University of Applied Sciences
HOCHSCHULE EMDEN·LEER

Manual

Time series analysis on stock market Nifty - 50 data Using ARIMA and LSTM modelsl

Author:	Sandesh Nonavinakere Sunil
Matriculation No.:	7026004
Course of Studies:	MBIDA

Author:	Nishan Chandrashekar Poojary
Matriculation No.:	7026796
Course of Studies:	MBIDA

First examiner:	Prof. Dr. Elmar Wings
Submission date:	May 21, 2025

University of Applied Sciences Emden/Leer · Faculty of Technology · Mechanical Engineering Department
Constantiaplatz 4 · 26723 Emden · http://www.hs-emden-leer.de

Contents

List of Figures

List of Tables

Acronyms

1 Overview

This project delivers a interactive web application for time series forecasting of the NIFTY 50 index using two widely adopted models: AutoRegressive Integrated Moving Average (ARIMA) and Long Short-Term Memory (LSTM) neural network. Built with Streamlit, the application allows users to make stock price predictions with minimal technical setup.

It offers two primary modes of use:

- **Next-day prediction** of **Open and Close prices** using historical data fetched live from Yahoo Finance.
- **Backtesting and analysis** by selecting any historical date and comparing model predictions to actual prices.

Key features of the application include:

- Dual-model support: AutoRegressive Integrated Moving Average (ARIMA) for statistical modeling and Long Short-Term Memory (LSTM) for deep learning-based forecasting.
- Simultaneous prediction for both `Open` and `Close` stock prices.
- Automatic data acquisition from Yahoo Finance via the `yfinance` API (Application Programming Interface) for maximum recency and convenience.
- Display of evaluation metrics, including Root Mean Squared Error (RMSE) and Mean Absolute Percentage Error (MAPE), to provide users with interpretable measures of prediction reliability.
- Graphical comparison between predicted and actual values using interactive time series plots.
- Summary tables that contextualize predictions within statistical confidence ranges based on recent performance.

This tool is intended for analysts, researchers, and enthusiasts who wish to apply machine learning (ML) and statistical models for financial forecasting, without needing to write code or build complex environments. The National Stock Exchange Fifty(NIFTY) 50 index, being India's most recognized benchmark stock index, makes this tool particularly valuable for studying market behavior, validating trading strategies, or educational demonstrations in time series modeling.

2 Model Description

This chapter describes the two forecasting models integrated into the application: AutoRegressive Integrated Moving Average (ARIMA) and Long Short-Term Memory (LSTM). Each model is designed to independently predict both the opening and closing prices of the National Stock Exchange Fifty (NIFTY 50) index.

2.1 ARIMA Model

2.1.1 Overview

The AutoRegressive Integrated Moving Average (ARIMA) model is a traditional statistical approach used for analyzing and forecasting univariate time series data. It relies on the assumption that future values of a variable can be explained by its own past values and past forecast errors. ARIMA is particularly effective in capturing linear trends and autocorrelations in historical financial data.

In the context of this application, separate ARIMA configurations are used to model the opening and closing prices of the NIFTY 50 index. The model is fitted using historical stock data to understand patterns that can be extrapolated for future values.

2.1.2 Model Training

Each ARIMA model is trained using historical price data available up to the begining of the year 2025. During training, the model learns the optimal combination of autoregressive, differencing, and moving average parameters required to produce stable and accurate forecasts. These parameter settings are retained and reused for generating predictions, ensuring consistency across sessions.

2.1.3 Model Prediction

When a prediction is requested, the model is dynamically refitted using the most recent available historical data up to the day before the selected forecast date. The model then generates a one-day-ahead forecast. The resulting prediction is compared against the actual value (if available), and both are presented to the user along with numerical evaluation metrics. In addition, statistical confidence intervals based on historical prediction errors are computed to support interpretability.

2.2 LSTM Model

2.2.1 Overview

The Long Short-Term Memory (LSTM) model is a type of recurrent neural network specifically designed to capture long-range dependencies in sequential data. It is particularly well-suited for applications where the temporal dynamics of multiple variables influence the target output. LSTM networks are known for their ability to model complex, non-linear relationships within financial time series data.

In this application, two LSTM models are used to independently predict the opening and closing prices of the NIFTY 50 index. The models incorporate multiple features simultaneously, including both target and auxiliary variables, to enhance predictive accuracy.

