# A PROJECT REPORT ON

# "Comparative Analysis of LSTM and GRU Algorithms for Stock Market Prediction: Development of a Web Application"

BY


| | |
|---|---|
| Sahil Jhodge | B190904220 |
| Ankit Singh | B190904252 |
| Vallabh Yedave | B190904263 |
| Vaibhav Survase | B190904253 |

**UNDER THE GUIDANCE OF**

**Prof. D. A. Gore**

**DEPARTMENT OF COMPUTER ENGINEERING**

**Navsahyadri Education Society's Group of Institutions. Naigaon, Pune. 041**

**SAVITRIBAI PHULE PUNE UNIVERSITY**


**2023-24**

# CERTIFICATE

This is to certify that,

| | |
|---|---|
| Sahil Jhodge | B190904220 |
| Ankit Singh | B190904253 |
| Vallabh Yedave | B190904263 |
| Vaibhav Survase | B190904253 |

of class B.E Computer Engineering, We have successfully completed their project stage one work on **”Comparative Analysis of LSTM and GRU Algorithms for Stock Market Prediction: Development of a Web Application”** at Navsahyadri Group of Institute Faculty of Engineering, Pune in the partial fulfillment of the Graduate Degree course in B.E at the Department of **Computer Engineering**, in the Academic Year 2023-2024 as prescribed by the Savitribai Phule Pune University.

**Prof. D. A. Gore**
Internal Guide
Dept. of Computer Engineering

**Dr. S. N. Gujar**
Head Of Department
Dept. Computer Engineering

Place : Pune

Date :

# CERTIFICATE FOR CONDUCTION OF EXAM

This is to certify that B.E. Project Stage-II examination of **Sahil Jhodge Ankit Singh Vallabh Yedave Vaibhav Survase** with titled **" Comparative Analysis of LSTM and GRU Algorithms for Stock Market Prediction: Development of a Web Application "** has been held on 27/05/2024 at Department of Computer Engineering of Navsahyadri Group of Institutes, Faculty of Engineering, Naigaon [Nasarapur], Pune412213

Internal Examiner External Examiner

Place : Navsahyadri Group of Institutes, Faculty of Engineering, [Nasarapur], Pune.
Date : 27/05/2024

## Acknowledgment

*It gives us great pleasure in presenting the project report on* ***'Comparative Analysis of LSTM and GRU Algorithms for Stock Market Prediction: Development of a Web Application)'****.*

*We would like to take this opportunity to thank our internal guide* ***Prof. D. A. Gore*** *for giving us all the help and guidance we needed. We are really grateful to them for their kind support. Their valuable suggestions were very helpful.*

*We are also thankful to all staff members and our project coordinator* ***Dr. S.N. Gujar*** *for his valuable guidance.*

*We are also grateful to* ***Dr. S.N. Gujar****, Head of Computer Engineering Department,* ***NGI-FOE, PUNE*** *for his indispensable support, suggestions.*

Sahil Jhodge B190904220
Ankit Singh B190904252
Vallabh Yedave B190904263
Vaibhav Survase B190904253

# Abstract

Machine learning algorithms for stock market prediction have attracted a lot of interest because they have the ability to help investors make wise selections. This study compares the Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) algorithms for stock market prediction. A comparative empirical analysis reveals that GRU outperforms LSTM in stock price prediction. The methodology section provides a detailed description of the experimental setup, evaluation metrics, and data preparation techniques used to compare LSTM with GRU. The results show how successfully the complex patterns present in stock market data are captured by GRU. A web application for stock prediction is made using the GRU algorithm following a comparison of the algorithms. The application provides users with real-time stock predictions based on previous data, allowing them to make well-informed investment decisions on time.

# INDEX

# List of Figures

# List of Tables

# CHAPTER 1

# INTRODUCTION

## 1.1 MOTIVATION

The stock market's inherent volatility and complexity present a constant challenge for achieving accurate predictions. While traditional methods like technical and fundamental analysis offer valuable insights, they often struggle to capture the intricate temporal dependencies and hidden patterns within historical stock data. This motivates us to explore the potential of Gated Recurrent Unit (GRU) networks. GRU outperforms LSTM in stock price prediction, showcasing its ability to effectively learn and model complex patterns in sequential data. By leveraging the power of GRU, we aim to develop a robust and reliable stock prediction system that provides investors and financial professionals with accurate and timely forecasts, ultimately leading to better-informed decision-making.

## 1.2 GOAL

The primary goal of this project is to develop and implement a web application for stock market prediction using GRU networks. This application will leverage historical stock data to provide users with accurate and insightful forecasts of stock price movements, empowering them to make informed investment decisions.

## 1.3 OBJECTIVES

### 1.3.1 Data Acquisition

Utilize the `yfinance` library to acquire historical stock data from Yahoo Finance API, ensuring a comprehensive and reliable data source.

