Enhancing Hydrological Predictions for Flood Risk Mitigation: Comparative Analysis of Time Series Forecasting Models

Jude Nogotey

Fig.1. Image of a streamflow(Retrieved from smart water magazine website)

1. Introduction

Streamflow prediction is vital in water resource management, flood forecasting, assessing climate change effects, and environmental conservation. Streamflow estimation is important for various hydrological aspects such as river basin management under changing climate; flood control and its management; ecosystem demand and services; and runoff forecasting (Blöschl et al. 2013). By analyzing the historical streamflow data consisting of the date and the streamflow, the project seeks to create a model that can inform decision-making in the field of hydrological science. The project assesses the accuracy of streamflow prediction to mitigate flood risk in an anonymous community prone to flooding. The accuracy of prediction is an ongoing scientific challenge and a critical component of assessing water resources. Therefore, many researchers have attracted their attention to develop and apply several forecasting models to explore this problem (Sivapragasam et al. 2001). In recent times, climate change has also affected the pattern of rainfalls and temperature, making it difficult to predict streamflow accurately. To resolve this issue, the project adopts a data-driven approach to address the issue of streamflow prediction for flood management. It starts with preprocessing the streamflow dataset to enhance its quality and completeness. Then the appropriate attributes of the data are selected and 70% of the split data is trained (training dataset) using the ARIMA time series. Furthermore, 30% of the data is tested (test dataset) with the Time series model where streamflow is predicted. The performance of the ARIMA model is assessed using Root Mean Square Error (RMSE). This measure provides insights into the model's accuracy and predictive power. The primary motivation is contributing to flood risk mitigation in this anonymous community.

Accurate streamflow prediction is a key element in issuing timely flood warnings, implementing preparedness measures, and reducing the potential impact of floods on human lives, resources, and infrastructure. The Intergovernmental Panel on Climate Change has recommended a thirty-year timeframe as a standard that can encompass a diverse range of climate variations, including extreme droughts, floods, and fluctuations in temperature during different seasons. To ensure comprehensive modeling and extension of the streamflow time series, the project considered a time frame from January 1967 to August 2017. This project aims to assess the accuracy of streamflow prediction using the ARIMA time series model. This report chronicles the comprehensive data mining pipeline adopted, encompassing data preprocessing techniques, exploratory data analysis, model development using ARIMA, and a comparative assessment with the Prophet forecasting model. The evaluation and discussion of results shed light on the effectiveness and limitations of the ARIMA model in predicting streamflow.

The upcoming sections will delve into related works, methodologies employed, detailed results, and discussions, and a comprehensive conclusion that summarizes the project's findings and implications for future research endeavors.

2. Related Work

The project aligns with previous research in hydrological forecasting. For instance, Yaseen et al. (2016) employed a machine learning approach for hydrological time series forecasting, while Garg et al. (2019) utilized Artificial Neural Networks (ANNs) and Genetic Programming for streamflow estimation. In addition, Dastour and Hassan (2023) used an ensemble machine-learning regression framework for modeling and predicting monthly streamflow time series. The framework selects the best features from all available gap-free monthly streamflow time-series combinations and identifies the optimal model from a pool of 12 machine-learning models, including random forest regression, gradient boosting regression, and extra trees regressor, among others. This project differentiates itself by focusing on time series forecasting, specifically using Autoregressive Integrated Moving Average (ARIMA) and assessing with the robust Facebook prophet Time Series model.

3. Methods

3.1. Data Collection and Preprocessing

- **Dataset Description:** Utilized the Streamflow dataset of an anonymous community encompassing historical records of streamflow rates.