2.2.2 Model Training

Each LSTM model is trained using a sequence of historical data comprising multiple variables. A fixed-size lookback window is used to define the sequence of input observations, which serve as the model's temporal context. Prior to training, all feature values are normalized to a uniform scale to improve learning stability and convergence. The training process involves optimizing internal network parameters to minimize prediction errors over the training period.

2.2.3 Model Prediction

For a given forecast date, the LSTM model extracts the most recent sequence of historical data to construct its input. This sequence is preprocessed in the same manner as the training data and then passed to the model to generate a forecast for the next day. The resulting prediction is transformed back into its original scale for interpretation.

Alongside the predicted value, the system presents evaluation metrics based on recent data, including the Root Mean Squared Error (RMSE) and the Mean Absolute Percentage Error (MAPE). These metrics are used to generate prediction ranges that provide context and uncertainty estimates around the forecasted price. This enables users to understand not just the point estimate, but also the likely range within which the actual value may fall.

3 Installation

3.1 System Requirements

The Stock Predictor application is delivered as a cloud-hosted web-based platform, accessible via any modern web browser. To ensure smooth usage and real-time responsiveness, users should access the application from a device with a stable internet connection is necessary, particularly when handling larger data queries or visualizations.

The application has been tested for compatibility with recent versions of major browsers, including Google Chrome (version 110 and above), Mozilla Firefox (version 100 and above), and Microsoft Edge (version 110 and above) on both Windows and macOS operating systems. While the application is accessible on mobile devices, optimal performance and full feature availability are ensured when accessed via desktop or laptop environments.

3.2 No Local Installation Required

Unlike traditional software distributions, this application does not require users to install any software packages, scripting environments, or dependencies on their local machine. The entire execution of forecasting models—such as AutoRegressive Integrated Moving Average (ARIMA) and Long Short-Term Memory (LSTM)—takes place on the cloud infrastructure.

All data preprocessing, model inference, evaluation metric calculations, and visualizations are performed on the backend server. This architecture eliminates the need for local processing, dependency management, or compatibility adjustments across operating systems.

3.3 Accessing the Application

To use the Stock Predictor application:

1. Visit the web link provided by the deployment platform or project maintainer.
2. Open the link in a supported browser to load the user interface.
3. Interact with the application by selecting a prediction model, choosing a date, and viewing the results in real time.

No registration or sign-in is required for standard use, unless access restrictions are configured for organizational deployment. The browser interface replicates the look and feel of a desktop dashboard, providing users with immediate access to forecasting tools, evaluation metrics such as Root Mean Squared Error (RMSE) and Mean Absolute Percentage Error (MAPE), and dynamic visualizations. All outputs are generated on demand and do not require local data storage or computation.

4 Functions and Features

4.1 Graphical User Interface (GUI)

The Stock Predictor application features a modern, browser-based Graphical User Interface (GUI) that is designed to be intuitive, clean, and responsive. The layout allows users to interact with forecasting models in a visually guided manner without requiring any technical background.

4.2 Dual Model Integration

The application provides seamless integration of two distinct forecasting models, allowing users to choose the one best suited to their analytical needs.

4.3 Fully Automated Cloud-Based Predictions

All predictions—whether for historical backtesting or future forecasting—are generated dynamically using live data sources. The platform eliminates the distinction between static and dynamic modes found in desktop applications.

4.3.1 Real-Time Data Integration

The system connects to a live financial data provider to retrieve historical and current stock market data. This allows users to generate predictions for both past and near-future dates without needing to upload or manage datasets manually.

4.3.2 No Manual Updates Required

The application automatically fetches and preprocesses all required data in the background. There is no need for users to provide custom data files or interact with file systems. This ensures data consistency and eliminates human error in data preparation.

4.4 Interactive Output and Evaluation Feedback

4.4.1 Real-Time Result Display

Once a prediction is generated, the results are immediately shown in the interface. The following information is presented clearly:

- The predicted opening and closing prices for the selected date.
- The actual values (if available) for historical dates.
- Evaluation metrics including Root Mean Squared Error (RMSE) and Mean Absolute Percentage Error (MAPE).
- Prediction intervals based on RMSE and MAPE, visualized alongside the main results.

4.4.2 Integrated Logging and Notifications

Instead of traditional log files, the cloud application provides in-app notifications to alert users of data validation issues, model selection warnings, or connectivity errors. This helps users troubleshoot common problems without needing access to system logs or backend infrastructure.

