.

LSTM

LSTMs stand for Long Short Term Memory networks that have the capacity of learning long term dependencies. They are a special kind of RNN that are widely used for sequence to sequence data. They were designed to tackle the long term dependency problem. They are a also capable of storing memory and can capture non-linearity in the data. LSTMs contain a cell state that runs through repeating LSTM modules and is responsible for the interaction between connecting LSTMs. For our task the best results were obtained from using 3 LSTM layers with 15 units per each layer.

Transformers

Transformers are a cutting-edge approach to Natural Language Processing (NLP). They work on the Multihead-Self-Attention (MSA) mechanism, which compares each token in an input sequence to every other token to gather information and learn dynamic contextual knowledge. Between its inputs, the Transformer learns an information-passing graph. Transformers largely overcome the vanishing gradient problem that Recurrent Neural Networks (RNNs) have in long-term prediction because they do not process their input sequentially. As a result, Transformers have been used on datasets having a lot of historical data, such as TSF.

Results

For the following section, the detailed analysis of the results has been given with respect to the multivariate time series analysis of Total Suspended Solids and the Flow since that was seen to be the most crucial factor. The results for other series also show similar results with slight variation. Unfortunately, due to time and computational limitations, we could not optimise DL based models (LSTMs and Transformers) on all the time series.

Metric Used

For our case, we used an 80-20 training and validation split respectively. To test the goodness of our model we forecasted the data for the dates in our validation set and computed the corresponding mean absolute percentage error (MAPE). We can represent MAPE as shown. Where A_t is our actual value at time t and F_t is our forecasted value. Evidently, the lower the MAPE, the better the model.

$$\text{MAPE} = \frac{100\%}{n} \sum_{t=1}^{n} \left| \frac{A_t - F_t}{A_t} \right|$$

Model	MAPE
Linear Regression	7.901
Random Forest	9.769
LSTM	11.765
Transformers	16.679

We can see from the above results that, in this case, Linear Regression obtained the best results, and Random Forest was also similar. The Deep Learning-based methods perform significantly worse than the traditional ML methods. This might be due to the lack of hyperparameter tuning. As future work, experimenting with different layers and hyperparameters may lead to better predictions. We also tried more complex architectures like Temporal Fusion Transformers but they performed significantly worse than the methods listed above.