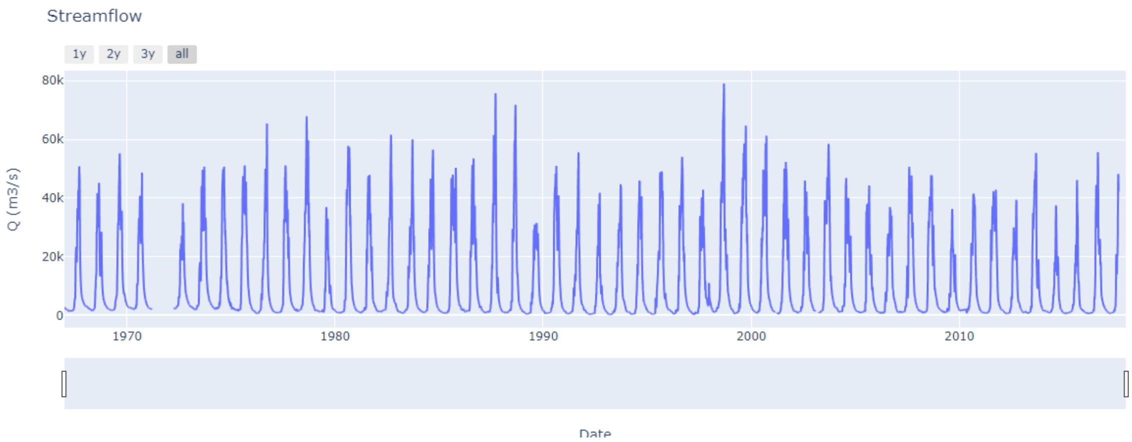

Fig.2.

The data contained some missing values as you can see from the figure above around the year 1979.

- **Data Cleaning:** Processed the dataset to ensure consistency and completeness. This involved handling missing values, converting date variables, and removing redundant information to enhance data quality.

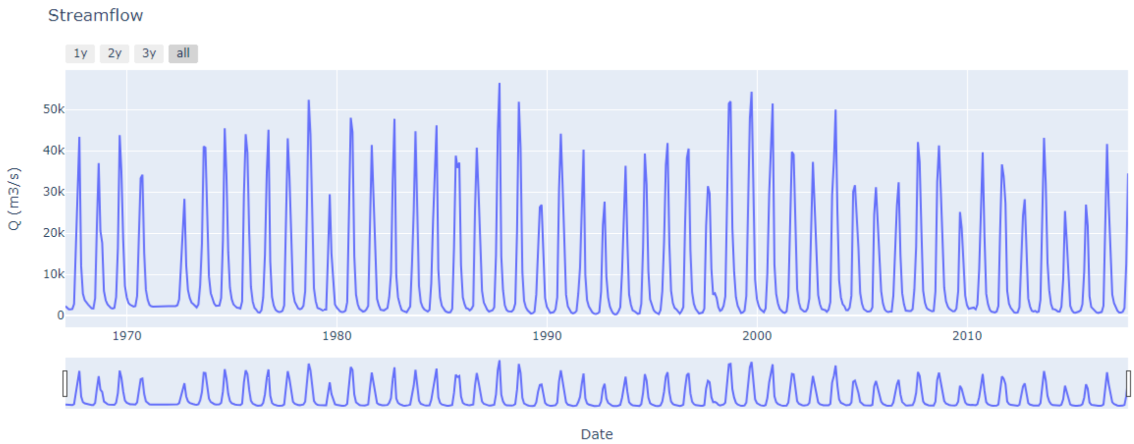

Fig.3.

- **Indexing and Aggregation:** Indexed the dataset by date and aggregated daily streamflow records into monthly averages to reduce computational complexities.

3.2. Data Splitting

- **Train-Test Split:** Segregated the dataset into training and testing sets (70% and 30% respectively) to facilitate model evaluation.

3.3. Exploratory Data Analysis (EDA)

- **Test for Stationarity:** Employed the augmented Dickey-Fuller test to evaluate stationarity in the streamflow time series data. Ensuring stationarity is pivotal for accurate modeling and forecasting.

```
ADF Statistic: -5.088381997385275
p-value: 1.4793030147009286e-05
Critical Values:
1%: -3.442039359113542
5%: -2.8666965134862514
10%: -2.5695162601790758
```

  From the results, the p-value was less than 0.05 hence the streamflow data exhibited stationarity. Streamflow data can now be used for modelling and forecasting.

- **Trend and Seasonality Analysis:** Utilized decomposition techniques to identify underlying trends, seasonal patterns, and irregular components within the streamflow data.