4.4.3 Functional Controls and Interactivity

The application interface includes several interactive elements that allow users to operate the system efficiently and intuitively. Each control has been designed with a clear function:

- **Model Selection Dropdown** – Enables the user to choose between the available forecasting models: Long Short-Term Memory (LSTM) or AutoRegressive Integrated Moving Average (ARIMA). The selected model determines the forecasting strategy to be executed.
- **Date Picker Calendar** – Provides users with a scrollable calendar to select the date for prediction. The system enforces strict constraints to maintain data integrity:
  - **Historical Forecasting:** Users can select any past trading date from the year 2007 onwards for backtesting and analysis.
  - **Forward Prediction:** Only the next available trading day (typically tomorrow) is enabled for forecasting future prices.
  - **Date Restriction:** All dates beyond tomorrow are disabled to prevent unsupported predictions. Similarly, weekends and non-trading days are automatically excluded.

  - **Date Restriction:** All the Saturday and Sundays are disabled as market is closed on those days

- **Prediction Trigger** – This control executes the selected forecasting model using the most recent market data. Upon activation, the system performs model inference and displays predicted prices, evaluation metrics, and visual plots on the interface.

5 First Step

5.1 Accessing the Application

The Stock Predictor application is deployed on a secure cloud platform and does not require installation on the user's local machine. It can be accessed directly through any supported web browser.

5.1.1 Opening the Platform

Users can begin by navigating to the application link provided by the project administrator or hosting environment. This link opens a responsive web-based dashboard that serves as the main interface for interaction.

5.1.2 User Interface Initialization

Upon loading, the browser automatically initializes the platform's components. This includes model loading, data connection setup, and preparation of interactive widgets. Depending on internet speed and system responsiveness, this initialization may take a few seconds during the first use.

5.2 No Setup or Installation Required

The entire application is managed in the cloud. Users do not need to download, configure, or install any software packages. All computations—including data preprocessing, model execution, evaluation, and plotting—are performed on the backend servers.

As a result, the user experience is consistent across devices and operating systems. The system is ready to use immediately after the web page is opened, allowing users to focus on forecasting tasks without any technical setup or overhead.

6 Data Requirements

6.1 Integrated Data Access

The Stock Predictor application retrieves stock market data for the National Stock Exchange Fifty (NIFTY 50) index directly from a real-time financial data provider. This eliminates the need for users to manually upload datasets or prepare data files.

6.1.1 Data Source

Historical and current stock prices are automatically fetched using a live connection to Yahoo Finance. This ensures that the application always operates on up-to-date information without requiring any user intervention. The system downloads data such as:

- Opening and closing prices of the NIFTY 50 index.
- Daily high and low values.

6.1.2 Date Coverage

The application supports predictions for any trading day from October 2007 up to the current date. Users may select either a past date (for backtesting) or a near-future date (e.g., tomorrow) to forecast the opening and closing prices.

6.2 No Manual Data Handling

6.2.1 No File Upload Required

Unlike traditional forecasting tools, this application does not require users to upload any Comma-Separated Values (CSV) files. All data ingestion is managed in the backend, ensuring that the correct format, column structure, and completeness are maintained automatically.

6.2.2 Automated Preprocessing

Once the data is fetched from the remote source, the application performs internal preprocessing to prepare it for prediction. This includes:

- Filtering for valid trading days (excluding weekends).
- Cleaning and aligning feature values (e.g., open, close, high, low, volume).
- Preparing input sequences for the forecasting models based on the most recent available data.

6.3 Data Consistency and Quality

All datasets used by the application adhere to consistent structures verified at runtime. This ensures that forecasting models such as AutoRegressive Integrated Moving Average (ARIMA) and Long Short-Term Memory (LSTM) receive clean, well-structured input for reliable performance. By automating the data layer, the platform significantly reduces the risk of user error and improves forecast accuracy.

7 Running the Model

7.1 Choosing the Prediction Date

The user initiates the forecasting process by selecting a specific date for which they wish to generate predictions. The system allows forecasting of both past and upcoming trading days, subject to the availability of historical data.

7.1.1 Date Selection Interface

The main interface features an interactive calendar widget labeled "Select prediction date." Users can click on the widget to open a date picker and choose any valid trading date. The application is designed to fetch historical stock data up to the selected date in order to support model input preparation.

