Monthly River Flow Forecasting in Kelantan with ARIMA and Deep Learning LSTM BY

Teoh Xu Xian

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF INFORMATION SYSTEMS (HONOURS) DIGITAL ECONOMY TECHNOLOGY

Faculty of Information and Communication Technology

(Kampar Campus)

June 2025

COPYRIGHT STATEMENT

ACKNOWLEDGEMENTS

I would like to extend my heartfelt gratitude to my supervisor, Dr. Abdulkarim Kanaan Jebna, for granting me the opportunity to undertake this project. This experience marks a significant first step toward building a future career in the field of data analytics. I am truly grateful for your guidance and support.

In addition, I wish to sincerely thank my parents and family for their unwavering love, encouragement, and support throughout the duration of this academic journey.

ABSTRACT

This project aims to develop a web-based river flow forecasting system tailored to Malaysian rivers by integrating two prominent time series forecasting models: ARIMA and Long Short-Term Memory (LSTM). The system focuses on Sungai Kelantan and Sungai Sokor, leveraging daily river discharge data sourced from the Department of Irrigation and Drainage (DID) Malaysia. The core objective is to deliver accurate monthly forecasts through a user-friendly interface powered by Streamlit. The methodology follows the CRISP-DM framework, including systematic data preprocessing, model training, and evaluation using Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and classification metrics. Forecast accuracy, especially for extreme flow conditions, is validated through comparative performance analysis. The final product allows real-time river flow forecasting with interactive model selection and visualization, contributing to improved decision-making for flood preparedness and water resource management in Malaysia.

Area of Study :

Time Series Forecasting, Web Application Development

Keywords :

River Flow Forecasting, ARIMA, LSTM, CRISP-DM, Streamlit, Time Series, RMSE, MAE

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

LIST OF ABBREVIATIONS

ARIMA	Autoregressive Integrated Moving Average
LSTM	Long Short-Term Memory
SVR	Support Vector Regression
ANN	Artificial Neural Networks
AIC	Akaike Information Criterion
RMSE	Root Mean Square Error
MSE	Mean Square Error
ADF	Augmented Dickey-Fuller
EDA	Exploratory Data Analysis

Chapter 1

Introduction

This chapter introduces the context and motivation behind the river flow forecasting project. It outlines the challenges faced in existing hydrological prediction systems, the objectives of the study, the scope and direction of development, and the contributions made by this project. The final section provides a roadmap of the remaining chapters in this report.

1.1 Problem Statement and Motivation

River flow forecasting plays a critical role in flood mitigation, infrastructure planning, and water resource management, particularly in flood-prone regions such as Kelantan, Malaysia. Traditional hydrological models often rely on linear assumptions that struggle to capture the complex temporal, and seasonal variations present in Kelantan river basins[1]. Furthermore, existing commercial systems like GloFAS, RiverMamba, and HydroMax either lack regional specificity or are inaccessible to public agencies and local users[23],[27],. This highlights the need for a customizable, interpretable, and accessible forecasting solution tailored to Malaysian Kelantan rivers. By developing a forecasting system that integrates both ARIMA and LSTM models within a Streamlit-based platform, this project aims to close the gap between advanced modelling techniques and practical deployment for river management[22]. Furthermore, the river data acquired also required careful handling, as the data obtained often has issues, such as raw data obtained mostly has missing values, which is a common problem when dealing with real-world data. [19]. This thesis is motivated by the pressing need for reliable river flow forecasting in the Kelantan region and the lack of comparative studies evaluating ARIMA and LSTM models specifically for the rivers in Kelantan.[1],[19].

1.2 Objectives

To develop a river flow forecasting system for Kelantan rivers using ARIMA and LSTM models.

To evaluate and compare the forecasting performance of ARIMA and LSTM models using statistical accurate metrics specifically RMSE and MAE.

To study ,preprocess and transform 2 Kelantan Rivers(Sungai Kelantan and Sungai Sokor) for forecasting task.

1.3 Project Scope and Direction

This project focuses on the development of a forecasting system for two rivers: Sungai Kelantan and Sungai Sokor, using daily river flow data from January 2000 to April 30 2020. The models are trained and tested separately for each river. The system supports forecasting up to 8 months (ARIMA) or 30 days (LSTM) and provides evaluation outputs and visualizations. The system is implemented using Python (Streamlit, TensorFlow, pmdarima), deployed on Streamlit Cloud, and developed according to the CRISP-DM methodology. The system does not attempt to predict rainfall or meteorological conditions but instead uses historical river discharge data only

1.4 Contributions

This project contributes a fully functional and customizable forecasting system capable of serving both research and operational needs. It offers a comparative performance analysis between ARIMA and LSTM models for the Kelantan rivers. This project also hopes to enhance accessibility to river forecasting for regional authorities, researchers, and educators. This project aims to create a clean, modular system architecture with the ability to load models dynamically and interactively visualise forecast results.

1.5 Report Organization

This report is structured into seven chapters. Chapter 1 introduces the project background, motivation, objectives, scope, contributions, and structure. Chapter 2 presents a literature review of past forecasting studies and evaluates existing systems such as GloFAS, RiverMamba, and HydroMax. Chapter 3 details the CRISP-DM methodology used in the project, including model design and preprocessing workflows. Chapter 4 explains the system design, including component specifications and data pipelines. Chapter 5 presents the implementation details, code structure, and frontend interface. Chapter 6 evaluates the model performance, discusses project challenges, and assesses whether the objectives were met. Chapter 7 concludes the report and suggests recommendations for future work.

Chapter 2
Literature Review

2.1 Previous works on River Flow Forecasting

This chapter reviews existing river flow and flood forecasting systems to contextualize the development of the proposed daily forecasting model for Malaysian rivers. The focus is on three representative systems – the Global Flood Awareness System (GloFAS), RiverMamba, and HydroMax – examining their purpose, methods, data inputs, coverage, usability, and reported performance. By analyzing these systems at a system level, we identify key features and limitations. The comparison highlights how each system's design and capabilities relate to the needs of a Malaysian daily river forecast and informs critical evaluation of their strengths and weaknesses in that context.

2.1.1 ARIMA models

From previous research [3], the ARIMA model has been used for river flow forecasting due to its ability to model linear relationships in time series data. From previous research, ARIMA is said to be effective for short-term predictions and capturing additional information that may not be extracted through traditional regression methods [3]. Furthermore, from previous research [5], ARIMA model is also used to predict hemorrhagic fever incidence from 2013 to 2018. Both ARIMA and LSTM were evaluated using rolling forecasting which forecast step used the most recent actual data. It is found out that the ARIMA model performed better for monthly and weekly forecasts, while LSTM is excelled in daily predictions. Best ARIMA models found in the research were ARIMA (2,1,1) for monthly data, ARIMA(1,1,3) for weekly data and ARIMA(5,0,1) for daily data[5].

2.1.2 LSTM models

Research [4] shows that there are more studies related to LSTM models, in which LSTM is a type of recurrent neural network that is designed to capture long-term dependencies and nonlinear patterns in data. Several studies [6,7] have demonstrated that LSTM consistently achieves better accuracy than traditional models such as Support Vector Regression (SVR) and Multilayer Perceptron (MLP) in forecasting river flow and water level fluctuations. Additionally, research has shown that LSTM outperforms conventional Artificial Neural Networks (ANN), particularly in its ability to detect peak flow events and handle datasets characterized by high variability and noise [8,9].LSTM also proves superior performance in predicting streamflow data several days in advance if compared with Support Vector Machine (SVM) and Gradient Boosting (GB) [20]. Based on [20], when predicting streamflow value 2 to 3 days ahead, LSTM proves to have a higher F1 score in several river streamflow evaluations.

2.1.3 Other relevant studies

There are also other studies regarding using different models for predicting streamflow. For example, there is a forecasting model that integrates a hybrid approach where ARIMA and LSTM are combined. The author uses data from the Chuhe river basin to conduct their research, whose dataset spanned from 1st January 2010 to 31st December 2015. During the test, it is found that ARIMA is used to test where the data becomes statistically stable over time. AIC (Akaike Information Criterion) is used to determine the best parameters of (p,d,q) for ARIMA. LSTM is then used to capture the forecast errors, which were obtained through comparing ARIMA forecasts and actual values. The results obtained demonstrate high forecast accuracy, which is 0.0078 for MSE, this indicates the model's predictions are very close to the actual data. After comparing with standalone ARIMA and LSTM, its performance outperforms all of them[10]. Besides that, [20] dl. Other studies also use ANN, SVM, and LSTM to predict streamflow data. One key finding is that ANN models perform the best overall across

Peninsular Malaysia(11 rivers), but LSTM still has some of its advantages in predicting streamflow in Sungai Kepis and Sungai Perak areas. One main difference of this study is that it is a short-term prediction that predict using 1 day, 2 days, and 3 days of previous streamflow data. Based from the experiment, it is stated that ANN3 (using 3 days of previous streamflow data) performs the best after conducting the test using 11 rivers which it have MAE of 7.2175 m^3/s, RMSE of 13.9196 m^3/s, and R^2 of 0.8851[20].Furthermore, there are studies regarding comparing ARIMA and LSTM in predicting economic and finance data. Its split its data into 70% training model and 30% for testing the models.After conducting the studies, the results show that for financial data LSTM achieved RMSE(Root Mean Square Error) 64.213 while ARIMA showed 511.481, showing large indication of prediction errors. Similarly,on economic data RMSE, which uses LSTM, shows 0.936 while ARIMA shows 5.999, showing a reduction of approximately 84.394%[3].

2.2 Review of Existing System

2.2.1 GloFAS (Global Flood Awareness System)

The Global Flood Awareness System (GloFAS) was developed by ECMWF under the Copernicus Emergency Management Service to provide ensemble-based streamflow forecasts globally [21], [22]. It integrates meteorological inputs from ECMWF's ensemble prediction systems into the LISFLOOD hydrological model, producing forecasts up to 30 days ahead [21]. GloFAS data and visualizations are publicly accessible through its web portal [24], and a global discharge reanalysis product (GloFAS-ERA5) supports historical flood research [22]. Studies have demonstrated its ability to deliver probabilistic skill scores at the global scale [23].

2.2.2 RiverMamba

RiverMamba is a recent deep learning-based flood forecasting model that uses state-space architecture and Mamba blocks to predict river discharge globally [25]. It is trained on GloFAS-ERA5 and ERA5-Land reanalysis data and outperforms both physics-based and LSTM baselines across several river basins [25], [26]. Although not yet operational, its performance metrics, including RMSE and MAE, have been thoroughly benchmarked in controlled experiments [25], [27].

2.2.3 HydroMax

HydroMax is a region-specific river forecasting tool operational in the Meuse Basin since the 1990s [28]. It relies on Hammerstein-type models and a dense gauge network to provide near-real-time warnings [29]. While not globally deployable, its performance in regional use has been validated through case studies and internal assessments [28], [29].

2.3 Strengths of ARIMA and LSTM Models.

2.3.1 Strength of ARIMA

The advantages of using ARIMA lie in its ability to model and forecast time series data with linear relationships. It is suitable for short-term forecasting. Its ability to predict temporal dependencies using historical datasets makes it useful for linear predictions, particularly when the data needs to be stationary. ARIMA model is widely used for a variety of forecasting jobs across multiple fields due to its adaptability and efficiency in managing linear relationships [3,7,11]. The model is effective at handling stationary data once data transformation is applied through the differencing method. Its clear structure through its parameters (p,d,q) that can be adjusted based on observed autocorrelation and partial autocorrelations in the data. When the data needed to be handled is not abundant, ARIMA can perform extremely well [5].

2.3.2 Weakness of ARIMA

One of the main weaknesses is its ability to only model linear relationships between observations. In real-world scenarios, many time series have non-linear relationships but ARIMA requires time series to be stationary so any sudden changes or trend in variance needs transformation may lead to oversimplification of the inherent complexity of the data [3,5,18].

It struggles to do multi-step ahead forecasts in which data dependency is dynamic, especially error accumulation becomes an issue during rolling forecasts[3,5].

2.3.3 Strengths and Weakness of LSTM

The advantage of using LSTM it excels in capturing nonlinear relationships and long-term dependencies in data. LSTM is effective where the data shows complex patterns, such as river flow forecasting. LSTM models have demonstrated better stability and reliability in peak flow predictions [8,9,20]. It is designed to overcome traditional neural networks as it can effectively retain information over long sequences. This is achievable using memory cells in LSTM and is extremely useful for tasks where the relationship between data spans a long time [9,20].