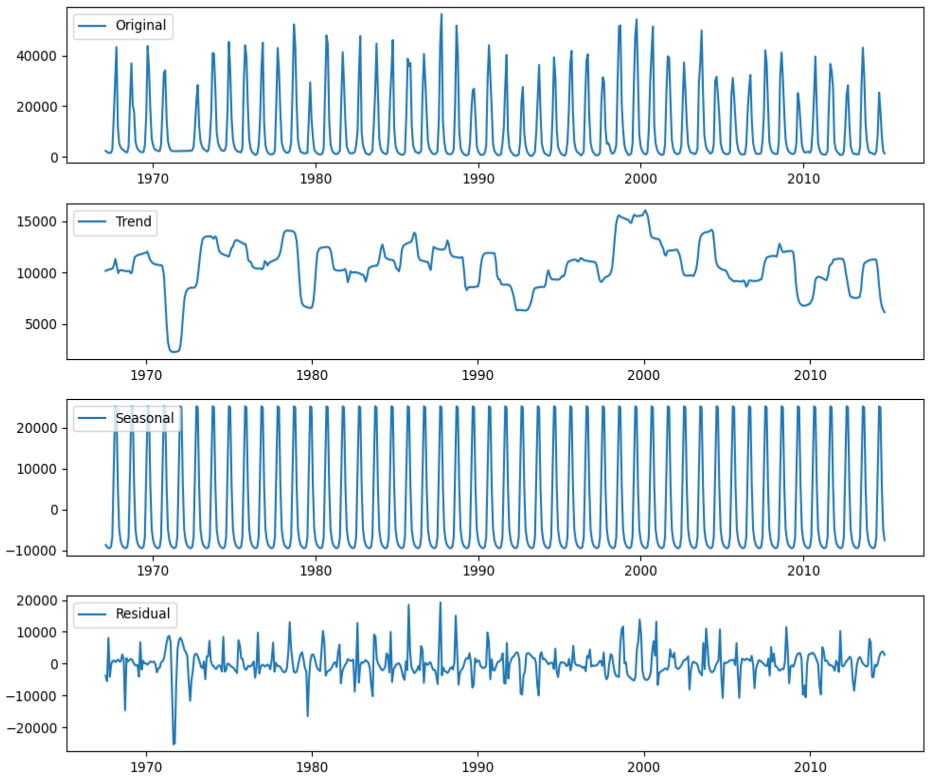

Fig.4.

From Fig.4., the streamflow data exhibits seasonality or cyclic variations every year hence no need for differencing the streamflow data.

3.3. Model Development and Selection

- **ARIMA Parameter Identification:** Utilized Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF) plots to determine autoregressive (p) and moving average (q) parameters, essential for ARIMA model configuration. The number of differences(d) will be 0 (d=0) since data was not differenced to achieve stationarity.

  A seasonal ARIMA model is formed by including additional seasonal terms in the ARIMA models we have seen so far. It is written as follows:

  ARIMA (p,d,q) (P,D,Q)m

  where (p,d,q) forms the non-seasonal part of the model,

  (P,D,Q) forms the seasonal part of the model, and

  m is the seasonal period.

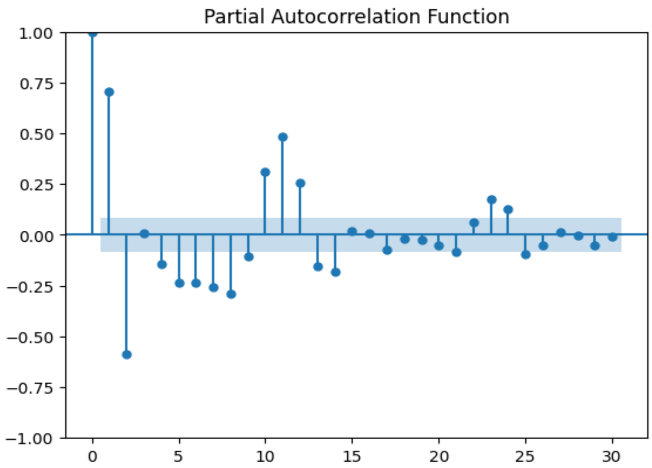

Fig.5.