7.1.2 Supported Date Range

The application supports forecasting from early October 2007 until the current calendar day. Users may also select the next business day (typically tomorrow) for short-term forecasting. Since the data is pulled dynamically from a financial data provider, no limitations exist based on internal or uploaded datasets.

7.2 Selecting the Forecasting Model

The system supports two distinct forecasting models: AutoRegressive Integrated Moving Average (ARIMA) and Long Short-Term Memory (LSTM).

7.2.1 Model Selection Interface

Users can choose the desired forecasting model using a dropdown menu on the main dashboard. The available options include:

- **LSTM**: A neural network-based model capable of learning complex temporal dependencies in multivariate time series data.
- **ARIMA**: A statistical model ideal for capturing linear trends and seasonality in univariate data.

Each model has been pre-trained and optimized for the prediction of stock market opening and closing prices.

7.3 Initiating the Prediction

7.3.1 Triggering the Forecast

Once the user selects the desired model and date, they may initiate the forecast by clicking the "Predict" button. The application will automatically:

1. Retrieve the most recent stock market data up to the selected date.
2. Prepare and scale the input sequence as required by the selected model.
3. Generate forecasts for both opening and closing prices of the selected trading day.

7.3.2 Results Display

After prediction, the application presents the following outputs on the main interface:

- **Predicted Opening Price** and **Predicted Closing Price** for the selected date.
- **Actual Opening and Closing Prices**, if available for historical dates.
- **Evaluation Metrics**, including Root Mean Squared Error (RMSE) and Mean Absolute Percentage Error (MAPE), calculated based on recent historical performance.
- **Forecast Uncertainty Ranges** based on RMSE and MAPE, shown alongside the predictions to assist with interpretability.

7.4 Important Considerations

- Users must ensure the selected date falls within the range supported by the data provider. Weekend dates are automatically excluded from prediction eligibility.
- If insufficient data is available or if a non-trading day is selected, the application will notify the user via in-app alerts.
- Since the system is cloud-hosted, all forecasting operations are handled remotely, and users are not required to manage any configurations or data manually.