Every model has its flaws, and LSTM is no exception it. It struggles when the dataset shows unclear or unstable time patterns. Based on paper [19], LSTM performs poorly compared to models like ANN, as the streamflow data was highly volatile and did not have clear sequential trends [19]. Although LSTM is powerful in capturing long dependencies, it still has a drawback if the training data is limited or highly volatile, as it will start memorising training data instead of finding a general rule or pattern and ultimately leads to overfitting [18,19,20]. In addition, LSTM takes a lot of computational resources as it involves handling multiple control gates and complex memory cell computations; this could be a huge disadvantage for quick model development [7,13,20]. Besides that, the dropout regularisation can also prove to be a flaw. Based on [19,20], it states that the dropout is originally aimed at reducing the overfitting problem, but sometimes it may lead to higher training loss instead of validation loss, as dropout is only active during training. Furthermore, the complexity of the model makes it so that it is less transparent compared to simple models, hence making troubleshooting more difficult for users [20].

2.3 .4 Limitations of Previous Studies on LSTM and ARIMA

Previous research in river flow prediction has predominantly relied on traditional models, particularly the ARIMA model. These models assume a linear relationship between historical and future river flows, which does not adequately capture the complex, nonlinear patterns that characterize real-world hydrological processes [3,5,6,10]. As a result, they often fail to predict extreme events such as sudden floods or sharp flow variations accurately. Furthermore, although techniques like differencing are employed to stabilize nonstationary time series, many studies have struggled to achieve true stationarity, leaving residual trends and variances that compromise forecasting accuracy. This inadequacy limits the reliability of linear models, especially under highly dynamic environmental conditions [11,18,20]. In addition, earlier research showed a notable gap in exploring advanced deep learning model, such as LSTM networks, which are better equipped to model complex temporal dependencies. Limited empirical investigations into LSTM's capabilities left open questions about its potential advantages over traditional methods [7,8,11,18,19]. These limitations highlight the need for more flexible and powerful forecasting techniques that can handle nonlinearities, adapt to nonstationary behaviours, and offer improved predictive performance for effective water resource management.

2.4 Comparative Summary Table 3 existing systems

This shows the comparison of the 3 reviewed systems across key operational dimensions

Aspect	GloFAS	RiverMamba	HydroMax
Purpose	Global early flood and streamflow forecasting	Deep learning-based global flood model	Regional flood warning for the Meuse Basin
Forecasting Method	Physics-based, ensemble LISFLOOD model	AI, state-space deep learning (Mamba blocks)	Parametric model (e.g. Hammerstein-type)
Input Data	ECMWF forecasts, historical reanalyses	ERA5-based meteorological and hydrological data	Real-time gauge and forecasted rainfall
Geographic Coverage	Global (10 km resolution)	Global (5 km resolution)	Regional (Meuse catchment)
Accessibility	Public, via online map and download portal	Research code only	Internal to government (not open-access)
Visualizations	Map viewer, hydrographs, thresholds	Plots generated manually	Alert maps, forecast tables
Usability	Professional forecasters	Research/technical use	Local hydrologists, civil protection officers
Evaluation Metrics	CRPSS, skill scores	RMSE, MAE, NSE	Threshold matching, visual comparison

Table 2.4.1 Comparative Summary Table

Each system offers unique strengths and limitations. GloFAS delivers broad global coverage and physical consistency but may lack the spatial resolution and calibration necessary for Malaysian basins [21],[22],[23]. RiverMamba provides excellent predictive accuracy and adaptability, though it is not yet operational and lacks model interpretability [25], [26]. HydroMax demonstrates long-term reliability and expert integration for its region, yet it is not generalizable beyond its specific deployment [28], [29]. These insights guide the rationale behind building a locally optimized yet generalizable river flow forecasting system using ARIMA and LSTM models.

Chapter 3

System Methodology

3.0 Summary

This project adopts a development-based methodology grounded in the CRISP-DM framework. The goal is to forecast monthly river flow for Sungai Kelantan and Sungai Sokor using two time series models: ARIMA and LSTM. This chapter presents the full system methodology, including model design, data processing, and evaluation protocols.

3.1 Methodological Framework(CRISP-DM)

Phase	**Description**
Business Understanding	To forecast monthly river flow trends for Kelantan rivers using ARIMA and LSTM.
Data Understanding	Raw .txt files from DID formatted by year/month/day with possible missing values.
Data Preparation	Includes parsing, cleaning, resampling daily to monthly means, log transform, and scaling.
Modelling	Trained ARIMA models selected using AIC/BIC; LSTM models trained with sliding windows and optimized using Keras Tuner.
Evaluation	RMSE and MAE on hold-out sets. Forecast plots used for visual inspection.
Deployment	Deployed using Streamlit with model and scaler files loaded at runtime.

3.1.1 System Workflow and Design Diagram

The development follows a linear pipeline beginning with file upload, parsing, and processing, before passing through the forecasting models. The result is visualized and downloadable with Streamlit.

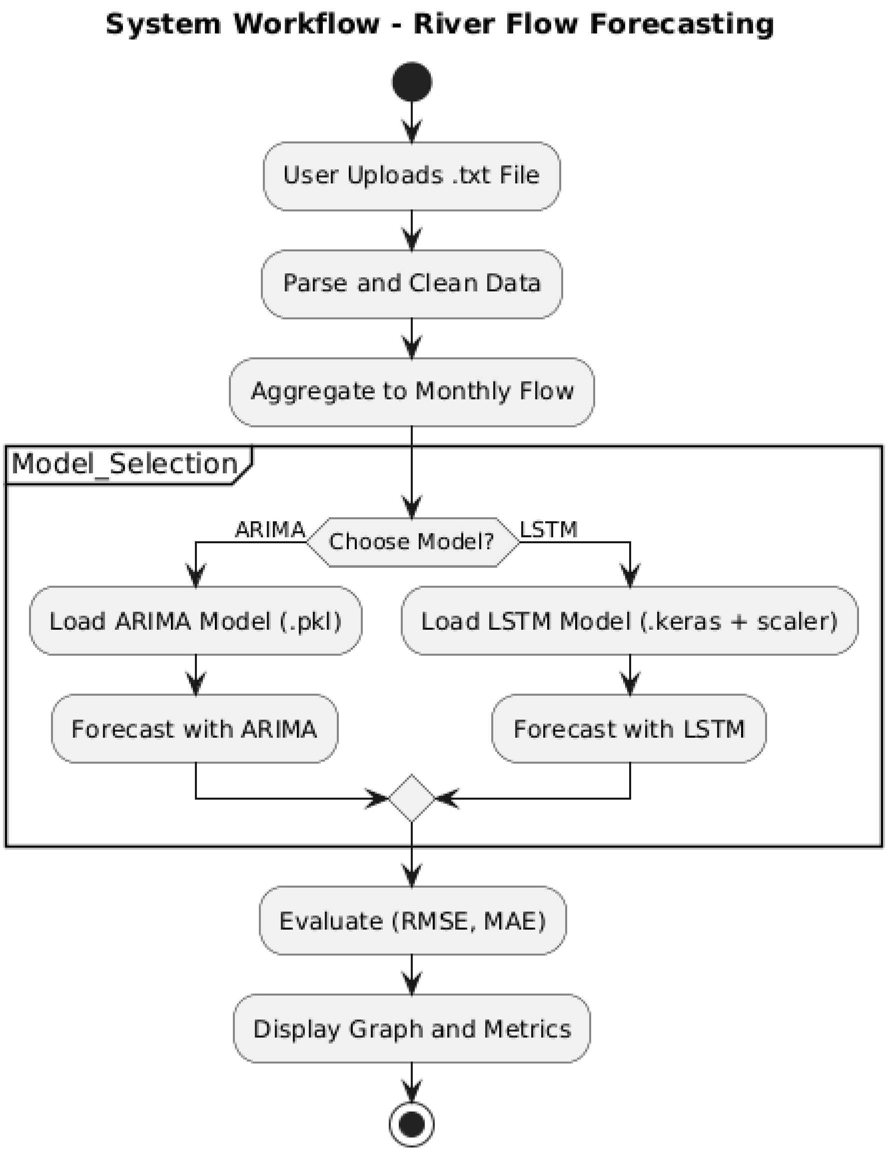


Figure 3.1.1 Flowchart System Design Diagram

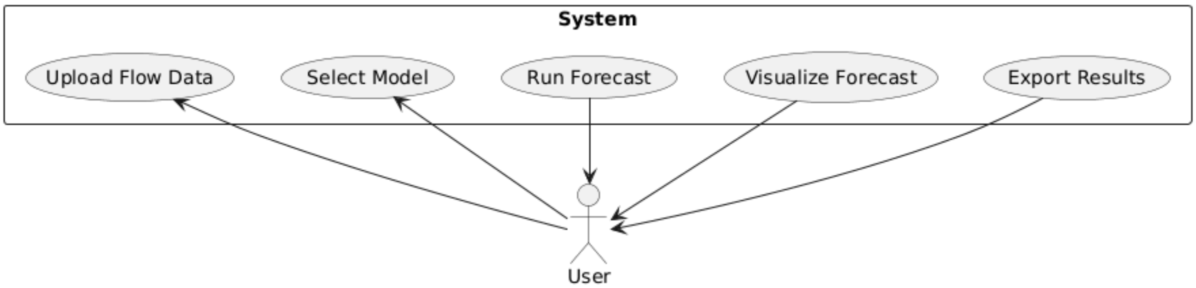


Figure 3.1..2 Use Case Diagram amd Description

The diagram above is a use case diagram that shows how the user interacts with the river flow forecasting system. The user can perform five main actions: uploading flow data, selecting a forecasting model (ARIMA or LSTM), running the forecast, visualizing the results, and exporting the forecast output. These functions represent the key features of the system and are designed to be user-friendly. This diagram helps illustrate the overall workflow of the system, from input to output, and highlights the main tasks a user can perform within the application.

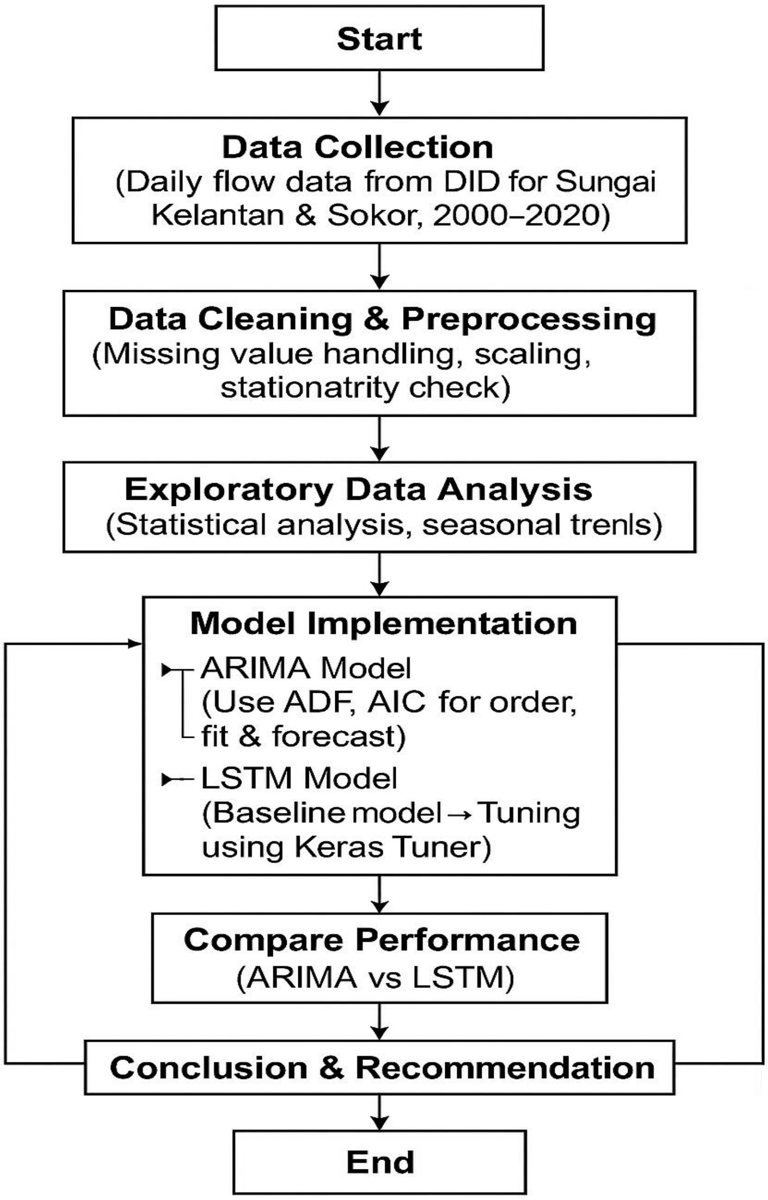


Figure 3.1.3 Process Flow Model LSTM and ARIMA

3.2 Model Design and Equations

3.2.1 ARIMA Model

The ARIMA model has been a traditional approach for time series forecasting, particularly in river flow predictions. It captures linear relationships in data, making it suitable for short-term forecasting tasks. Despite its long-standing use in time series forecasting, particularly for hydrological data, the ARIMA model face difficulties when applied to datasets with nonlinear behaviour and complex temporal structures. First introduced by Box and Jenkins in the 1970s [10], ARIMA is designed to model stationary series but can also be extended to non-stationary data by applying differencing techniques. It combines autoregressive (AR) terms, which leverage past values, with moving average (MA) terms, which utilize past forecasting errors, to model temporal dependencies [11,18]. This model is a type of time series model that uses past values to predict future values. The formula for an AR(p) model is:

$$y_t = m + \sum_{i=1}^{p} \varphi i \mathrm{y} \mathrm{t} - \mathrm{i} + \epsilon t$$

Notation of AR(p) indicates that the model uses 'p' previous values(lags) of the variable.