Below you can see the forecasts by the above 4 models on the validation dataset

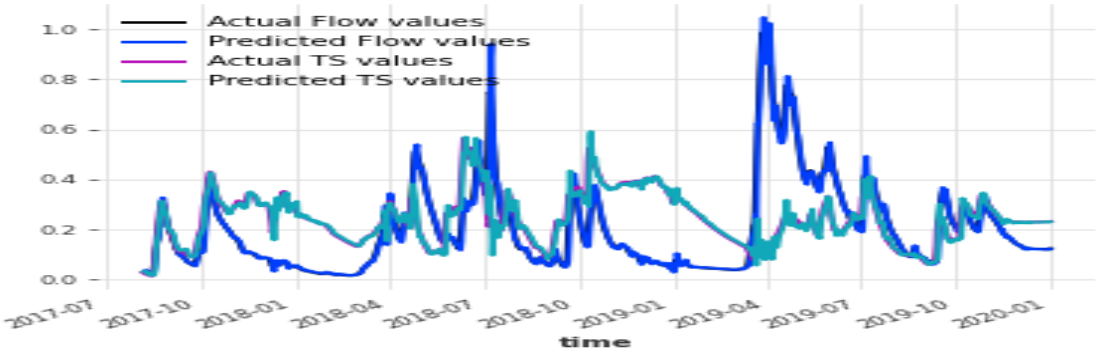


Figure 1 Forecast by the Linear regression model

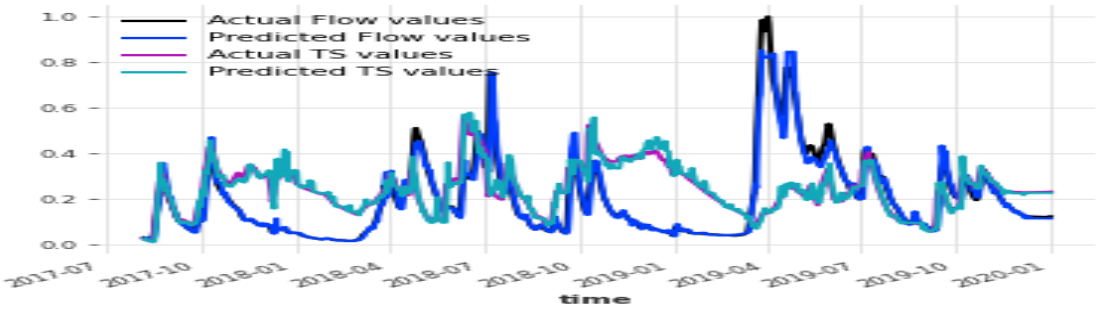


Figure 4 Forecasting using the random forest model

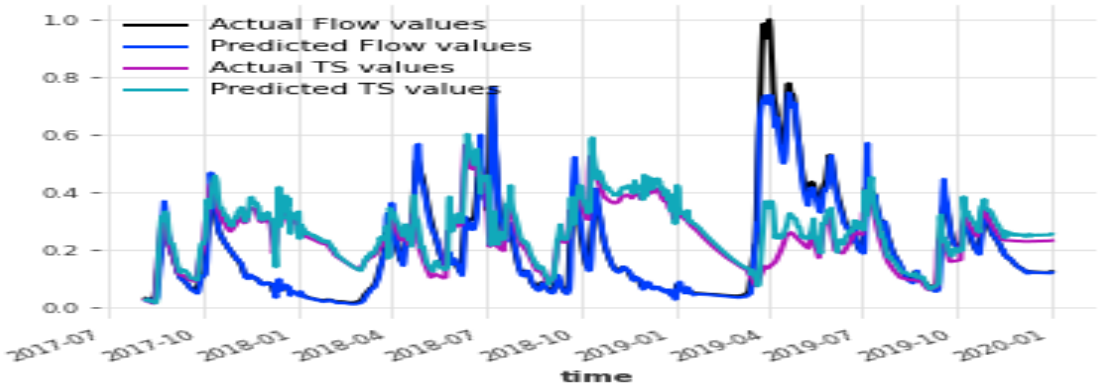


Figure 3 Forecasting using LSTM

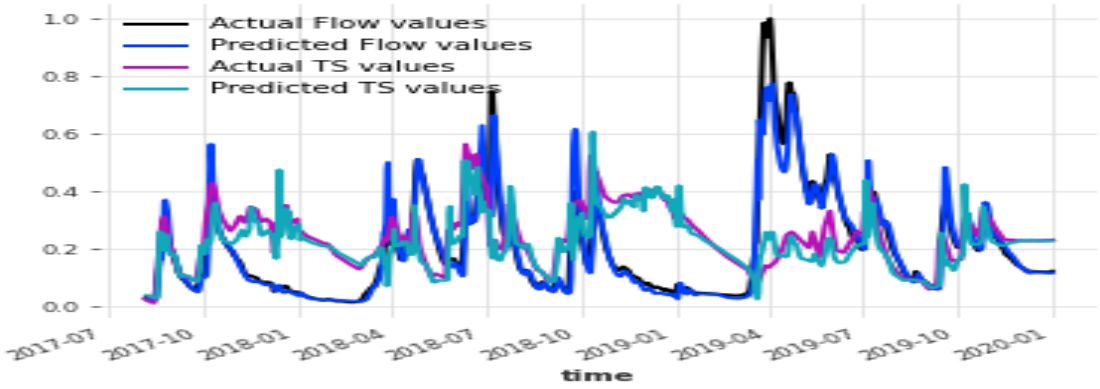


Figure 2 Forecasting using Transformer model