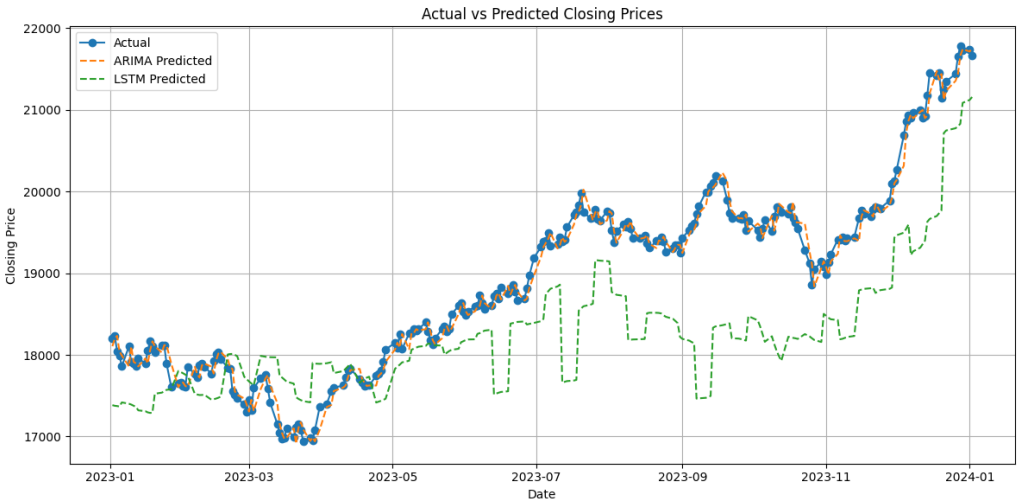

Figure 7.1: LSTM and ARIMA prediction closing price graph

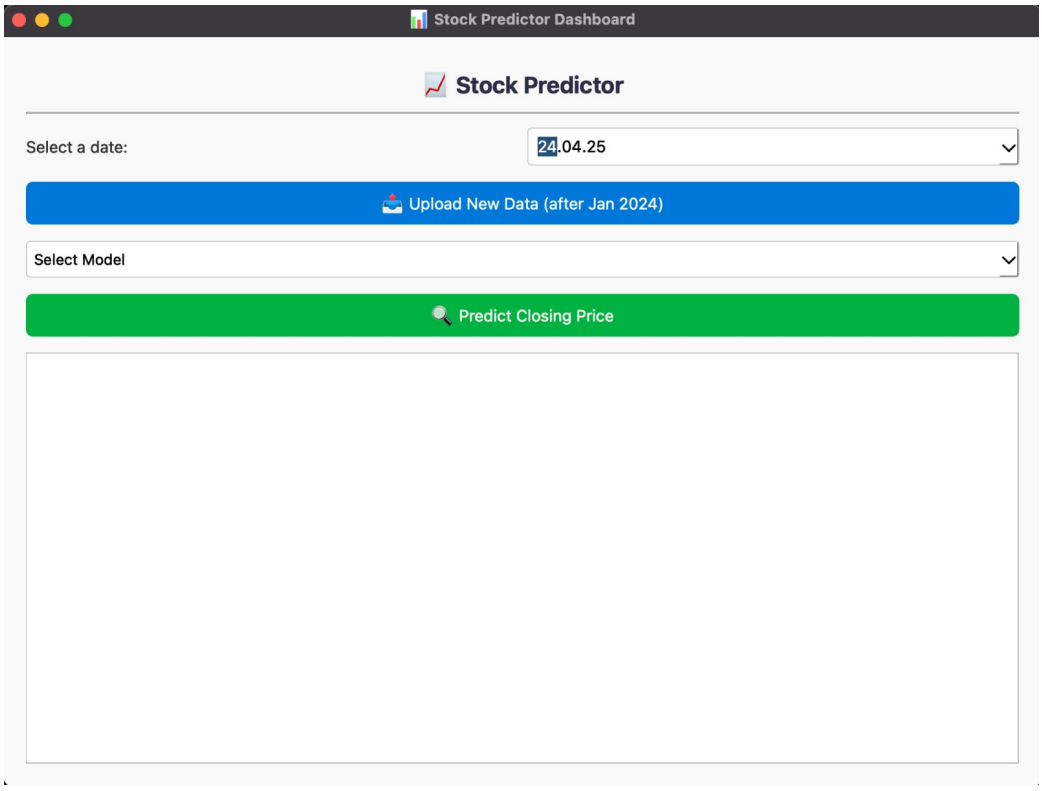

Figure 7.2: Stock predictor GUI

8 Module Evaluation Metrics

Both models are evaluated using the following metrics:

Metric	Description
RMSE (Root Mean Squared Error)	Penalizes larger errors
MAPE (Mean Absolute Percentage Error)	Scale-independent error measure

Table 8.1: Evaluation metrics used for model performance

We have seen that we get the model attributes table also as an downloadable file . how do we measure which model performs better ? lets consider a sample output of the performance metric as shown below .

Model	RMSE	MAPE
ARIMA	113.67	1.21%
LSTM	101.45	1.05%

Table 8.2: Sample Evaluation Metric Output

lets understand the above metrics. lets take the LSTM values to begin with:

- An RMSE of 101.45 means large errors are present, even if the average error (MAE) is lower.
- A MAPE of 1.05% means your predictions are, on average, 1.05% off from the actual NIFTY 50 values.
- LSTM outperforms ARIMA across all metrics , because It's more accurate on average (lower MAE). It has fewer large errors (lower RMSE).It also shows Smaller error relative to the actual index value (lower MAPE).

So, in plain terms: your LSTM model is making better, more consistent predictions for the NIFTY 50 than your ARIMA model based on this evaluation.

9 Maintenance

9.1 System Monitoring and Logging

The cloud-hosted Stock Predictor application includes an integrated monitoring system to ensure stability, detect anomalies, and support issue resolution.

9.1.1 Purpose

The internal monitoring infrastructure records key application events such as model executions, forecasting outcomes, and system-level errors. This functionality is critical for identifying issues related to data access, prediction anomalies, or service disruptions. It also supports diagnostic reviews by the development team to improve future performance and robustness.

9.1.2 User Visibility

Users receive immediate feedback through on-screen alerts or informational messages within the Graphical User Interface (GUI). These notifications cover scenarios such as invalid date selections, unavailable data, or missing model selections. In case of a critical failure, a descriptive error message is displayed to guide the user without exposing technical details.

9.2 Application Updates

The application is maintained through a centralized deployment system that ensures all users operate on the most recent and stable version. Since the application is web-based, updates are handled automatically on the server side.

9.2.1 Update Lifecycle

Improvements to model accuracy, performance enhancements, interface changes, and bug fixes are deployed periodically. Updates do not require any action from the user. Once changes are published to the cloud environment, all users accessing the platform benefit from the enhancements in real time.