- y_t =the current value of the variable we are trying to predict.
- m is a constant that helps adjust the model.
- φ_i are coefficients that measure how much influence the past values (lags) have on the current value. Each coefficient corresponds to a specific lag.
- y_{t-i} represents the past values of the variable, where 'i' ranges from 1 to 'p'.

- ϵ_t is a error term, which accounts for the noise in the data.

Formula MA model of order (q) , is

$$y_t = \mu + \sum_{i=0}^{q} \omega \mathrm{i}\, \epsilon \mathrm{t} - \mathrm{I}$$

Notation of MA(q) indicates that the model uses 'q' previous error terms.

- $\mu = y_t$, , μ=0

- ω_i =weights that determine how much influence the past error terms have on the current value when $\omega_0 = 1$.

- ϵ_{t-I} represent past error terms where 'i' ranges from 0 to 'q' term.

3.2.2 LSTM Model

Long Short-Term Memory is an advanced architecture derived from RNN and is used to forecast and learn time-series data.[12]. This model is a powerful alternative to traditional methods such as ARIMA. LSTM is designed to learn and predict data by retaining memory of past values. There has been research that proven that LSTM outperforms ARIMA in terms of the prediction of river flow changes compared to ARIMA [13,11,18]. Its ability to store memory and learn from past data makes it extremely powerful when handling nonlinear and time series data. There has been research that shows LSTM models have demonstrated greater consistency and robustness in predicting extreme flow values and are especially suitable for low-volume river systems where subtle changes are critical [9]. The LSTM architecture includes mechanisms such as a forget gate, input gate, output gate, and memory cell, which work together to selectively retain or discard information across time steps [14]. These gating operations help LSTM networks manage long-term dependencies and reduce vanishing gradient issues during training. While the specific mathematical operations are standard across implementations [15,16,18], the combination of these components allows LSTM to adapt effectively to complex temporal dynamics

Forget Gate: $Ft = \sigma(Wf \cdot [ht-1, xt] + bf)$

Input Gate: $It = \sigma(Wi \cdot [ht-1, xt] + bi)$

Candidate Memory: $\tilde{C}t = tanh(Wc \cdot [ht-1, xt] + bc)$

Cell State: $Ct = Ft * Ct-1 + It * \tilde{C}t$

Output Gate: $O_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$

Hidden State: $h_t = O_t (x) \tanh(C_t)$

In the Long Short-Term Memory (LSTM) model, memory and information flow are managed through a set of gates that operate at each time step to maintain long-term dependencies in sequential data [18]. The process begins with the forget gate, which evaluates the previous hidden state and the current input to determine the proportion of past memory to retain. This is achieved using a sigmoid activation function that outputs values between 0 and 1—values near 0 indicate forgetting, while those near 1 suggest retention [18]. The input gate determines how much of the new information should be stored in the memory. Like the forget gate, it uses the sigmoid function. In parallel, a candidate memory vector is created using the tanh activation function, which normalizes the input between -1 and 1 to ensure stability in the update process [18]. Next, the cell state is updated by combining the previous memory, scaled by the forget gate, and the candidate memory, weighted by the input gate. This step integrates both old and new information in a balanced manner [18]. The output gate controls what portion of the updated memory should be visible to the next hidden state. A sigmoid function determines the relevance of the memory for the current output, and the tanh function ensures the output values are scaled between -1 and 1. The final hidden state is obtained by applying tanh to the updated cell state and multiplying it by the output gate's activation [18].

Below is the image of the basic architecture of LSTM models.[18].

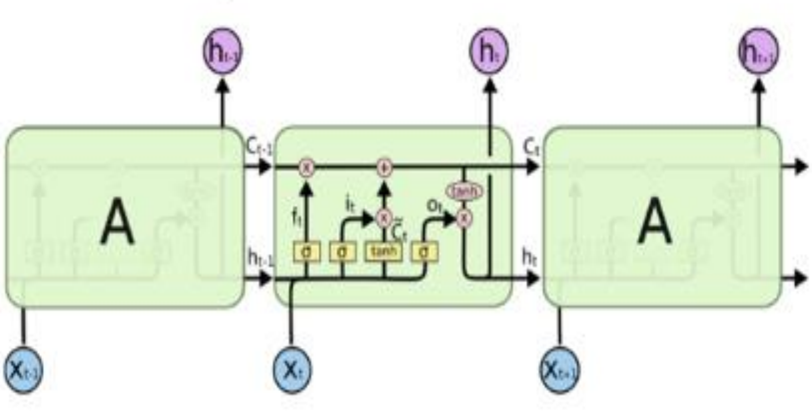


Figure 3.2.2 Basic Architecture of LSTM Model

Chapter 4

System Design

4.1 System Components Specifications

This section describes the specifications of hardware, software, and libraries used to develop and run the forecasting system.

Programming Language: Python 3.10+

Development Environment: Google Colab

Deployment Framework: Streamlit

Libraries for requirements .txt

streamlit==1.35.0

pandas==2.2.2

numpy==1.24.3

matplotlib==3.7.1

seaborn==0.12.2

openpyxl==3.1.2

scikit-learn==1.2.2

pmdarima==2.0.4

statsmodels==0.14.1

tensorflow==2.13.0

keras-tuner==1.4.6

joblib==1.4.2

The system was tested on cloud-hosted environments such as Google Colab and deployed via Streamlit Cloud. Model files (.pkl, .keras) and scalers were pre-trained and loaded at runtime. Model Training and testing were conducted using Google Colab. For the research and implementation of this project, a personal computer with the following specifications was utilized. The system used was an ASUS TUF Gaming A15 (model FA506ICB) laptop, equipped with an AMD Ryzen 5 4600H processor that includes integrated Radeon Graphics and operates with 6 cores and 12 threads at a base clock speed of 3.0 GHz. The device runs on the Windows 11 operating system, ensuring compatibility with modern development tools and libraries. In terms of graphics capability, it features an NVIDIA GeForce GT 930MX GPU with 2GB of DDR3 video memory, sufficient for basic visualization and model training processes. The machine is supported by 8GB of DDR4 RAM, which facilitates moderate multitasking and data processing tasks. For storage, a 600GB SATA HDD was available, providing insufficient space for storing raw datasets, model outputs, and project files. The system BIOS is version FA506ICB.303, dated 23rd February 2022, which ensures compatibility with required hardware-level functionalities throughout the project execution phase.

4.2 Component Design and Data Pipeline

4.2.1 Data Collection and Preprocessing

River streamflow datasets for Sungai Kelantan and Sungai Sokor were acquired from the Department of Irrigation and Drainage (DID), Malaysia. The datasets originally span from 1960 to 2020 (Kelantan) and 1995 to 2020 (Sokor). Daily values were collected and later aggregated into monthly values for forecasting. Only data between January 2000 and May 2020 was used for model development to ensure number of datasets alligned. . Stationary testing is carried out by applying ADF test to determine the order of differencing needed for ARIMA while LSTM input data was scaled using MinMaxScaler to a range between 0 and 1. LSTM series was transformed into supervised format using 12-month time windows.Log transformation is done for ARIMA during preprocessing phase.

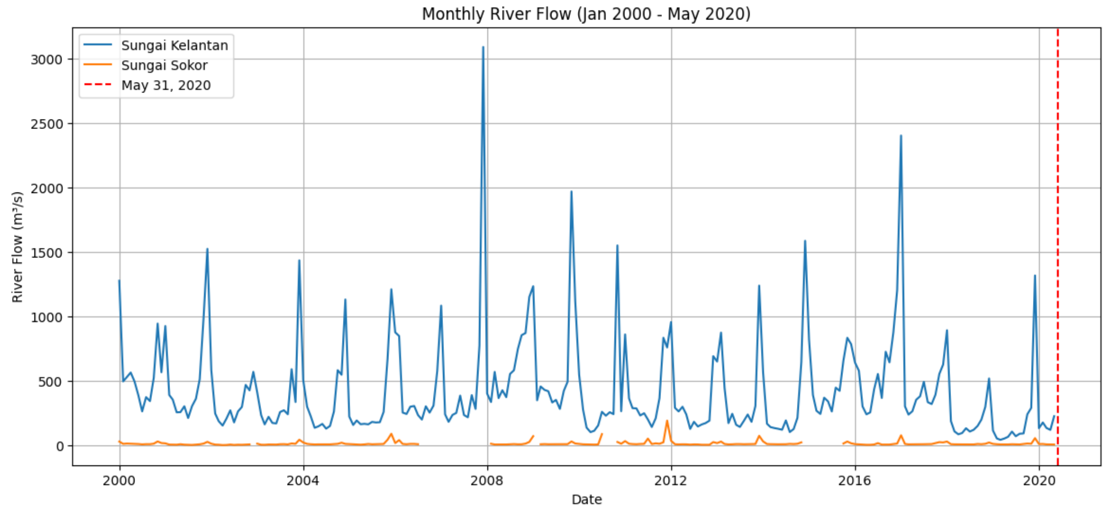


Figure 4.2.2 Exploratory Data Analysis

EDA was conducted to understand the characteristics of river flow. Summary statistics such as mean, skewness, and kurtosis were calculated. Boxplots and line plots were used to detect seasonality and variability.

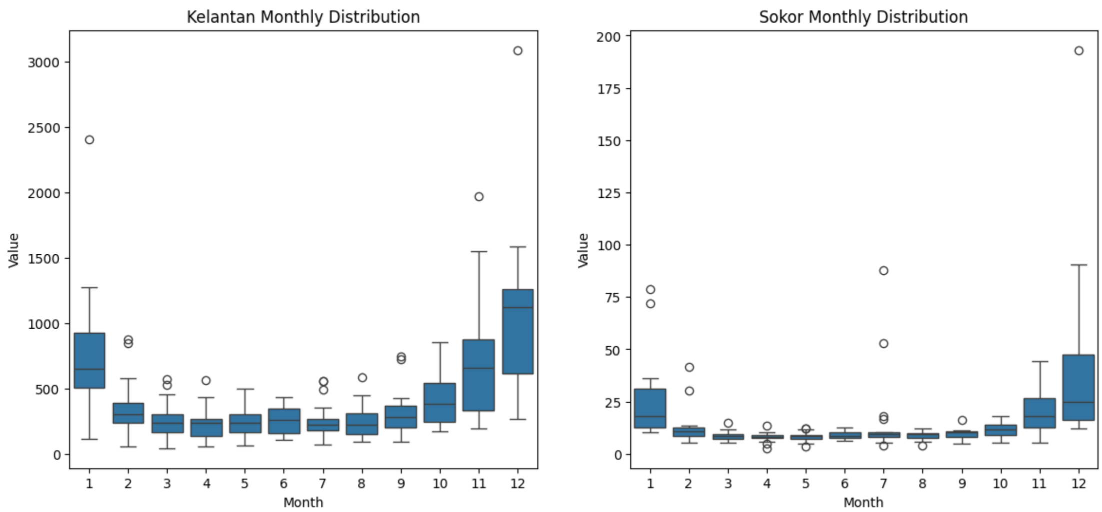


Figure 4.2.3. Boxplot for both Sungai Kelantan and Sungai Sokor

Based on Figure 4.3.2a, there are extreme outliers in months like December and January (>3000 m³/s), suggesting flood events in Sungai Kelantan. Sungai Sokor has a similar seasonal pattern to Sungai Kelantan, but on a much smaller scale. Overall flow values are significantly lower, with median values below 20m³/s in most of the months. To conclude, the boxplots reveal strong monthly seasonality in both rivers, with pronounced wet season peaks in November–January. Kelantan shows much greater flow and variability than Sokor.