### 1.3.2 Data Preprocessing

Employ `NumPy` and `Pandas` libraries for efficient data preprocessing and manipulation, including tasks like data cleaning, normalization, and feature engineering.

### 1.3.3 GRU Model Implementation

Leverage the `TensorFlow` library to implement a GRU network architecture, taking inspiration from the successful architecture described in the research paper.

### 1.3.4 Model Training and Optimization

Train the GRU model on the prepared historical data and fine-tune hyperparameters (e.g., number of layers, units per layer) to achieve optimal prediction accuracy.

### 1.3.5 Web Application Development

Utilize the `Django` framework for creating an intuitive online application that incorporates the GRU model to forecast stock prices in real time.

### 1.3.6 Performance Evaluation

Assess the effectiveness of the GRU model and the web application by employing metrics such as Root Mean Squared Error (RMSE) and conducting visual comparisons between predicted and actual values.

## 1.4 PROBLEM STATEMENT

The intricate and unpredictable nature of financial markets makes it extremely difficult to predict stock market fluctuations with any degree of accuracy. Conventional prediction techniques frequently fall short in capturing the complex non-linear correlations and temporal dependencies found in historical data. In order to overcome this difficulty, this study makes use of GRU networks, which have outperformed LSTM in stock price prediction tests.

## 1.5 SCOPE OF THE WORK

This project encompasses the development of a GRU-based stock prediction web application using historical data acquired from the Yahoo Finance API. The scope includes data acquisition, preprocessing, GRU model implementation, training, evaluation, web application development, and performance assessment. The project does not include the integration of sentiment analysis or other external data sources at this stage.

## 1.6 OUTCOMES

### 1.6.1 Functional Web Application

A user-friendly web application built with Django that provides real-time stock price predictions using a trained GRU model.

### 1.6.2 Accurate Predictions

The GRU model integrated into the web application demonstrates a high degree of prediction accuracy, as evaluated by RMSE and visual analysis.

### 1.6.3 Enhanced User Experience

The web application provides users with an intuitive interface, insightful visualizations, and the ability to customize prediction parameters.

# CHAPTER 2

# LITERATURE SURVEY

This chapter explores key research papers relevant to stock market prediction using machine learning, particularly focusing on LSTM and GRU models. We'll analyze each paper's propositions and limitations to gain a comprehensive understanding of the current research landscape.

## 2.1 PAPER REVIEWS

### 2.1.1 Research on Improved GRU-Based Stock Price Prediction Method

**Authors:** Chi Chen, Lei Xue, Wanqi Xing

**Publication:** Applied Sciences, 2023

**Paper Proposed:**

This paper tackles the critical issue of overfitting in GRU-based stock prediction models, especially when working with limited datasets. They introduce a novel approach using a reconstructed dataset, incorporating data from related stocks within the same industry. This method aims to enrich the features learned by the GRU, allowing it to capture broader industry trends and improve generalization capabilities.

**Limitations:**

The paper focuses primarily on historical stock price data, neglecting the potential impact of external factors like news sentiment and economic indicators. Additionally, the dataset size, even with the reconstruction technique, might still be limited for capturing highly complex market dynamics.

### 2.1.2 Stock Prediction Based on Bidirectional GRU with CNN and Feature Selection

**Authors:** Qi Zhou, Chao Zhou, Xiaojun Wang

**Publication:** PLOS ONE, 2022

**Paper Proposed:**

This research explores a powerful hybrid approach, combining a bidirectional GRU with a Convolutional Neural Network (CNN) and feature selection techniques for enhanced stock price prediction. The CNN layer extracts local patterns from the stock data, while the bidirectional GRU captures both forward and backward temporal dependencies, leading to a more comprehensive understanding of market trends.

**Limitations:** The computational complexity of this hybrid model can be significant, requiring substantial resources for training and prediction. Additionally, the interpretability of such complex models can be challenging, hindering the understanding of the factors driving the predictions.

### 2.1.3 Application of Artificial Intelligence in Stock Market Forecasting: A Critique, Review, and Research Agenda

**Authors:** Ritika Chopra, Gagan Deep Sharma

**Publication:** Journal of Risk and Financial Management, 2021

**Paper Proposed:**

This paper presents a comprehensive review and analysis of 148 studies on AI-based stock market forecasting. The authors emphasize the need for rigorous data selection, model optimization, and transparency in AI-driven prediction systems. They highlight the challenges of data bias, overfitting, and the need for robust evaluation metrics.

**Limitations:** The review broadly covers AI techniques in stock prediction without detailing the pros and cons of specific architectures like GRU. It also lacks concrete solutions for the identified challenges, calling instead for future research.

#### 2.1.3.1 Stock Market Prediction Using LSTM Recurrent Neural Network

**Authors:** Adil Moghar, Mhamed Hamiche

**Publication:** Procedia Computer Science, 2020

**Paper Proposed:**

This research highlights the effectiveness of LSTM networks for stock market prediction. They demonstrate the model’s ability to capture temporal dependencies in financial time series data, leading to promising results in predicting future price movements.