9.2.2 No Manual Installation Required

Because the application is hosted online, users do not need to download or install any files. This removes the burden of managing software versions or manually replacing outdated builds. The web-based architecture ensures seamless upgrades and uniform user experience across different sessions and devices.

10 Troubleshooting

10.1 Application Fails to Load in Browser

If the application does not load properly or remains stuck on the initial screen:

10.1.1 Possible Causes

- A slow or unstable internet connection may prevent the application from retrieving data.
- The browser may be using an unsupported version or have JavaScript disabled.
- Temporary server issues may affect availability.

10.1.2 Recommended Actions

- Ensure you are connected to a reliable internet network.
- Use a supported and updated web browser such as Google Chrome, Mozilla Firefox, or Microsoft Edge.
- Refresh the browser page or clear the cache and try again after a few minutes.
- If the issue persists, check for platform-specific service announcements.

10.2 Prediction Does Not Trigger

If clicking the "Predict" button does not produce any output:

10.2.1 Likely Causes

- No forecasting model has been selected.
- An invalid date was chosen (e.g., a weekend or a public holiday).

10.2.2 Suggested Fixes

- Select a valid model from the dropdown list, either Long Short-Term Memory (LSTM) or AutoRegressive Integrated Moving Average (ARIMA).
- Make sure the selected date corresponds to a valid trading day.

10.3 Unexpected or Missing Results

If the predicted values appear to be incorrect or are not displayed at all:

10.3.1 Potential Reasons

- The selected date may be too close to the current day, leading to insufficient prior data.
- For LSTM forecasts, the required input sequence of 60 prior trading days may not be available due to data gaps.
- The financial data provider may not have released data for the selected day yet.

10.3.2 Resolution Steps

- Choose a date that is at least one trading day ahead and well-supported by historical data.
- Prefer dates for which financial markets have already closed and official data is likely available.
- For Long Short-Term Memory (LSTM) forecasts, ensure at least 60 valid consecutive trading days precede the selected date.

10.4 Performance and Compatibility Issues

If the interface becomes slow, unresponsive, or visually distorted:

10.4.1 Possible Causes

- The browser session may be overloaded with cached data or extensions.

10.4.2 Solutions

- Try restarting your browser or switching to a different supported browser.
- Close other tabs or applications running in the background to free up memory.

11 FAQs

11.1 Do I need to install Python or any other software to use this application?

No. The Stock Predictor is a fully cloud-hosted application that runs in a standard web browser. All computations, forecasting models, and data processing are handled on remote servers. Users do not need to install Python, libraries, or any additional software on their local machine.

11.2 Can I predict stock prices for any date?

Yes, within the range supported by the financial data provider. The application allows users to forecast prices for:

- Historical trading dates, for backtesting and model comparison.
- The next available trading day, typically tomorrow, for forward-looking analysis.

Dates falling on weekends or market holidays are automatically excluded to ensure data validity.

11.3 Do I need to upload any stock data?

No. The application automatically retrieves historical and current stock market data directly from a live financial data provider. This eliminates the need for manual uploads, file formatting, or data validation by the user.

11.4 How do I know if the prediction was successful?

Once the "Predict" button is clicked, the application immediately displays the following in the interface:

- Predicted values for both opening and closing prices.
- Actual observed values (if available for historical dates).
- Evaluation metrics such as Root Mean Squared Error (RMSE) and Mean Absolute Percentage Error (MAPE), along with corresponding confidence intervals.

If the prediction could not be completed (due to unavailable data or model constraints), the system will provide a clear notification.

11.5 Can I use my own forecasting models?

No. The application is built around validated and pre-integrated forecasting models, including Long Short-Term Memory (LSTM) and AutoRegressive Integrated Moving Average (ARIMA). Custom models are not supported in the cloud-hosted version to maintain consistency, accuracy, and security across all users.

11.6 What should I do if the application freezes or stops responding?

If the application becomes unresponsive:

- Refresh your browser and ensure your internet connection is stable.
- Confirm that your device meets the recommended minimum of 8 GB Random Access Memory (RAM).
- Try switching to a supported browser such as Google Chrome or Mozilla Firefox.
- If the issue persists, report it through the designated support contact.

11.7 Will the application receive updates?

Yes. Because the application is deployed on a cloud platform, updates are managed centrally by the development team. Users automatically receive the latest improvements and new features without needing to install or download anything manually.