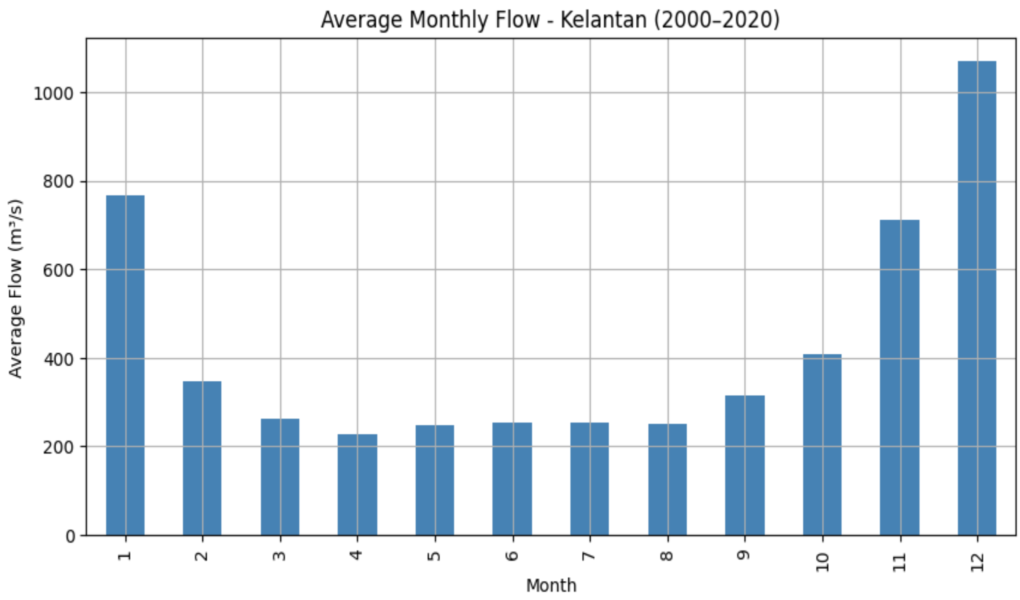


Figure 4.2.4. Histogram Kelantan

From Figure 4.3.2b, Sungai Kelantan, we can observe the highest flows in December and January. There is a secondary peak in November. Flow is the lowest from April to July with consistent low values. This pattern reflects the Northeast Monsoon, which normally peaks from November to January. The flow pattern in Kelantan shows strong seasonal variation, with monsoon-driven peaks and long dry periods.

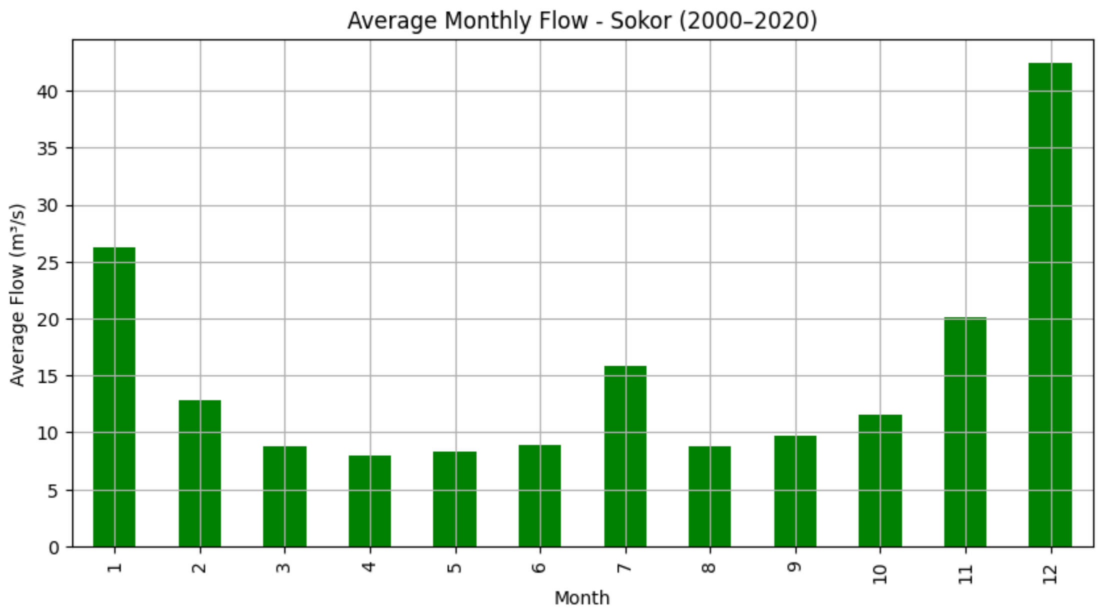


Figure 4.3.2c. Histogram Sokor

From Figure 3.1.3c, the highest peak flow is again in December. Like Sungai Kelantan, the lowest flow occurs from March to September. Furthermore, we could also observe that overall discharge levels are much lower, with average flows below 10 m^3/s for most months. Sungai Sokor mirrors the same monsoon-influenced seasonality, though at a smaller scale. The consistency in pattern supports the use of similar model structures, though with less complexity due to lower variance.

4.4 System Design

The river flow forecasting system integrates several key components that operate in a sequential and modular pipeline. The process begins with data preprocessing, where uploaded .txt files are parsed and converted from daily to monthly flow values. Missing data is handled through linear interpolation to ensure consistency.

Once preprocessed, the user selects a forecasting model—ARIMA or LSTM—via the Streamlit interface. The system then routes the data to the appropriate backend pipeline. For ARIMA, the model applies differencing and uses parameters selected via auto-tuning. For LSTM, the data is scaled and transformed into supervised sequences before prediction.

The output from each model is passed to a visualization layer that plots both historical and forecasted river flows. Metrics such as RMSE and MAE are also displayed for evaluation. All components are coordinated through app.py, which manages user input, model invocation, and result rendering. This integrated workflow ensures a smooth transition from data upload to result interpretation.

User Input → Preprocessing → Model Selection → Forecasting → Visualization.

Chapter 5

System Evaluation

5.0 Overview

This chapter describes the practical implementation of the river flow forecasting system, covering the development of backend scripts, the design of the user interface, and the structure of the application environment. The system integrates multiple Python modules for data transformation and modelling, alongside a Streamlit-based frontend to enable real-time user interaction.

5.1 Frontend Interface

The Kelantan River Forecasting System was implemented using Streamlit to allow users to interactively generate river flow forecasts for Sungai Kelantan and Sungai Sokor. The interface is clean, responsive, and suitable for technical and non-technical users alike.
The homepage contains a title and introductory markdown , a dropdown to select the river station (Sungai Kelantan or Sungai Sokor),a radio button to choose the model (ARIMA or LSTM) and a forecast button to trigger backend processing as shown in 5.1.1 figure.

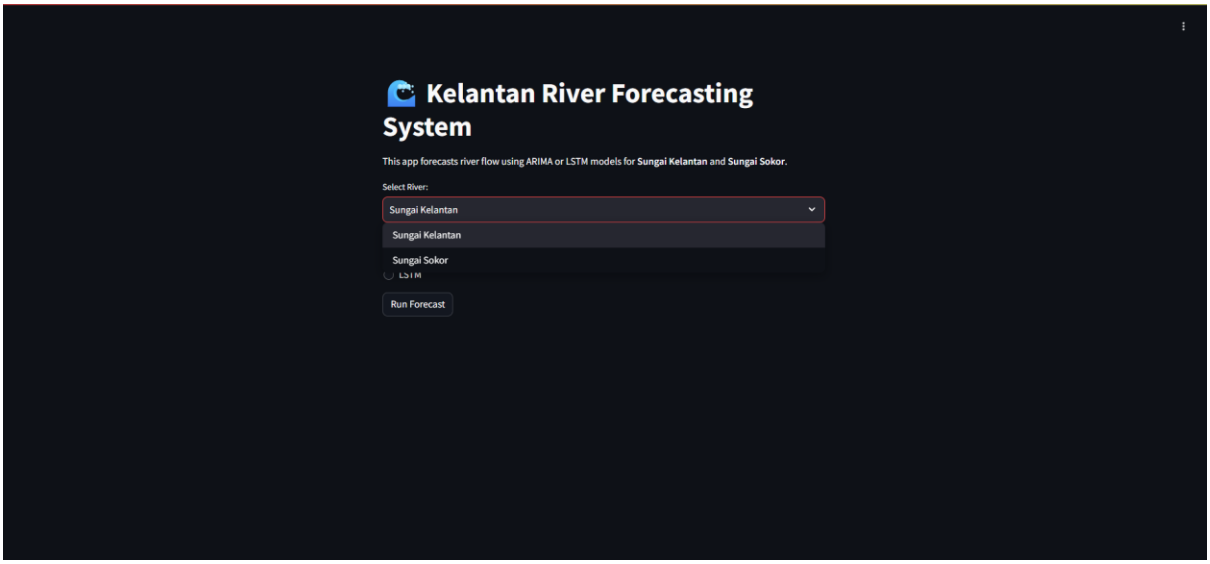


Figure 5.1.1 Input interface

A graph that displays the **last 30 days of actual data** and the **next 30 days of predicted values will be shown after clicking run forecast.**