**Limitations:** The paper focuses on LSTM without exploring the potential of GRUs, which are known for their computational efficiency and comparable performance. It also primarily relies on historical stock data, neglecting the potential benefits of incorporating news sentiment and other external factors.

### 2.1.4 Conclusion

This literature survey highlights the evolving landscape of AI-driven stock prediction, noting the efficiency and effectiveness of GRUs compared to LSTMs. Our project leverages GRU’s strengths by mitigating overfitting through data augmentation and regularization, integrating news sentiment and other data for a comprehensive market view, and ensuring computational efficiency for real-time applications. We also emphasize transparency and ethical use within risk management. By addressing key challenges identified in the literature, our project aims to advance AI-powered stock prediction, providing a reliable tool for investors and financial professionals.

# CHAPTER 3

# REQUIREMENT ANALYSIS

## 3.1 RESPONSIBILITIES OF DEVELOPER

The development team's roles are crucial for the success of the project. Key responsibilities are divided among team members with specialized expertise:

- **Data Engineer/Scientist:** Identifies and accesses reliable sources of historical stock data, preprocesses and cleans the data, and engineers meaningful features for model training.
- **Machine Learning Engineer/Researcher:** Selects and designs appropriate GRU models for stock prediction, trains and optimizes the models for accuracy, and evaluates their performance.
- **Software Developer:** Designs and implements the system architecture, ensuring efficiency and scalability. Develops a user-friendly interface (web application or API) for interacting with the system.

Additionally, the team shares responsibilities for testing and validation, deployment and maintenance, and continuous monitoring and evaluation of the system.

## 3.2 EXTERNAL INTERFACE REQUIREMENT

### 3.2.1 User Interface

The system's user interface should be intuitive and accessible:

- **Input:** Allow users to easily input stock symbols for prediction.

- **Output:** Clearly display predicted stock prices with options for visualization and analysis.
- **Visualization:** Utilize effective visualizations for data and prediction presentation. Furthermore, the interface should be responsive across devices and prioritize a positive user experience.

- ### 3.3 SYSTEM FEATURES

  The core functionalities of the stock prediction system include:

- **Real-Time Prediction:** Generate real-time stock price predictions using the trained GRU model.
- **Historical Data Analysis:** Allow users to access and analyze historical stock price data.
- **Visualization Tools:** Provide interactive visualizations to understand data trends and predictions.
- **Customization:** Enable users to adjust parameters for tailored predictions.

## 3.4 CONCLUSION

This chapter has defined the project's key requirements, outlining developer responsibilities, interface needs, and system features. These specifications will guide the successful development and implementation of the GRU-based stock prediction system.

# CHAPTER 4

# PROPOSED SYSTEM

## 4.1 BACKGROUND NEEDED

To effectively understand and implement the proposed GRU-based stock prediction system, a foundational understanding of the following concepts is essential:

### 4.1.1 Financial Time Series Data

- **Characteristics:** Proper data pretreatment and model selection require an understanding of the distinctive features of financial time series data, such as non-stationarity, seasonality, trends, and noise.

- **Technical Indicators:**Knowledge of popular technical indicators (such as Bollinger Bands, relative strength index, and moving averages) that are based on past prices can give context and possibly increase prediction accuracy.

- **Fundamental Analysis:** While our project focuses on technical analysis using historical prices, a basic understanding of fundamental analysis concepts (e.g., company financials, economic indicators) can provide context for interpreting predictions and market trends.

### 4.1.2 Recurrent Neural Networks (RNNs)

- **Sequence Modeling:** Understanding the core principles of RNNs and their ability to process sequential data is essential as GRU is a type of RNN architecture.

- **Vanishing/Exploding Gradient Problem:** Familiarity with the challenges of vanishing and exploding gradients in RNNs and how GRUs address these issues through their gating mechanism is crucial for appreciating the advantages of GRUs.

### 4.1.3 Gated Recurrent Unit (GRU)

- **Architecture:** It is essential to have a thorough understanding of the GRU (Gated Recurrent Unit) architecture, including its update gate and reset gate functions and how they control information flow in the network, in order to apply and improve the model.

- **Advantages and Limitations:** Understanding the advantages of GRUs over other RNN architectures (e.g., LSTMs) in terms of efficiency and suitability for financial time series, as well as their limitations, is important for making informed design choices.

### 4.1.4 Data Preprocessing Techniques

- **Handling Missing Values:** Knowledge of techniques for handling missing values in time series data (e.g., imputation, deletion) is crucial for preparing the data for GRU model training.

- **Normalization and Standardization:** Understanding methods for data normalization or standardization (e.g., MinMaxScaler, StandardScaler) is important to ensure features are on a similar scale and improve model convergence.