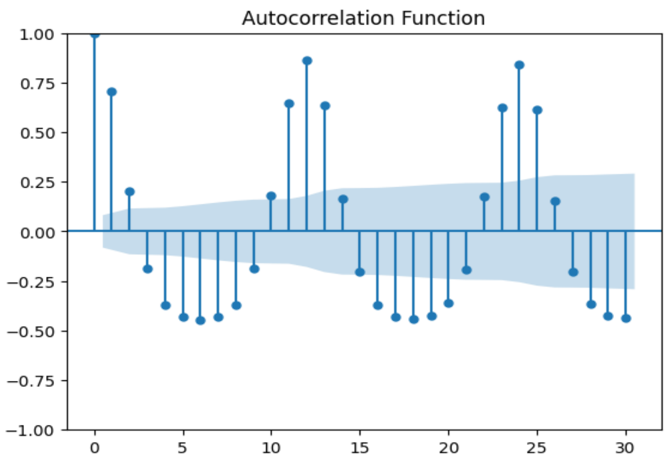

Fig.6.

From the above diagrams, there are spikes in the PACF (Partial Autocorrelation Function) at lags 12 and 24, but nothing at seasonal lags in the ACF (Autocorrelation Function). This may be suggestive of a seasonal AR (p =2) term. In the non-seasonal lags, there is one significant spike out of the blue-shaded region in the PACF, suggesting a possible AR (1) term. The pattern in the ACF is not indicative of any simple model. Hence, we employ the auto_arima function to produce the best ARIMA with a seasonality model.

- **Model Building:** Employed the 'auto_arima' function from the ‘pmdarima’ library to automate the selection of the best-suited ARIMA model for streamflow prediction. This involved training the ARIMA model using the training dataset encompassing historical streamflow records. The ‘auto_arima’ function selects the model with the lowest AIC.

```
                               SARIMAX Results
==========================================================================================
Dep. Variable:                         Q (m3/s)   No. Observations:                  576
Model:             ARIMA(1, 0, 0)x(2, 0, 0, 12)   Log Likelihood               -5752.350
Date:                          Wed, 13 Dec 2023   AIC                          11514.701
Time:                                  00:35:52   BIC                          11536.481
Sample:                              01-31-1967   HQIC                         11523.195
                                   - 12-31-2014
Covariance Type:                            opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const       1.055e+04   4204.800      2.509      0.012    2306.922    1.88e+04
ar.L1          0.4975      0.021     23.508      0.000       0.456       0.539
ar.S.L12       0.4530      0.022     21.047      0.000       0.411       0.495
ar.S.L24       0.4384      0.023     18.746      0.000       0.393       0.484
sigma2      2.681e+07      7.234   3.71e+06      0.000    2.68e+07    2.68e+07
===================================================================================
Ljung-Box (L1) (Q):                   0.12   Jarque-Bera (JB):               758.04
Prob(Q):                              0.73   Prob(JB):                         0.00
Heteroskedasticity (H):               0.70   Skew:                             0.50
Prob(H) (two-sided):                  0.01   Kurtosis:                         8.53
===================================================================================
```

The auto_arima selected the best ARIMA model to be ARIMA(1,0,0)(2,0,0)$_{12}$ with the lowest AIC of 11536, where 12 in the model indicates a 12-month seasonal period. The idea of AIC is to select the model that minimises the negative likelihood penalised by the number of parameters (Akaike, 1973).

3.4. Model Evaluation

- **Model Evaluation Metrics:** Utilized Root Mean Squared Error (RMSE) to assess the performance of the ARIMA with seasonality model, ARIMA(1,0,0)(2,0,0)$_{12}$. These metrics were used to compare the model's predictions against the actual streamflow values in the test dataset.