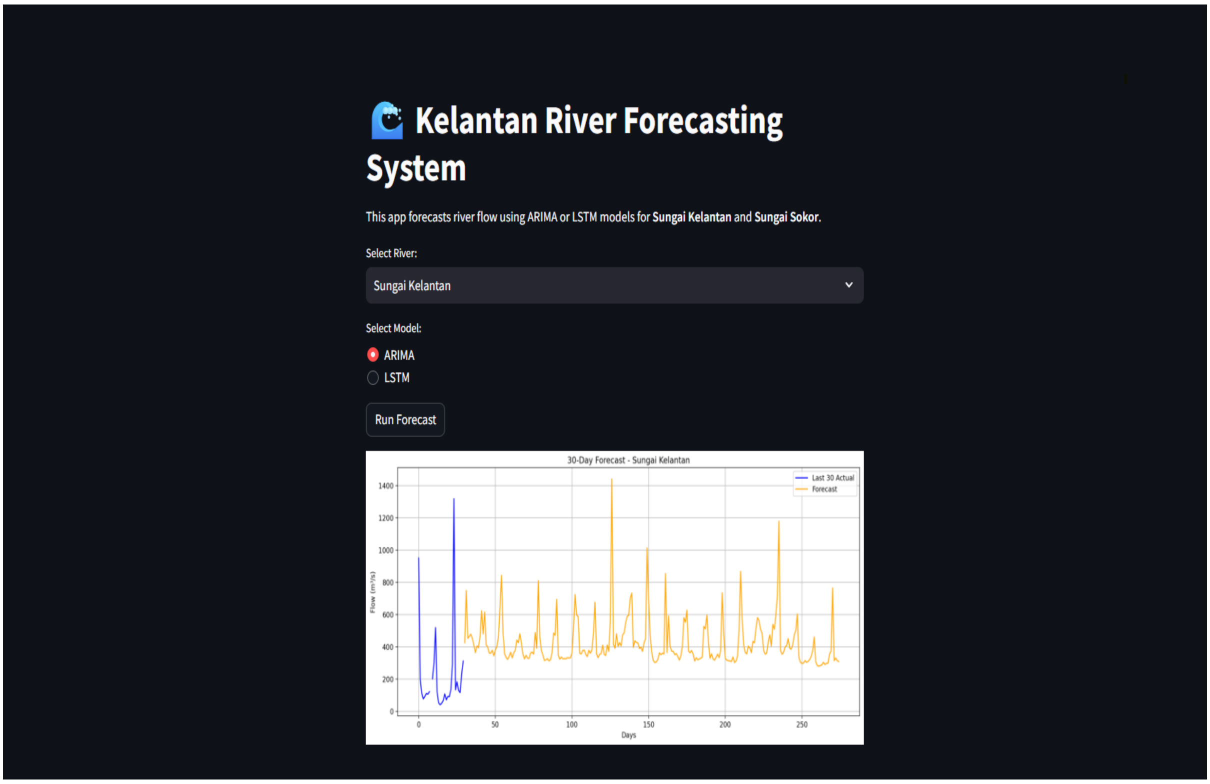


Figure 5,1,2 Sungai Kelantan 30 days future forecast using ARIMA

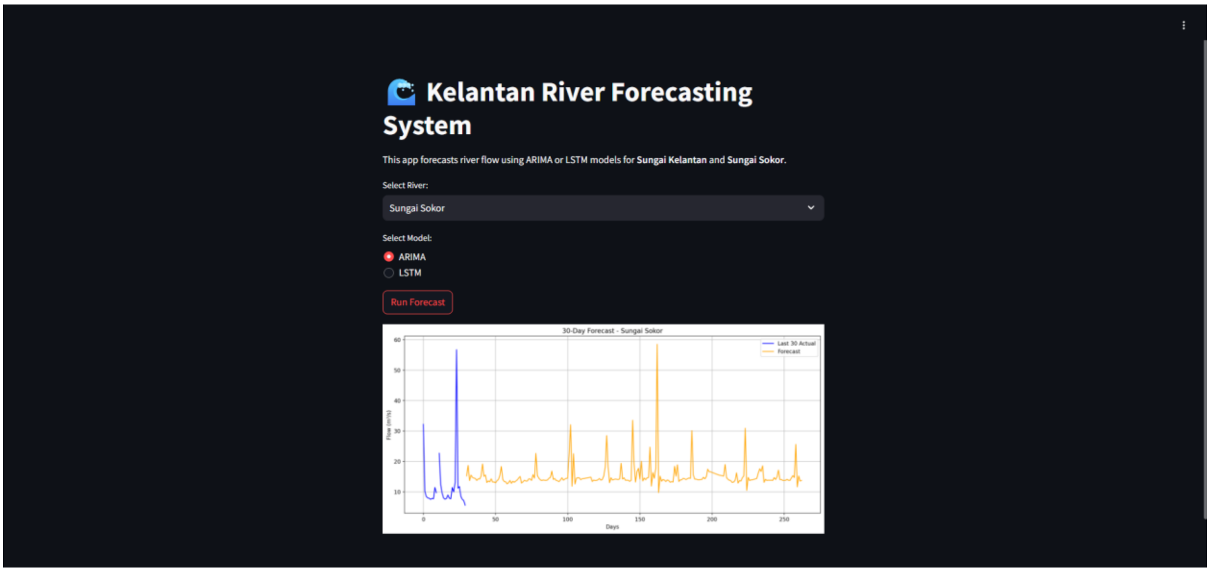


Figure 5.1.3 Sungai Sokor 30 days forecast using ARIMA

5.2 Code Structure summary(app.py)

```python
def load_data(river):
        lines = f.readlines()

    data = []
    current_year = None
    for line in lines:
        line = line.strip()
        if line.startswith('Daily means') and 'Year' in line:
            current_year = int(line.split('Year')[1].split()[0])
        elif line and line[0].isdigit():
            parts = line.split()
            if len(parts) >= 13 and current_year:
                day = int(parts[0])
                for i, value in enumerate(parts[1:13]):
                    try:
                        val = float(value)
                        date = pd.Timestamp(year=current_year, month=i+1, day=day)
                        data.append((date, val))
                    except:
                        continue
    df = pd.DataFrame(data, columns=['Date', 'Flow'])
    df.set_index('Date', inplace=True)
    df = df.resample('M').mean()
    df['Flow_log'] = np.log(df['Flow'] + 1)
    return df
```

Figure 5.2.0 Load data functions

5.2.1 ARIMA Forecasting

```python
# === Forecast Functions ===
def forecast_arima(df, river_name, steps=30):
    model = joblib.load(f"models/arima_{river_name.lower().replace(' ', '_')}.pkl")
    forecast = model.predict(n_periods=steps)
    last_30 = df['Flow'].values[-30:]
    return last_30, forecast
```

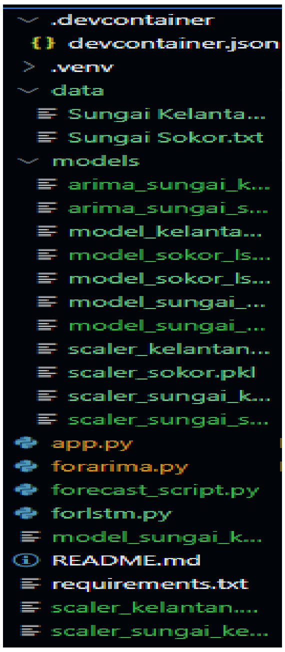
Figure 5.2.1 ARIMA Forecasting code

5.2.2 LSTM Forecasting

```python
def forecast_lstm(df, river_name, steps=30):
    scaler = joblib.load(f"models/scaler_{river_name.lower().replace(' ', '_')}.pkl")
    model = load_model(f"models/model_{river_name.lower().replace(' ', '_')}_lstm.keras")

    input_seq = df['Flow_log'].values[-30:]
    input_scaled = scaler.transform(input_seq.reshape(-1,1)).flatten().tolist()
    forecast_scaled = []

    for _ in range(steps):
        x_input = np.array(input_scaled[-30:]).reshape(1, 30, 1)
        yhat = model.predict(x_input, verbose=0)[0][0]
        forecast_scaled.append(yhat)
        input_scaled.append(yhat)

    forecast_unscaled = scaler.inverse_transform(np.array(forecast_scaled).reshape(-1,1)).flatten()
    forecast = np.exp(forecast_unscaled) - 1
    last_30 = np.exp(input_seq) - 1
    return last_30, forecast
```

Figure 5.2.2 LSTM Forecasting code

5.3 File structure and Workflow



The forecasting system is organized using a modular file structure that separates core functionalities into manageable components. At the root level, key Python scripts such as app.py, dataconvert.py, arima.py, and lstm.py serve specific purposes. The app.py script acts as the main entry point, orchestrating the system's logic and user interface through Streamlit. The dataconvert.py module is responsible for reading, cleaning, and aggregating raw river flow data from .txt files. The arima.py and lstm.py modules contain model-specific pipelines that handle training, evaluation, and forecasting tasks.

A dedicated models/ directory stores the serialized ARIMA and LSTM models along with their respective scalers. This modular storage makes it easier to load the correct model based on user selections in the interface. Additional files, such as requirements.txt, ensure environment reproducibility by listing all necessary Python libraries required to run the application.

This organized structure supports workflow clarity and makes the system maintainable, scalable, and easier to deploy. Each component is loosely coupled but tightly integrated, allowing future upgrades or replacements of individual parts without affecting the overall pipeline.

Chapter 6 Model Performance

6.0 Summarise

This chapter evaluates the performance and effectiveness of the developed river flow forecasting system. Both ARIMA and LSTM models were assessed across two different rivers, Sungai Kelantan and Sungai Sokor, using Root Mean Square Error (RMSE), Mean Absolute Error (MAE).. Beyond numerical results, this chapter reflects on the challenges faced during development, assesses how well the project objectives were achieved, and concludes with key insights gained throughout the process.

6.1 Model Performance Comparison

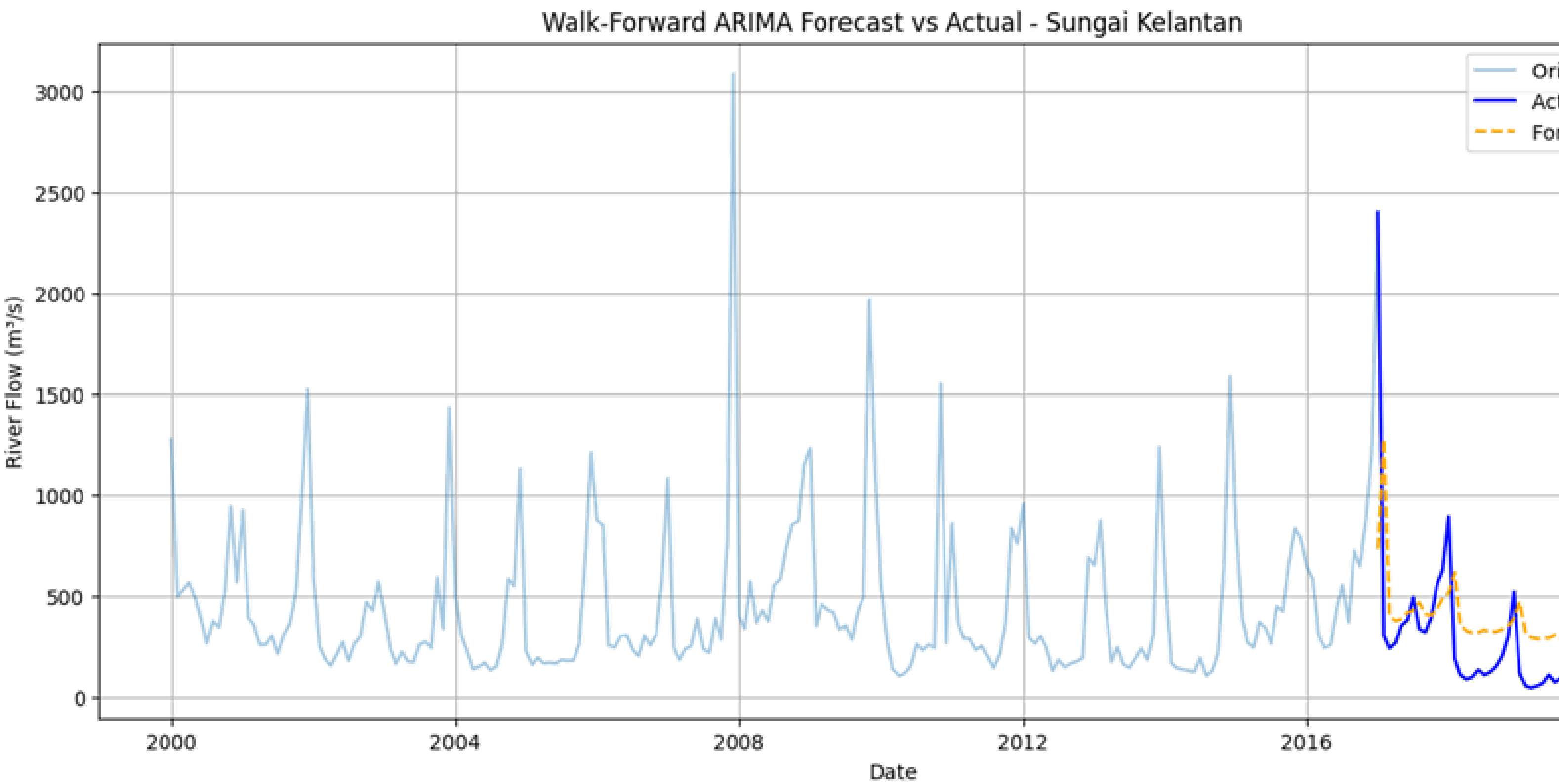


Figure 6.1.1 Walk-Forward ARIMA Forecast Vs Actual (Kelantan)

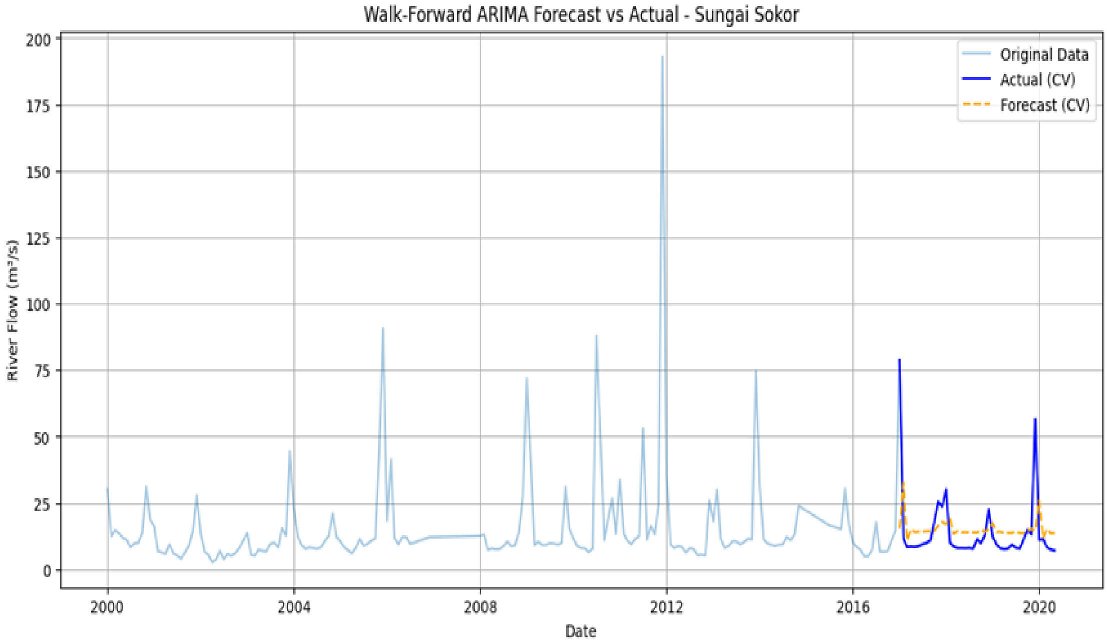


Figure 6.1.2 Walk-Forward ARIMA Forecast Vs Actual (SOKOR)

```
LSTM Evaluation - Sungai Kelantan
RMSE: 367.9571
MAE: 86.1908
R² Score: 0.5308

Classification Metrics (Threshold: 749.08 m³/s)
Accuracy: 0.9741
Recall: 0.8485
Precision: 0.8889
F1-Score: 0.8682
Confusion Matrix:
[[1167   14]
 [  20  112]]
```