## 4.2 ALGORITHM

### 4.2.1 Long Short-Term Memory (LSTM)

Recurrent neural networks (RNNs) of the Long Short-Term Memory (LSTM) variety were created in response to the vanishing gradient issue that causes conventional RNNs to struggle with lengthy input sequences. Their distinctive architecture, which consists of memory cells and gates that regulate information flow, allows them to accomplish this.

**LSTM Advantages:**

- Great for identifying enduring dependencies: LSTMs are ideally suited for time series prediction applications such as stock price forecasting because they can efficiently learn and retain knowledge from previous time steps.

- Robustness to noise: LSTMs exhibit good performance even with noisy data, a common characteristic of financial markets.

**LSTM Disadvantages:**

- Computational complexity: Training LSTMs can be computationally expensive due to their complex architecture.

- Black box nature: Understanding the internal workings and decision-making of LSTMs can be challenging, making it difficult to interpret their predictions.

### 4.2.2 Gated Recurrent Unit (GRU)

GRUs are another type of RNN architecture similar to LSTMs but with a simplified structure. They also address the vanishing gradient problem and are effective for sequential data processing.

**GRU Advantages:**

- Comparable performance to LSTMs: GRUs often achieve similar prediction accuracy as LSTMs on many tasks.

- Lower computational cost: Due to their simpler architecture, GRUs require less computational resources than LSTMs during training and prediction.

- Faster training: GRUs generally train faster than LSTMs, making them more efficient for time-sensitive applications.

**GRU Disadvantages:**

- Potentially less powerful for very long sequences: While effective for many tasks, GRUs might be slightly less powerful than LSTMs for extremely long sequences with complex dependencies.

### 4.2.3 Why GRU is a Better Choice for this Project

Considering the pros and cons of each algorithm, GRU is the preferred choice for our stock prediction project due to the following reasons:

- Balance of performance and efficiency: GRUs offer a good balance between prediction accuracy and computational efficiency, suitable for real-time applications.

- Suitability for financial data: GRUs handle the non-linear nature of financial time series data well, capturing temporal dependencies essential for stock forecasting.

- Lower resource requirements: GRUs’ reduced computational demands make them ideal for resource-constrained environments or large-scale data processing.

While LSTMs are strong contenders, GRUs’ computational advantages make them more practical. MLR is less suitable due to its inability to capture non-linear relationships inherent in stock market dynamics.

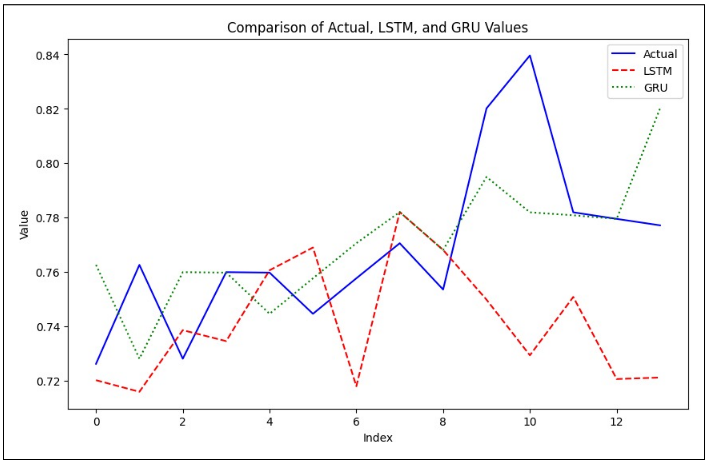



Figure 4.1: NP Problem

Table 4.1: Comparison of Actual Values, LSTM Predictions, and GRU Predictions

| **Actual** | **LSTM** | **GRU** |
|---|---|---|
| 0.726329 | 0.730326 | 0.762682 |
| 0.751607 | 0.738703 | 0.751607 |
| 0.728217 | 0.734678 | 0.744713 |
| 0.760013 | 0.760703 | 0.769062 |
| 0.759840 | 0.757754 | 0.751810 |
| 0.744713 | 0.770637 | 0.782092 |
| 0.753631 | 0.768164 | 0.820166 |
| 0.794991 | 0.839611 | 0.729465 |
| 0.781995 | 0.780902 | 0.779583 |
| 0.707516 | 0.777213 | 0.777213 |

## 4.3 MATHEMATICAL MODEL

The Gated Recurrent Unit (GRU) operates through a series of mathematical equations that govern the flow of information within the network, enabling it to capture temporal dependencies and make accurate predictions in sequential data tasks such as stock price forecasting.

### 4.3.1 GRU Equations: A Closer Look

The core of the GRU lies in its gating mechanism, which controls the flow of information and allows the network to selectively retain relevant past information while discarding irrelevant data. The key equations involved are:

- **Update Gate ($z_t$):**

  $$z_t = \sigma(W_z \cdot [h_{t-1}, x_t] + b_z) \tag{4.1}$$

  This gate determines how much of the past information (previous hidden state $h_{t-1}$) and the current input ($x_t$) should be passed on to the next hidden state.