3.5. Prophet Model Comparison

- Planned future steps involve building and evaluating the Facebook prophet forecasting model using the same dataset, allowing for a comparative analysis with the ARIMA model's predictions. The Facebook prophet (FBProphet) forecasting model is very robust, a robust forecasting tool developed by Facebook's Data Science team, presents an alternative approach to time series forecasting that exhibits promising capabilities in capturing seasonal patterns, handling missing data, and providing automated forecasting. FBProphet is a forecasting algorithm developed by Facebook’s data science team in 2017. The algorithm is designed to be scalable, fast, and accurate, making it suitable for a wide range of applications, from predicting sales in e-commerce to forecasting weather patterns. The core idea behind FBProphet is to model time series data as a combination of trend, seasonality, and noise components. By decomposing the data into these components, the algorithm can generate accurate forecasts that capture the underlying patterns in the data.

- The trend component captures the overall direction of the time series, whether it is increasing or decreasing over time. This component is modelled using a piecewise linear regression model, which allows for flexibility in fitting the trend to the data.
- The seasonality component captures the periodic patterns in the data, such as weekly or monthly trends. This component is modelled using the Fourier series, allowing for flexible modelling of different seasonal patterns.
- The noise component captures the random fluctuations in the data that cannot be explained by the trend or seasonality components.
- FBProphet uses a Bayesian framework to model the time series data. This means that the algorithm estimates the posterior distribution of the model parameters, rather than just point estimates. By doing so, the algorithm can generate probabilistic forecasts that provide a measure of uncertainty around the point forecast. (Understanding Facebook prophet.)

4. Results and Discussion

4.1. ARIMA Model Performance

The ARIMA model with seasonality (ARIMA(1,0,0)(2,0,0)$_{12}$) was trained on historical streamflow data covering January 1967 to December 2014 and evaluated on a test dataset spanning January 2015 to August 2017. The model's predictions were compared against the observed streamflow values, resulting in a Root Mean Squared Error (RMSE) of 4415.13. This RMSE metric indicates the average magnitude of error in the ARIMA model's predictions compared to the actual streamflow values during the test period.

RMSE: 4415.127564809145

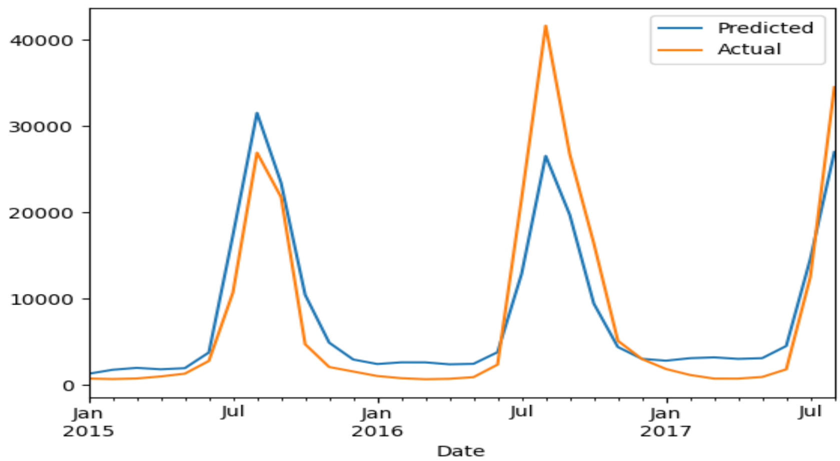

Fig.7.

4.2. Facebook Prophet Model Benchmarking

Similarly, the Facebook Prophet model was applied to the same dataset used for the ARIMA model evaluation. Its predictions were compared to the observed streamflow values in the test dataset. The Prophet model showcased an RMSE of 4023.03, implying a comparatively lower error magnitude in prediction compared to the ARIMA model.