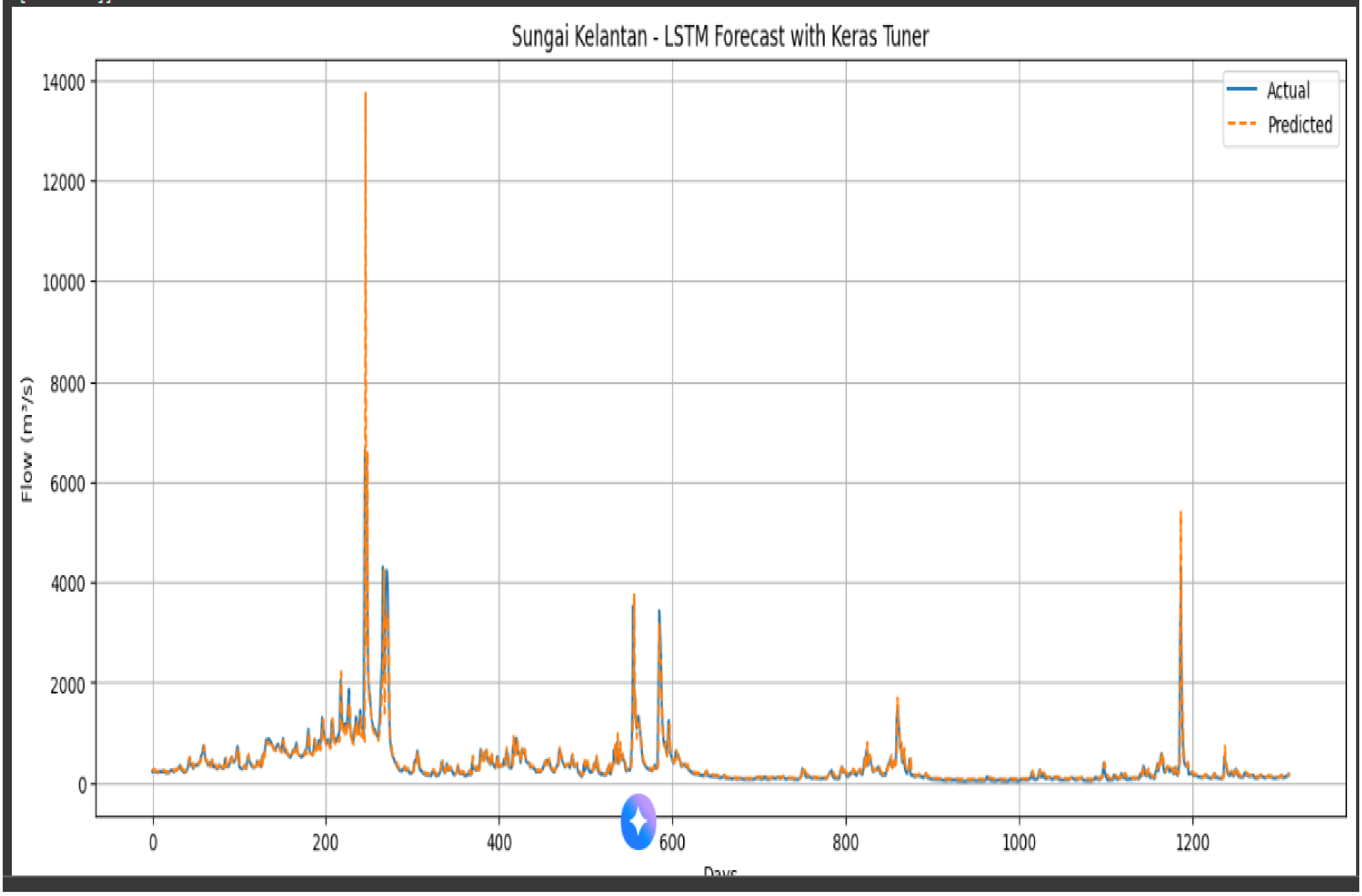


Figure 6.1.3 LSTM Forecast Vs Actual (Kelantan)

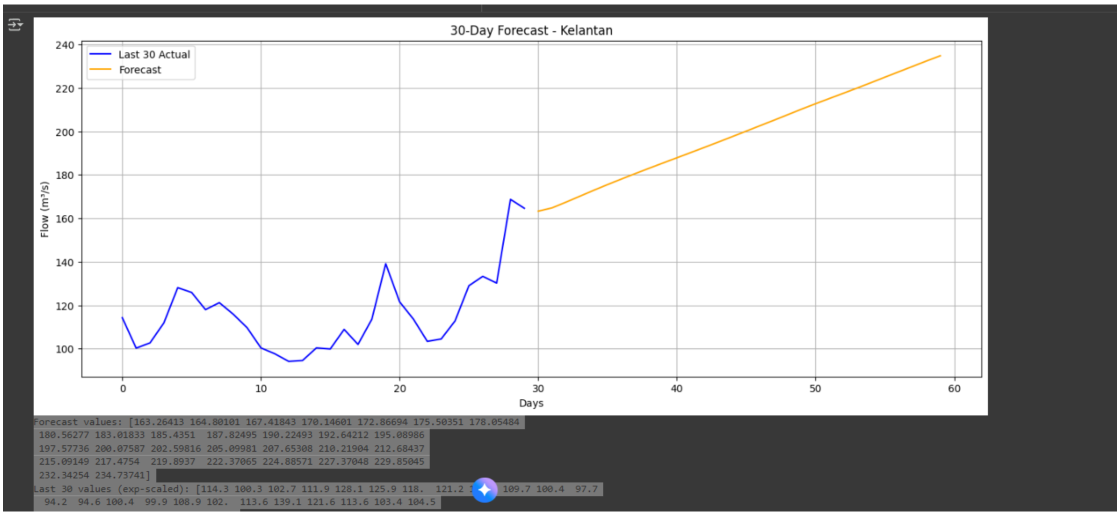


Figure 6.1.4 30-Day-Future-Forecast (Kelantan)

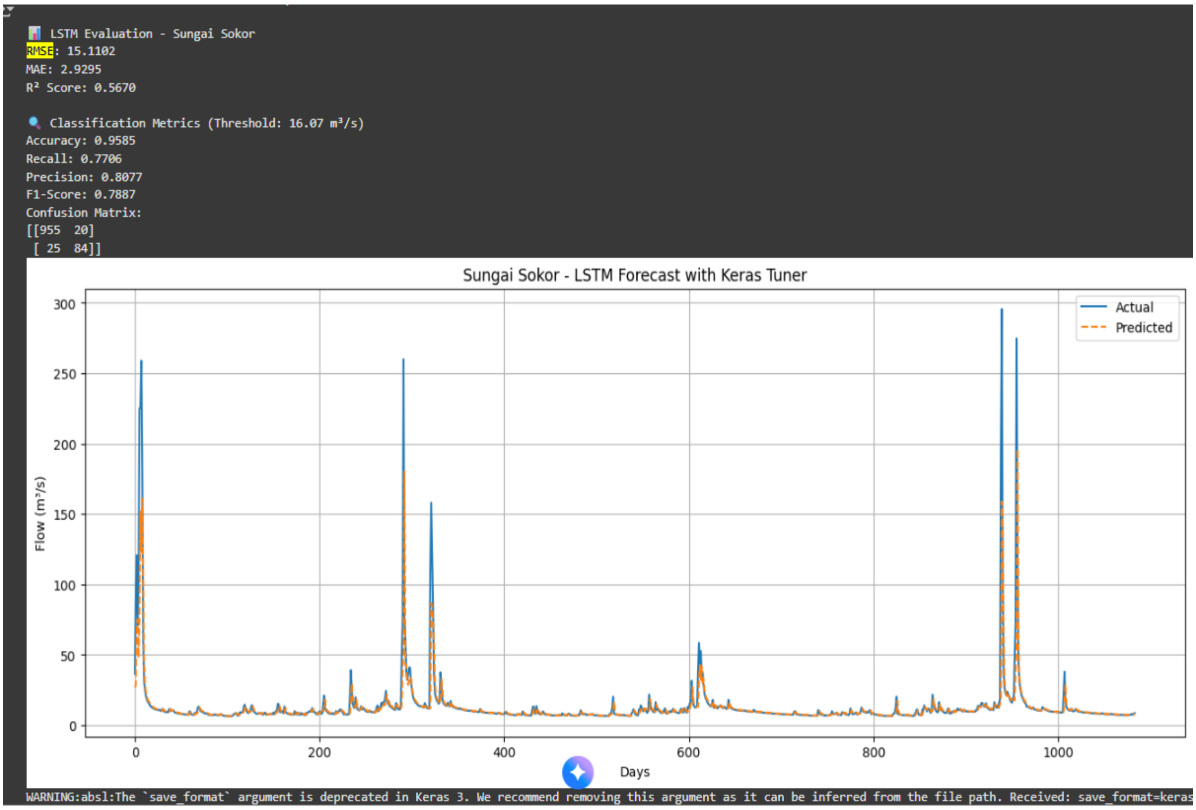


Figure 6.1.5 LSTM Forecast Vs Actual (SOKOR)

Table 6.1.6: Model Comparison Metrics for Both Rivers

Model	River	RMSE	MAE	R^2	Accuracy	Precision	F1-score
ARIMA	Kelantan	397.51	259.04	-	-	-	-
ARIMA	Sokor	13.69	8.17	-	-	-	-
LSTM	Kelantan	367.96	86.19	0.5308	0.9741	0.8889	0.8682
LSTM	Sokor	15.11	2.93	0.5670	0.9585	0.8077	0.7887

6.2 Classification Evaluation for Flood Detection

In addition to traditional regression metrics such as RMSE and MAE, this project also employed classification metrics to evaluate the model’s ability to detect critical flow events, particularly flood risks. This was done by introducing threshold-based classification, where values exceeding predefined thresholds (749.08 m^3/s for Sungai Kelantan and 16.07 m^3/s for Sungai Sokor) were labeled as potential flood events. For Sungai Kelantan, the LSTM model achieved high accuracy (97.41%) alongside a precision of 88.89% and recall of 84.85%. These values indicate that the model not only avoided frequent false alarms but was also sensitive enough to detect actual flood occurrences with minimal false negatives. The F1-score of 0.8682 further confirms a well-balanced performance between precision and recall. The confusion matrix also revealed that only 20 flood events were missed out of a significantly large dataset, demonstrating the model’s effectiveness in supporting early warning systems in high-risk flood regions.

For Sungai Sokor, although the river exhibits much lower overall discharge and fewer extreme events, the classification approach still yielded meaningful insights. The LSTM model achieved a respectable accuracy of 95.85%, with a precision of 80.77% and recall of 77.06%. These slightly lower recall values suggest that while the model is reliable in general prediction, there is still room for improving sensitivity in capturing low-volume flood events. The F1-score of 0.7887 reflects a moderately balanced performance, particularly suitable for a river system with less variability. Overall, the classification evaluation added a practical perspective to the model’s predictive power, confirming that the LSTM approach is not only suitable for continuous forecasting but also for identifying risk thresholds essential for flood management applications.

6.3 Project Challenges

Throughout the project, several challenges emerged during both the modelling and development stages. One of the most critical issues was dealing with missing data in the daily streamflow datasets, especially from Sungai Sokor, which required careful interpolation and filtering to ensure reliability. Another challenge was handling the non-stationarity and extreme seasonality patterns present in the Kelantan river system, which often caused instability in ARIMA model fitting. Additionally, integrating LSTM models required substantial preprocessing and tuning, including sequence reshaping, scaling, and hyperparameter tuning, sometimes even the platform I am running will crashed. On the deployment side, converting Colab-based experiments into a fully functional and user-friendly Streamlit application posed extremely technical and interface design challenges.Moreover, there is also the collision of version in the Google Colab between 2 Machine learning model such as ARIMA and LSTM. These 2 cannot be run together in a same file as function required by ARIMA (pdmarima) requires installing version 1.24.3 for numpy and this version of numpy cannot run LSTM.