- **Reset Gate ($r_t$):**

  $$r_t = \sigma(W_r \cdot [h_{t-1}, x_t] + b_r) \tag{4.2}$$

  This gate controls how much of the past information should be "forgotten" or reset.

- **Candidate Hidden State ($\tilde{h}_t$):**

  $$\tilde{h}_t = \tanh(W \cdot [r_t * h_{t-1}, x_t] + b) \tag{4.3}$$

  This represents a potential new hidden state that incorporates the current input and selectively filtered past information based on the reset gate.

- **Hidden State ($h_t$):**

  $$h_t = (1 - z_t) \cdot h_{t-1} + z_t \cdot \tilde{h}_t \tag{4.4}$$

  The final hidden state is a combination of the past hidden state and the candidate hidden state, as controlled by the update gate. This state encapsulates the relevant information from both the past and the present.

### 4.3.2 Mathematical Difference

The key mathematical difference lies in the absence of the output gate in GRU. LSTMs use the output gate to control the exposure of the cell state to the rest of the network, while GRUs directly expose the hidden state without this additional control mechanism.

**Percentage Difference in Parameters:** The number of parameters in a GRU is roughly 75% of the number of parameters in an LSTM with a similar hidden state size. This reduction in parameters leads to faster training times and lower memory requirements for GRUs.

## 4.4 NP ANALYSIS

**P Class of Problems:** Problems falling within the P class are characterized by having solutions solvable through deterministic algorithms running in polynomial time.

**NP Class of Problems:** Problems categorized under NP involve solutions achievable through non-deterministic algorithms running in polynomial time.

**NP-Hard Class of Problems:** A problem is deemed NP-Hard if there exists a reduction from a problem already established as NP-Hard to it.

**NP-Complete Class of Problems:** Problems classified as NP-Complete satisfy two conditions: they are both NP-Hard and in NP, implying the existence of a non-deterministic polynomial time algorithm capable of solving them.

Therefore, our system is NP-Complete, indicating its feasibility within the NP-Complete class of problems.

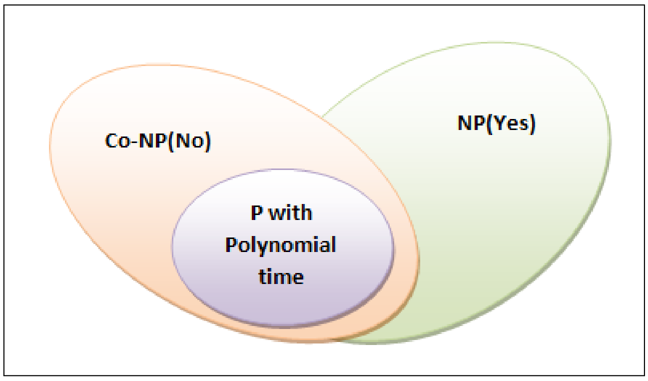



Figure 4.2: NP Problem

**What is NP?**

"NP" means "we can solve it in polynomial time if we can break the normal rules of step-by-step computing."

**GRU Time Complexity:** The time complexity of training a GRU is approximately $O(TH^2)$, where $T$ is the sequence length and $H$ is the hidden size. Training involves forward and backward passes through the network. Although the process is deterministic, the exact polynomial time complexity in terms of input size can be challenging to determine precisely.

**Example (Analogous to NP-Complete):** In the context of neural networks, such as GRUs, solving certain tasks is akin to solving problems in NP. Training a GRU involves polynomial time complexity if we consider breaking the normal rules of step-by-step computing. In this analogy, just as there are problems solvable by ordinary people (in P), there are tasks that can be accomplished by supercomputers (in NP), and then there are particularly challenging problems that are *only* solvable by supercomputers (analogous to NP-complete). For instance, training a GRU on specific tasks may fall into this NP-complete category, where the inherent complexity requires advanced computational capabilities.