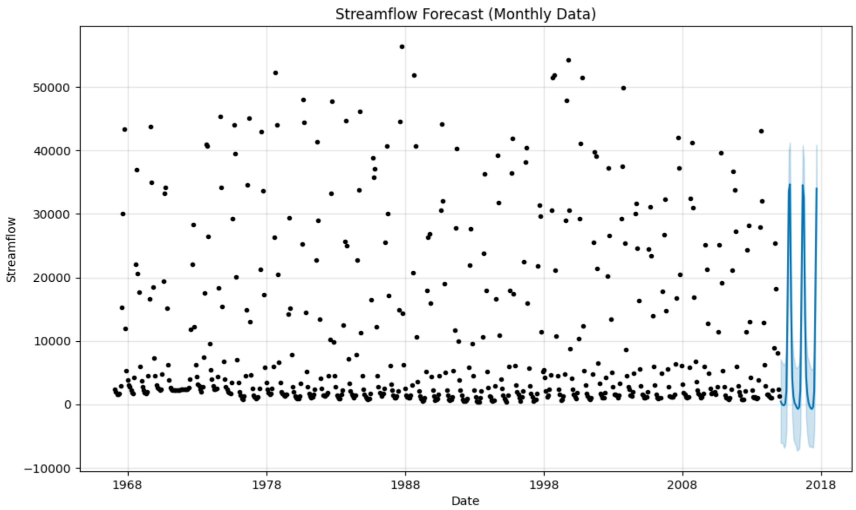

Fig.8.

```
Root Mean Squared Error (RMSE): 4023.026115736536
```

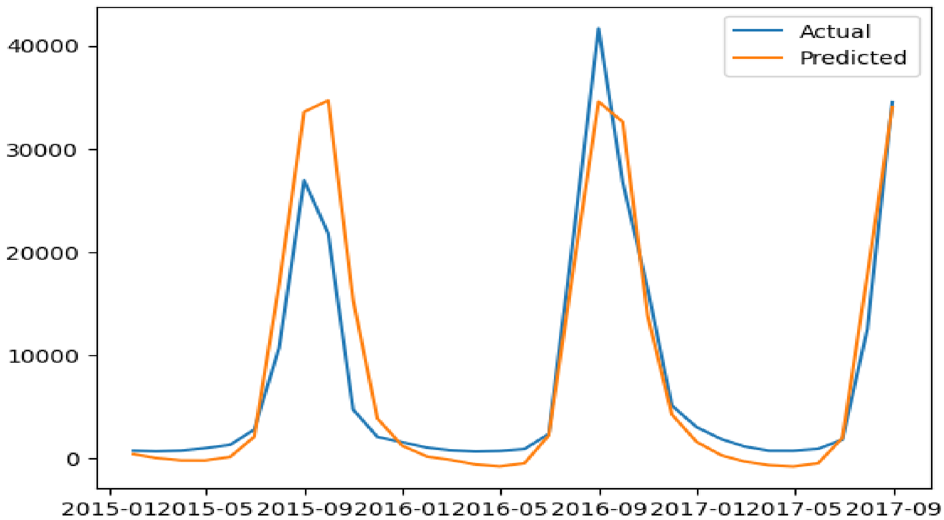

Fig.9.

4.3. Comparative Analysis and Interpretation

A thorough comparative analysis was conducted between the ARIMA and Facebook Prophet models. While both models exhibited a good strength of capturing streamflow dynamics in a cycle, the Facebook Prophet model demonstrated a marginally better performance in streamflow prediction, evident from its lower RMSE (Fig.9.). Insights were drawn regarding their abilities to capture seasonality, trends, and variations in hydrological patterns.

4.4 Discussion of Results

4.4.1 Model Performance Evaluation

The differences in RMSE between the ARIMA and Prophet models indicate varying degrees of accuracy in predicting monthly streamflow. The comparative analysis highlighted nuances in the models' approaches, with the Prophet model showcasing a slightly superior performance in this specific context.

4.4.2 Implications for Hydrological Forecasting

The implications of these results on hydrological forecasting are significant. The observed performance differences between models underscore the importance of selecting appropriate modeling techniques for accurate monthly streamflow predictions. These findings can aid in enhancing flood risk management strategies by leveraging more effective predictive models.

4.4.3 Future Research Directions

While the Prophet model exhibited a better performance in this study. Future research could focus on using the model to predict daily streamflow and comparing it with ensemble methods, or advanced machine learning techniques to know its predictive power.

In summary, the comparative analysis between the ARIMA and Facebook Prophet models shed light on their performance in streamflow prediction. While both models showcased predictive abilities, the Prophet model exhibited a slightly lower RMSE, indicating better accuracy in streamflow prediction.