6.4 Objective Evaluation

This project successfully addressed its primary objective, which was to develop a river flow forecasting system for Kelantan rivers using both ARIMA and LSTM models. A functional prototype was implemented using Streamlit, providing users with an interactive interface to select rivers, choose forecasting models, and visualize prediction outputs. The second objective, which focused on evaluating and comparing the forecasting performance of ARIMA and LSTM, was achieved through the use of standard statistical accuracy metrics such as RMSE (Root Mean Square Error) and MAE (Mean Absolute Error). The results showed that the LSTM model consistently outperformed ARIMA in rivers with high flow variability, such as Sungai Kelantan, while ARIMA remained competitive in low-flow environments like Sungai Sokor. Lastly, the third objective—studying, preprocessing, and transforming data from the two rivers—was thoroughly executed. The raw daily data were cleaned, missing values interpolated, and the time series aggregated into monthly averages. Appropriate scaling and windowing techniques were also applied to make the data suitable for LSTM input, while stationarity tests guided ARIMA modeling. Overall, each objective set at the beginning of the project was addressed systematically and aligned well with the results and findings obtained throughout the development process.

6.5 Concluding Remark

In conclusion, the project is not successfully developed as it cannot load LSTM models due to differences in versions for TensorFlow. TensorFlow 2.2.2 is needed, even after re-running my code file; Streamlit still cannot run it. Evaluation metrics still show that LSTM outperforms ARIMA in detecting flood-related threshold breaches, particularly for Sungai Kelantan or even rivers that have high river flows. This LSTM model might not work well with region that have low water flow. For example, Sungai Sokor. Meanwhile, ARIMA remained a competitive and lightweight model for rivers with less variability, such as Sungai Sokor.

Chapter 7 Conclusion and Recommendations

7.1 Conclusion

This project aimed to develop a river flow forecasting system using both ARIMA and LSTM models for two rivers in Kelantan: Sungai Kelantan and Sungai Sokor. While most parts of the system were successfully implemented, one major limitation affected the final deployment. The LSTM model could not be fully loaded in the Streamlit application due to compatibility issues with TensorFlow. Specifically, the deployed environment required version 2.2.2, and even after adjustments, the application failed to support the model in real time. Despite this setback, the forecasting models were properly trained, tested, and evaluated using historical river flow data. The results show that the LSTM model generally performs better in rivers with high variability and complex flow patterns, such as Sungai Kelantan. It was more accurate in detecting flood-level flows, as shown by the evaluation metrics. However, for smaller rivers like Sungai Sokor, where the data is less dynamic, ARIMA proved to be a simpler and more effective model. Although the LSTM model could not be fully deployed, the project still achieved important outcomes, such as a working interface, a preprocessing pipeline, and complete performance analysis. The findings also provide useful guidance on model selection depending on river characteristics. This experience highlights the importance of both technical accuracy and deployment feasibility in real-world forecasting systems.

7.2 Recommendations

There are several ways to improve this project in the future. First, it would be helpful to include more environmental data, such as rainfall or temperature, as this might improve the accuracy of the predictions. Second, the system could be extended to support more rivers across Malaysia. This would make the tool more useful on a national level

Another improvement would be to link the system to live data using APIs, so forecasts can be updated automatically. This would be especially useful for early warning systems. More advanced models, such as GRU or Transformer-based networks, could also be explored to improve prediction accuracy. Finally, testing the system with real users, such as government agencies or local communities, could help improve the design and make the system easier to use for non-technical users.

REFERENCES

[1] M. Abdul Majid, M. Hafidz Omar, M. S. M. Noorani, and F. Abdul Razak, "River-flood forecasting methods: the context of the Kelantan River in Malaysia," *IOP Conference Series: Earth and Environmental Science*, vol. 880, no. 1, p. 012021, Oct. 2021, doi: https://doi.org/10.1088/1755-1315/880/1/012021.

[2] Z. M. Yaseen, A. El-shafie, O. Jaafar, H. A. Afan, and K. N. Sayl, "Artificial intelligence-based models for stream-flow forecasting: 2000–2015," *Journal of Hydrology*, vol. 530, pp. 829–844, Nov. 2015, doi: https://doi.org/10.1016/j.jhydrol.2015.10.038.

[3] K. Albeladi, B. Zafar, and A. Mueen," Time Series Forecasting using LSTM and ARIMA", International Journal of Advanced Computer Science and Applications, vol. 14, no. 1, pp. 313–320, 2023

[4] Z. Xie, Q. Liu, and Y. Cao, "Hybrid deep learning modeling for water level prediction in Yangtze River", Intelligent Automation and Soft Computing, vol. 28, no. 1, pp. 153–166, 2021. https://doi.org/10.32604/iasc.2021.016246

[5] R. Zhang, H. Song, Q. Chen, Y. Wang, S. Wang, and Y. Li, "Comparison of ARIMA and LSTM for prediction of hemorrhagic fever at different time scales in China", PLoS ONE, vol. 17, no. 1, January 2022. https://doi.org/10.1371/journal.pone.0262009.

[6] Y. Liu, H. Wang, W. Feng, and H. Huang, "Short term real-time rolling forecast of urban river water levels based on LSTM: A case study in Fuzhou city, China", International Journal of Environmental Research and Public Health, vol. 18, no. 17, 2021.

[7] M. A. Al Mehedi, M. Khosravi, M. M. S. Yazdan, and H. Shabanian, "Exploring Temporal Dynamics of River Discharge Using Univariate Long Short-Term Memory (LSTM) Recurrent Neural Network at East Branch of Delaware River", Hydrology, vol. 9, no. 11, 2022. https://doi.org/10.3390/hydrology9110202

[8] H. Chu, Z. Wang, and C. Nie, "Monthly Streamflow Prediction of the Source Region of the Yellow River Based on Long Short-Term Memory Considering Different Lagged Months", Water (Switzerland), vol. 16, no. 4, 2024.

[9] T. Kim, T. Yang, S. Gao, L. Zhang, Z. Ding, X. Wen, J. J. Gourley, and Y. Hong, "Can artificial intelligence and data-driven machine learning models match or even replace process-driven hydrologic models for streamflow simulation?: A case study of four watersheds with different hydro-climatic regions across the CONUS", Journal of Hydrology, vol. 598, September 2020, article no. 126423. https://doi.org/10.1016/j.jhydrol.2021.126423.

[10] Z. Wang and Y. Lou, ”Hydrological time series forecast model based on wavelet de-noising and ARIMA-LSTM”, in Proceedings of 2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), 2019, pp. 1697–1701. https://doi.org/10.1109/ITNEC.2019.8729441

[11] S. Siami-Namini, N. Tavakoli, and A. Siami Namin, ”A Comparison of ARIMA and LSTM in Forecasting Time Series,” in Proceedings of the 17th IEEE International Conference on Machine Learning and Applications (ICMLA), April 2019, pp. 1394–1401. https://doi.org/10.1109/ICMLA.2018.00227.
[12] H. Chu, Z. Wang, and C. Nie, ”Monthly Streamflow Prediction of the Source Region of the Yellow River Based on Long Short-Term Memory Considering Different Lagged Months,” Water (Switzerland), vol. 16, no. 4, 2024. https://doi.org/10.3390/w16040593

[13] Y. Sudriani, I. Ridwansyah, and H. A. Rustini, ”Long short term memory (LSTM) recurrent neural network (RNN) for discharge level prediction and forecast in Cimandiri river, Indonesia”, IOP Conference Series: Earth and Environmental Science, vol. 299, no. 1, 2019. https://doi.org/10.1088/1755-1315/299/1/012037.

[14] H.B. Chu, J. Wu, W.Y. Wu, and J.H. Wei, ”A dynamic classificationbased long short-term memory network model for daily streamflow forecasting in different climate regions,” Ecol Indic., vol. 148, p. 110092, 2023.
[15] F. Kratzert, D. Klotz, G. Shalev, G. Klambauer, S. Hochreiter, and G. Nearing, ”Towards learning universal, regional, and local hydrological behaviors via machine learning applied to large-sample datasets,” Hydrology and Earth System Sciences, vol. 23, no. 12, pp. 5089–5110, 2019. https://doi.org/10.5194/hess-23-5089-2019.
[16] L. Wang, B. Xu, C. Zhang, G. Fu, X. Chen, Y. Zheng, and J. Zhang, ”Surface water temperature prediction in large-deep reservoirs using a long short-term memory model,” Ecological Indicators, vol. 134, October 2021. https://doi.org/10.1016/j.ecolind.2021.108491.
[17] L. Shuofeng, L. Puwen, and K. Koyamada, ”LSTM Based Hybrid Method for Basin Water Level Prediction by Using Precipitation Data,” Journal of Advanced Simulation in Science and Engineering, vol. 8, no. 1, pp. 40–52, 2021. https://doi.org/10.15748/jasse.8.40

[18] N. Ibrahim, N. Ahmad, N. Amalina Mat Jan, Z. Zainudin, N. Syafidah Jamil and A. Azlan, "Comparative Analysis of ARIMA and LSTM Approaches for Monthly River Flow Forecasting in Terengganu," *2024 5th International Conference on Artificial Intelligence and Data Sciences (AiDAS)*, Bangkok, Thailand, 2024, pp. 1-6, doi: 10.1109/AiDAS63860.2024.10730554.

[19]
Y. Essam, Y. F. Huang, J. L. Ng, A. H. Birima, A. N. Ahmed, and A. El-Shafie, “Predicting streamflow in Peninsular Malaysia using support vector machine and deep learning

algorithms," *Scientific Reports*, vol. 12, no. 1, Mar. 2022, doi: https://doi.org/10.1038/s41598-022-07693-4.

[20]
Nouar AlDahoul *et al.*, "Streamflow classification by employing various machine learning models for peninsular Malaysia," *Scientific Reports*, vol. 13, no. 1, Sep. 2023, doi: https://doi.org/10.1038/s41598-023-41735-9.

[21] L. Alfieri et al., "GloFAS—Global ensemble streamflow forecasting and flood early warning," Hydrology and Earth System Sciences, vol. 17, no. 3, pp. 1161–1175, 2013.

[22] S. Harrigan et al., "GloFAS-ERA5 operational global river discharge reanalysis 1979–present," Earth Syst. Sci. Data, vol. 12, pp. 1977–1995, 2020.

[23] C. Prudhomme et al., "Global hydrological reanalyses: The value of river discharge information," Meteorol. Appl., 2024, doi: 10.1002/met.2223.

[24] GloFAS, "Copernicus Emergency Management Service – Global Flood Awareness System," ECMWF, 2024. [Online]. Available: https://www.globalfloods.eu/

[25] M. H. S. Eddin et al., "RiverMamba: A State Space Model for Global River Discharge and Flood Forecasting," arXiv preprint, arXiv:2505.22535, 2024. [Online]. Available: https://arxiv.org/abs/2505.22535

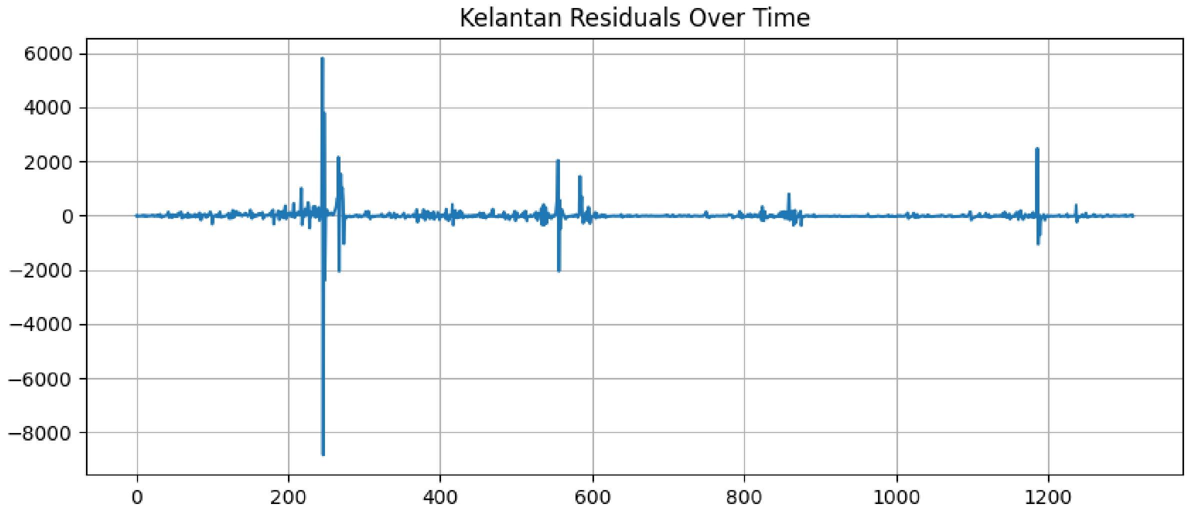
[26] C. J. Gleason and L. C. Smith, "Toward global mapping of river discharge using satellite remote sensing," Water Resour. Res., vol. 50, no. 7, pp. 5351–5360, 2014.