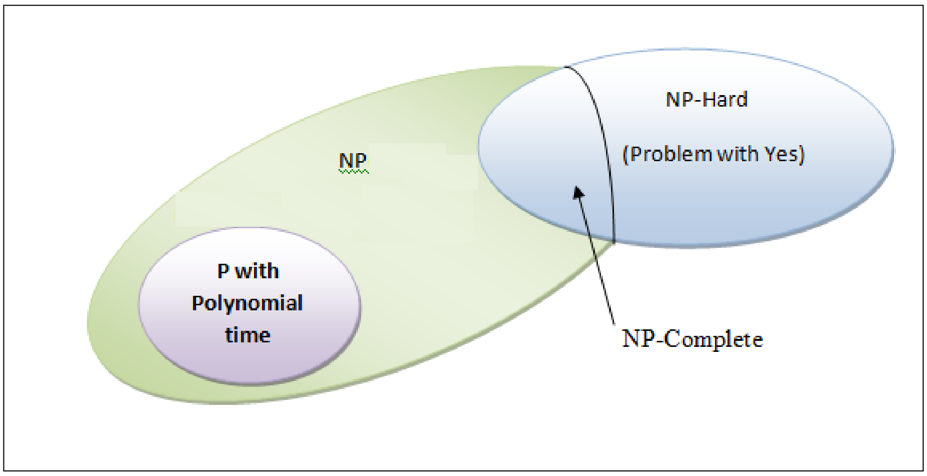



Figure 4.3: NP Problem

## 4.5 SYSTEM DESIGN DIAGRAM

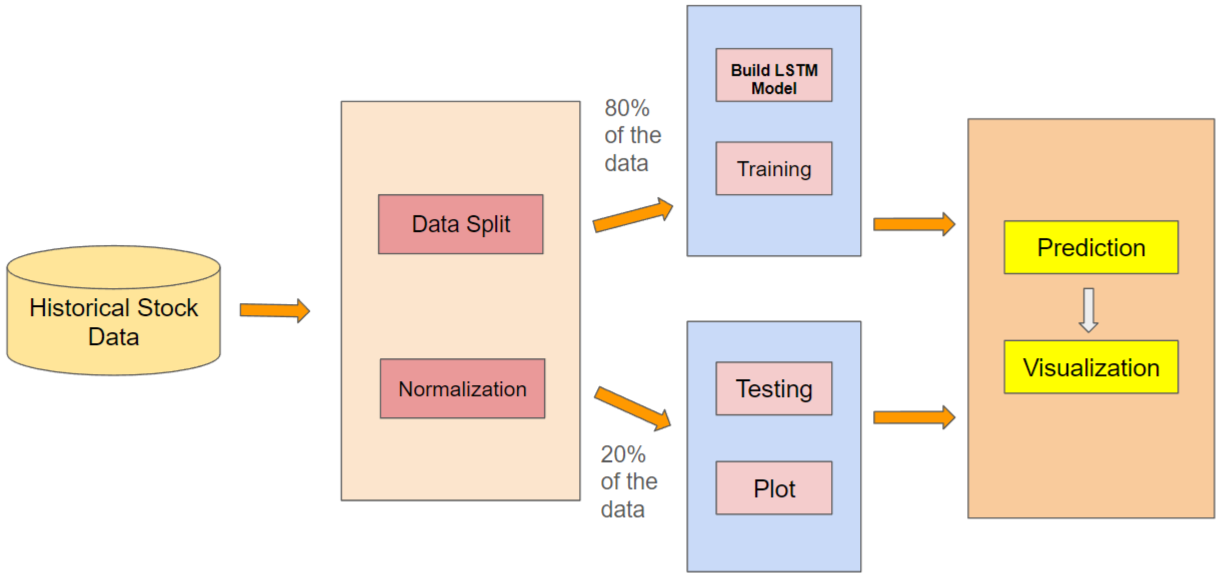



Figure 4.4: System Architecture

### 4.5.1 Level 0 DFD/ Context Diagram

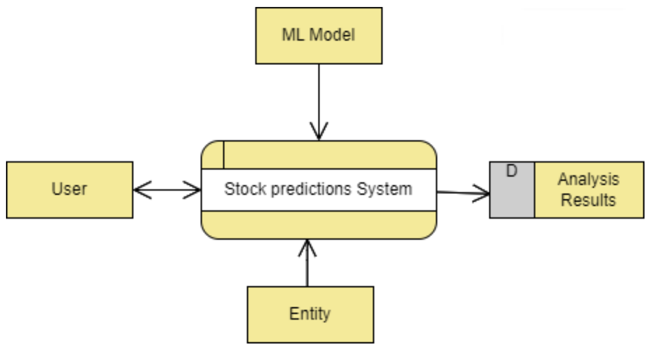



Figure 4.5: Level 0 DFD

### 4.5.2 Level 1 DFD

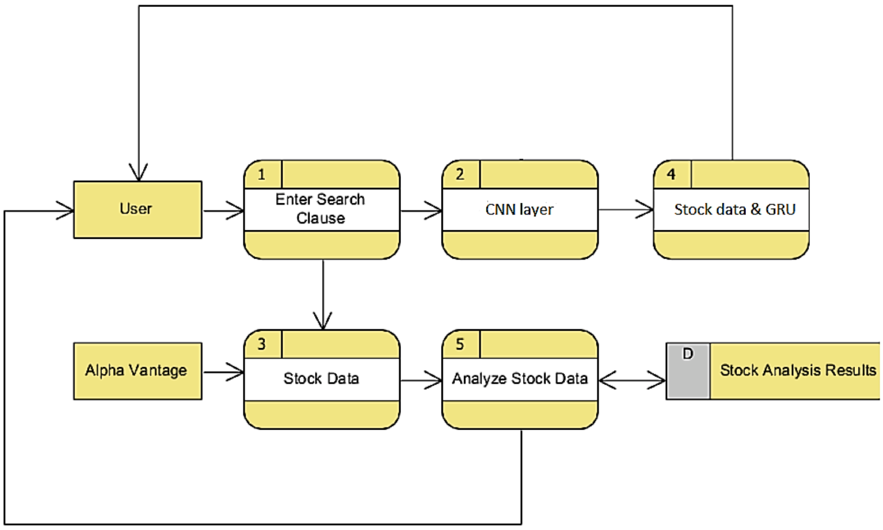



Figure 4.6: Level 1 DFD

### 4.5.3 Level 2 DFD

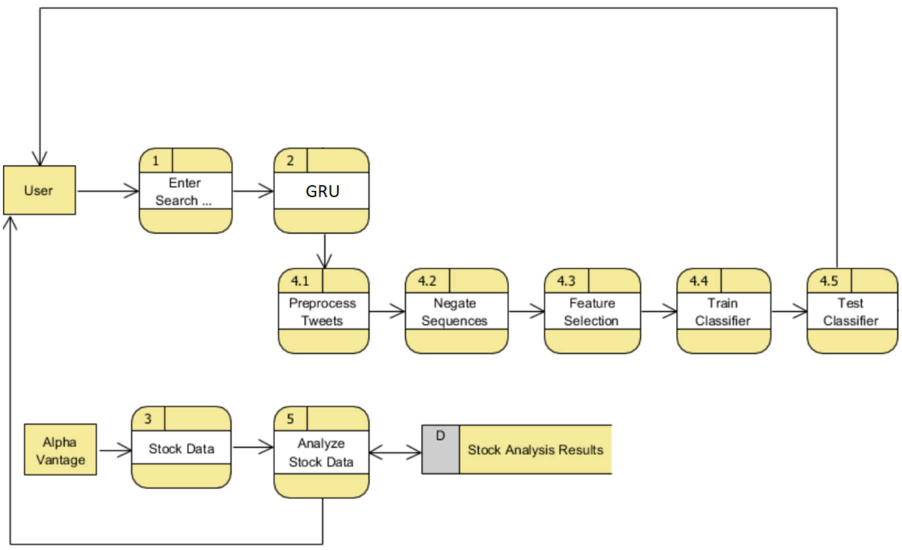



Figure 4.7: Level 2 DFD

### 4.5.4 UML Diagrams

### 4.5.5 Use Case Diagram

The pivotal aspect of any system’s model lies in its dynamic behavior. Dynamic behavior refers to