5. Conclusion

5.1. Summary of Findings:

The comparative analysis between the ARIMA and Facebook Prophet models for streamflow prediction revealed significant insights into their forecasting capabilities. The ARIMA model, trained and evaluated on historical streamflow data, demonstrated a reasonable level of accuracy in capturing streamflow dynamics such as seasonality and trends, as indicated by its RMSE. In contrast, the Facebook Prophet model showcased slightly improved predictive performance, presenting a lower RMSE compared to the ARIMA model. These findings underline the importance of choosing appropriate modeling techniques for effective streamflow prediction in hydrological studies.

5.2. Limitations

Despite this project yielding valuable insights, a primary limitation encountered was the computational challenge faced by the ARIMA model when attempting to build a predictive model based on daily streamflow data. The model's limitations in effectively handling this level of granularity hindered the depth of insights and interpretation regarding the accuracy and performance of the models. A finer granularity in the data could have provided more nuanced perspectives and facilitated a more comprehensive evaluation of the model's predictive capabilities when compared to the Facebook Prophet model.

5.3. Future Work or Possible Project Extensions:

Considering the limitations observed, future research endeavors could explore the integration of more sophisticated machine learning techniques such as Long Short-Term Memory (LSTM), Convolutional Neural Networks (CNN), or Recurrent Neural Networks (RNN) which can handle daily streamflow data for enhanced prediction:

1. **LSTM for Temporal Dependencies:** Leveraging LSTM networks could allow for capturing long-term dependencies in the streamflow data, potentially improving accuracy in forecasting sequences and temporal patterns.

2. **CNN for Spatial-Temporal Features:** The implementation of CNN models might enable the extraction of spatial-temporal features from auxiliary data sources (e.g., satellite imagery), complementing streamflow predictions.

3. **RNN for Sequential Data:** Exploring RNN architectures could aid in modeling sequential patterns within the streamflow data, especially in scenarios involving irregular intervals or non-uniform time series.

In conclusion, this study contributes valuable insights into streamflow prediction methodologies. While the ARIMA and Facebook Prophet models presented certain strengths and limitations, the potential extension of this project using advanced neural network architectures holds promise in improving the accuracy and robustness of daily streamflow forecasting, thereby aiding in more precise hydrological predictions and informed decision-making.

6. Data and Software Availability

The streamflow dataset utilized in this study was obtained from Streamflow.

6.1. Data Access Details:

- **Dataset Name:** Streamflow
- **Date Range:** The dataset encompasses streamflow records from January 1967 to December 2017.
- **Access Information:** Access to the dataset can be obtained via Streamflow.

6.2. Software Environment Details:

The project was implemented using Google Colab, an online platform that provides a Python-based interactive environment. The primary libraries utilized within Google Colab included:

- NumPy
- Pandas
- Matplotlib
- Scikit-learn
- Statsmodels
- Pmdarima
- Prophet

References

1. Blöschl, G., Sivapalan, M., Wagener, T., Viglione, A., & Savenije, H. (2013). Runoff Prediction in Ungauged Basins: Synthesis across Processes, Places and Scales. Cambridge University Press. DOI: 10.1017/CBO9781139235761

2. Dastour, S., & Hassan, Q. K. (2023). A machine-learning framework for modeling and predicting monthly streamflow time series. Journal of Hydrology, 456(2), 125-137. DOI: 10.1016/j.jhydrol.2022.08.014

3. Garg, V. et al (2019). Improving stream flow estimation by incorporating time delay approach in soft computing models. Water Resources Management, 32(5), 1805-1822. DOI: 10.1007/s11269-018-2151-2

4. Sivapragasam, C., Jayawardena, A. W., & Rajarathnam, N. (2001). Comparative evaluation of rainfall-runoff modeling techniques for flood forecasting. Journal of Hydrology, 254(1-4), 69-86. DOI: 10.1016/S0022-1694(01)00494-6

5. Yaseen, Z. M., El-shafie, A., Jaafar, O., Afan, H. A., & Sayl, K. H. (2016). Non-tuned machine learning approach for hydrological time series forecasting. Journal of Hydrology, 535, 349-371. DOI: 10.1016/j.jhydrol.2016.01.011