[27] M. H. S. Eddin et al., "RiverMamba evaluation repository," GitHub, 2024. [Online]. Available: https://github.com/river-ai/rivermamba

[28] DGO2, "Hydrométrie Wallonie – HydroMax system," 2024. [Online]. Available: https://hydrometrie.wallonie.be/

[29]
G. Bastin, L. Moens, and P. Dierickx, "On-line river flow forecasting with 'Hydromax': successes and challenges after twelve years of experience," *IFAC Proceedings Volumes*, vol. 42, no. 10, pp. 1774–1779, 2009, doi: https://doi.org/10.3182/20090706-3-fr-2004.00295.

Appendix:




```python
# ADF Test Function(Stationary testing)
def adf_test(series, title=''):
    print(f'Augmented Dickey-Fuller Test: {title}')
    result = adfuller(series.dropna(), autolag='AIC')
    labels = ['ADF Statistic', 'p-value', '# Lags Used', '# Observations']
    out = pd.Series(result[0:4], index=labels)
    for key, val in result[4].items():
        out[f'Critical Value ({key})'] = val
    print(out.to_string())
    if result[1] <= 0.05:
        print("=> Stationary: Reject the null hypothesis (H0)")
    else:
        print("=> Non-Stationary: Fail to reject the null hypothesis (H0)")

# Run ADF Test
adf_test(train_kelantan['Value'], title="Sungai Kelantan")
adf_test(train_sokor['Value'], title="Sungai Sokor")
```

```
Augmented Dickey-Fuller Test: Sungai Kelantan
ADF Statistic           -2.137911
p-value                  0.229546
# Lags Used             11.000000
# Observations         192.000000
Critical Value (1%)     -3.464875
Critical Value (5%)     -2.876714
Critical Value (10%)    -2.574859
=> Non-Stationary: Fail to reject the null hypothesis (H0)
Augmented Dickey-Fuller Test: Sungai Sokor
ADF Statistic         -1.048999e+01
p-value                1.153654e-18
# Lags Used            0.000000e+00
# Observations         1.910000e+02
Critical Value (1%)   -3.465059e+00
Critical Value (5%)   -2.876794e+00
Critical Value (10%)  -2.574901e+00
=> Stationary: Reject the null hypothesis (H0)
```

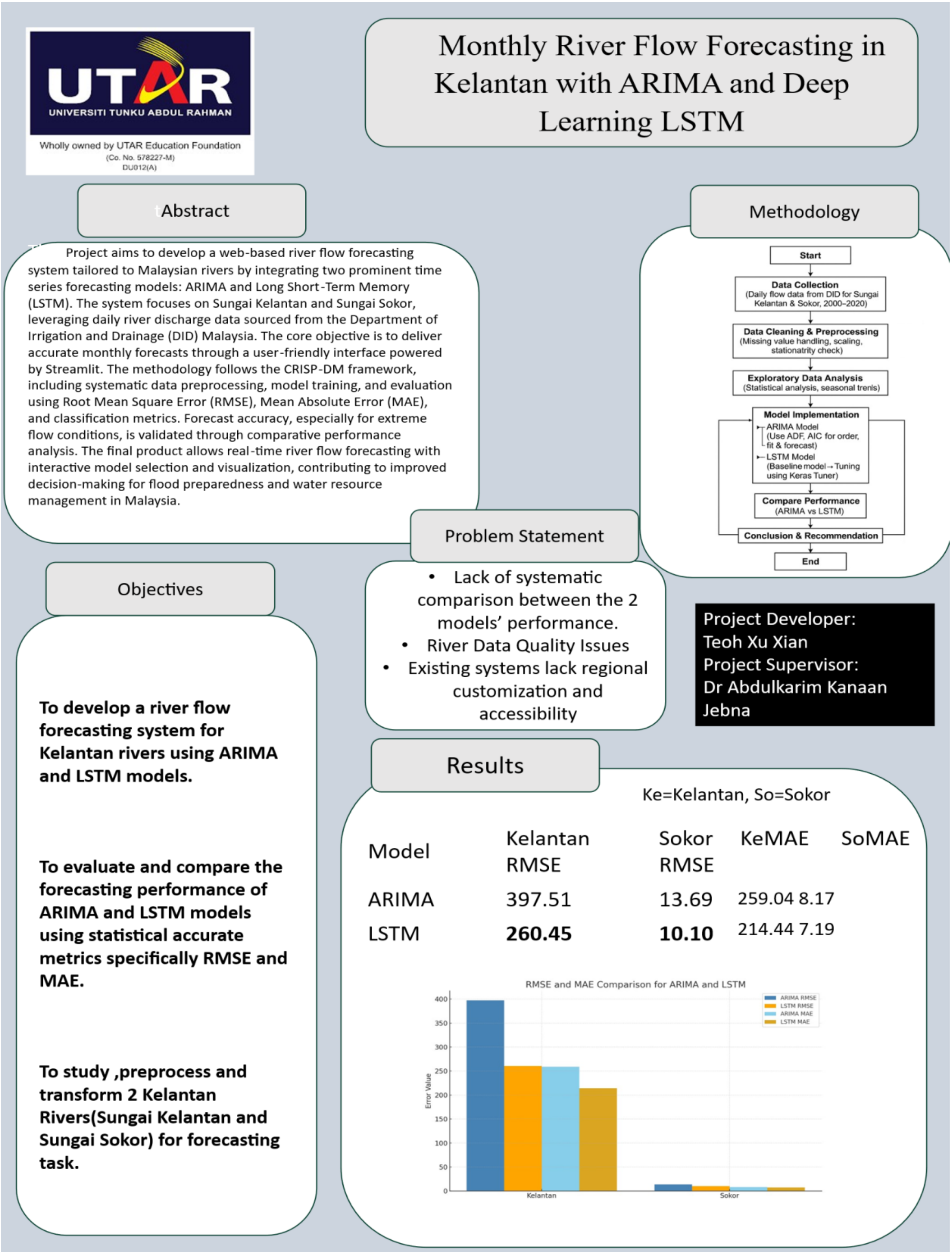
Use auto_arima to Find Best(p,d,q) via AIC minimization.

```
# Auto ARIMA for Sungai Kelantan
auto_model_kelantan = auto_arima(train_kelantan['Value'], seasonal=False, trace=True, information_criterion='aic')

# Auto ARIMA for Sungai Sokor
auto_model_sokor = auto_arima(train_sokor['Value'], seasonal=False, trace=True, information_criterion='aic')

# Summary of best models
print(auto_model_kelantan.summary())
print(auto_model_sokor.summary())
```

```
  self._init_dates(dates, freq)
Final ARIMA Model Summary for Sungai Kelantan:
                               SARIMAX Results
==============================================================================
Dep. Variable:                  Value   No. Observations:                  245
Model:                 ARIMA(1, 0, 0)   Log Likelihood               -1787.288
Date:                Sun, 21 Sep 2025   AIC                           3580.575
Time:                        05:55:54   BIC                           3591.079
Sample:                    01-01-2000   HQIC                          3584.805
                         - 05-01-2020
Covariance Type:                  opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const        424.7630     59.725      7.112      0.000     307.705     541.821
ar.L1          0.3812      0.038      9.929      0.000       0.306       0.456
sigma2      1.249e+05   5822.346     21.453      0.000    1.13e+05    1.36e+05
===================================================================================
Ljung-Box (L1) (Q):                   0.03   Jarque-Bera (JB):              2161.19
Prob(Q):                              0.86   Prob(JB):                         0.00
Heteroskedasticity (H):               1.59   Skew:                             2.75
Prob(H) (two-sided):                  0.04   Kurtosis:                        16.47
===================================================================================

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-step).

Final ARIMA Model Summary for Sungai Sokor:
                               SARIMAX Results
==============================================================================
Dep. Variable:                  Value   No. Observations:                  233
Model:                 ARIMA(0, 0, 1)   Log Likelihood                -990.923
Date:                Sun, 21 Sep 2025   AIC                           1987.847
Time:                        05:55:54   BIC                           1998.200
Sample:                             0   HQIC                          1992.022
                                - 233
Covariance Type:                  opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const         15.2159      2.581      5.895      0.000      10.157      20.274
ma.L1          0.2469      0.059      4.161      0.000       0.131       0.363
sigma2       289.3466      9.520     30.393      0.000     270.687     308.006
===================================================================================
```

```
!python forecast_script.py

Enter the number of months to forecast: 8
/usr/local/lib/python3.11/dist-packages/statsmodels/tsa/base/tsa_model.py:837: ValueWarning: No supported index is available. Prediction results will be given with an integer index beginning at `start`.
  return get_prediction_index(
/usr/local/lib/python3.11/dist-packages/statsmodels/tsa/base/tsa_model.py:837: FutureWarning: No supported index is available. In the next version, calling this method in a model without a supported index will result in an exception.
  return get_prediction_index(

Sungai Kelantan Forecast:
2020-06-01    349.283694
2020-07-01    395.991273
2020-08-01    413.795589
2020-09-01    420.582361
2020-10-01    423.169390
2020-11-01    424.155531
2020-12-01    424.531435
2021-01-01    424.674725
Freq: MS, dtype: float64

Sungai Sokor Forecast:
2020-06-01    13.584005
2020-07-01    15.215901
2020-08-01    15.215901
2020-09-01    15.215901
2020-10-01    15.215901
2020-11-01    15.215901
2020-12-01    15.215901
2021-01-01    15.215901
Freq: MS, dtype: float64
```

POSTER


UTAR
UNIVERSITI TUNKU ABDUL RAHMAN
Wholly owned by UTAR Education Foundation
(Co. No. 578227-M)
DU012(A)
Monthly River Flow Forecasting in Kelantan with ARIMA and Deep Learning LSTM
Abstract
Project aims to develop a web-based river flow forecasting system tailored to Malaysian rivers by integrating two prominent time series forecasting models: ARIMA and Long Short-Term Memory (LSTM). The system focuses on Sungai Kelantan and Sungai Sokor, leveraging daily river discharge data sourced from the Department of Irrigation and Drainage (DID) Malaysia. The core objective is to deliver accurate monthly forecasts through a user-friendly interface powered by Streamlit. The methodology follows the CRISP-DM framework, including systematic data preprocessing, model training, and evaluation using Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and classification metrics. Forecast accuracy, especially for extreme flow conditions, is validated through comparative performance analysis. The final product allows real-time river flow forecasting with interactive model selection and visualization, contributing to improved decision-making for flood preparedness and water resource management in Malaysia.
Methodology
Start
Data Collection
(Daily flow data from DID for Sungai Kelantan & Sokor, 2000–2020)
Data Cleaning & Preprocessing
(Missing value handling, scaling, stationatrity check)
Exploratory Data Analysis
(Statistical analysis, seasonal trenls)
Model Implementation
ARIMA Model
(Use ADF, AIC for order, fit & forecast)
LSTM Model
(Baseline model → Tuning using Keras Tuner)
Compare Performance
(ARIMA vs LSTM)
Conclusion & Recommendation
End
Problem Statement
• Lack of systematic comparison between the 2 models' performance.
• River Data Quality Issues
• Existing systems lack regional customization and accessibility
Objectives
To develop a river flow forecasting system for Kelantan rivers using ARIMA and LSTM models.
To evaluate and compare the forecasting performance of ARIMA and LSTM models using statistical accurate metrics specifically RMSE and MAE.
To study ,preprocess and transform 2 Kelantan Rivers(Sungai Kelantan and Sungai Sokor) for forecasting task.
Project Developer:
Teoh Xu Xian
Project Supervisor:
Dr Abdulkarim Kanaan Jebna
Results
Ke=Kelantan, So=Sokor
Model | Kelantan RMSE | Sokor RMSE | KeMAE | SoMAE
ARIMA | 397.51 | 13.69 | 259.04 | 8.17
LSTM | 260.45 | 10.10 | 214.44 | 7.19
RMSE and MAE Comparison for ARIMA and LSTM
Error Value
0
50
100
150
200
250
300
350
400
Kelantan
Sokor
ARIMA RMSE
LSTM RMSE
ARIMA MAE
LSTM MAE