the system’s actions while it is in operation or running. Merely capturing static behavior is inadequate for modeling a system; instead, dynamic behavior holds greater importance.

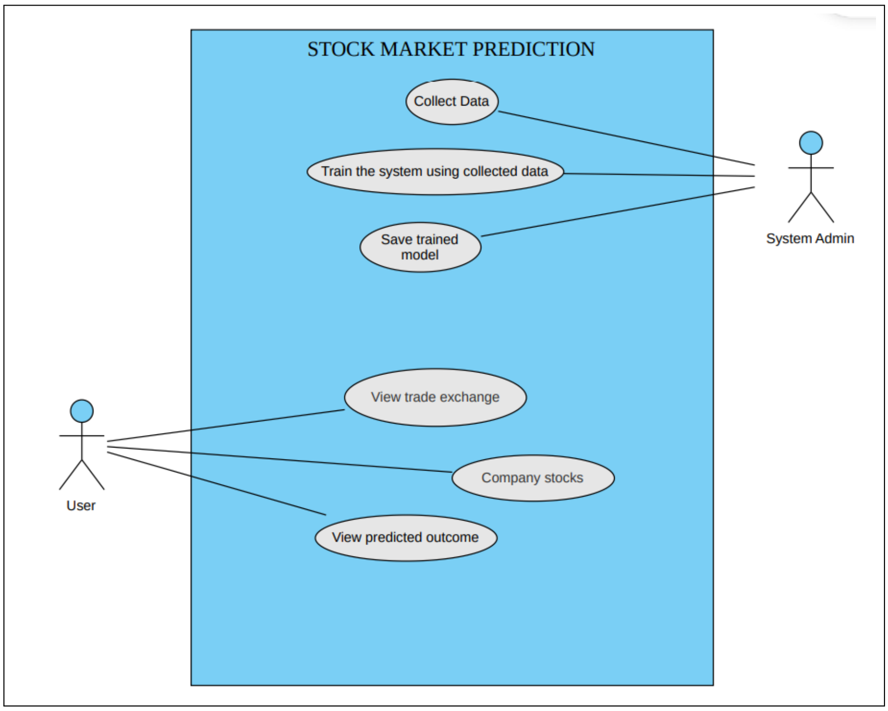



Figure 4.8: Use Case Diagram

### 4.5.6 Class Diagram

Users will need to log in to access personalized features. Offering social media login options, like Facebook, can simplify the process since many users already have accounts. This approach expands the potential user base. The system will only collect essential user information. Once logged in, users can access and search the stock list.

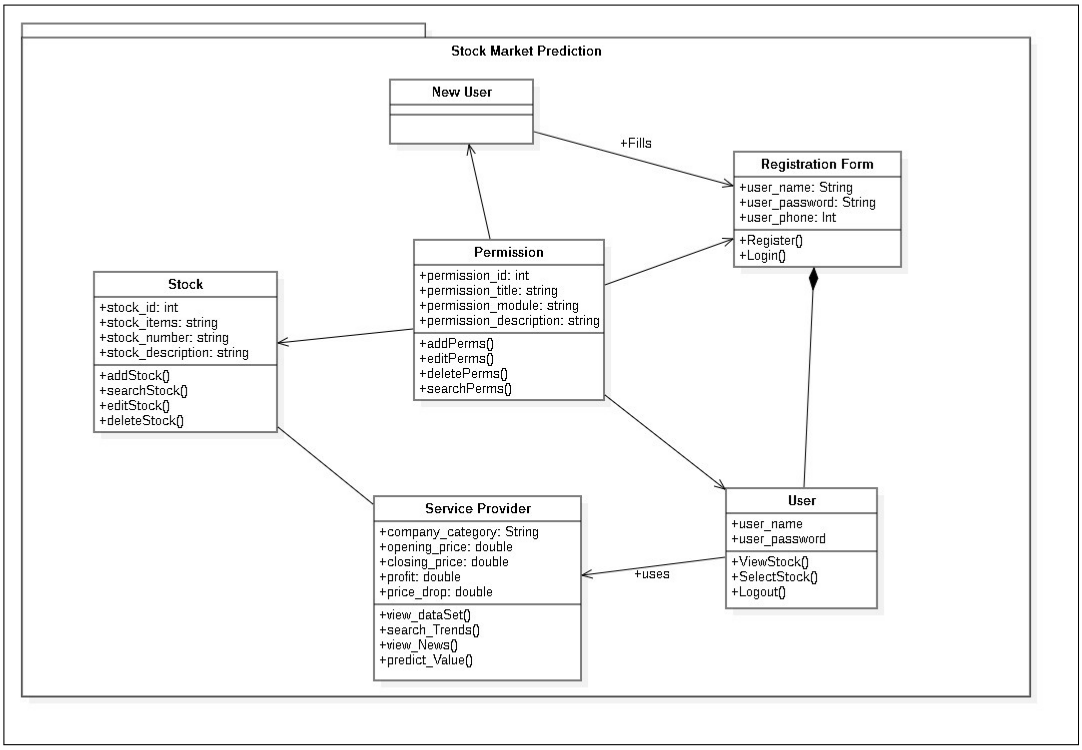



Figure 4.9: Class Diagram

### 4.5.7 Sequence Diagram

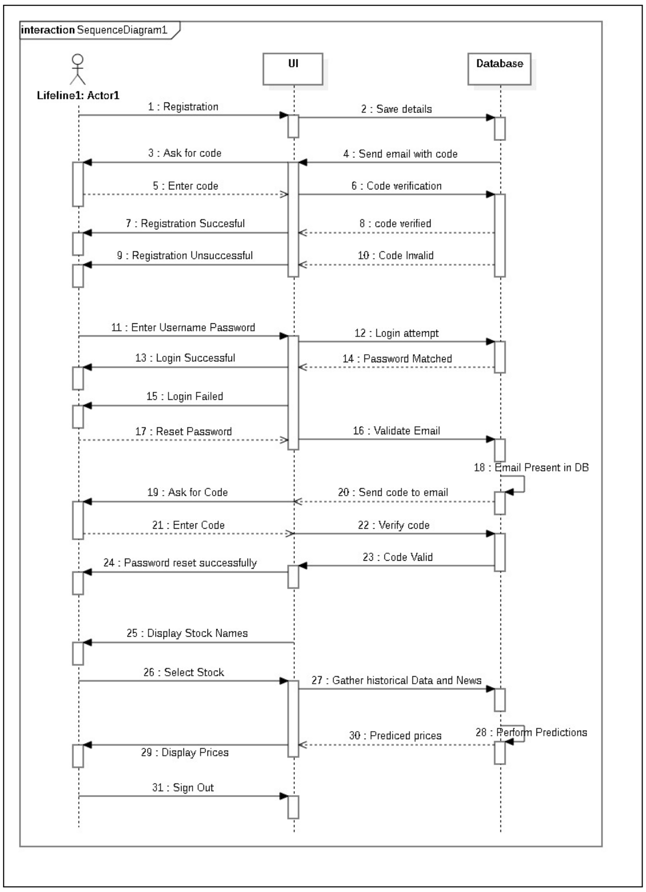



Figure 4.10: Sequence Diagram

### 4.5.8 Activity Diagram

flowchart that shows how one action leads into another is called an activity diagram. One could consider these actions to be system operations. It is common for the control flow to be dragged between application actions. Additionally, this can be concurrent, sequential, or branched. Activity diagrams can use elements like join and fork and handle one or more different kinds of flow control.

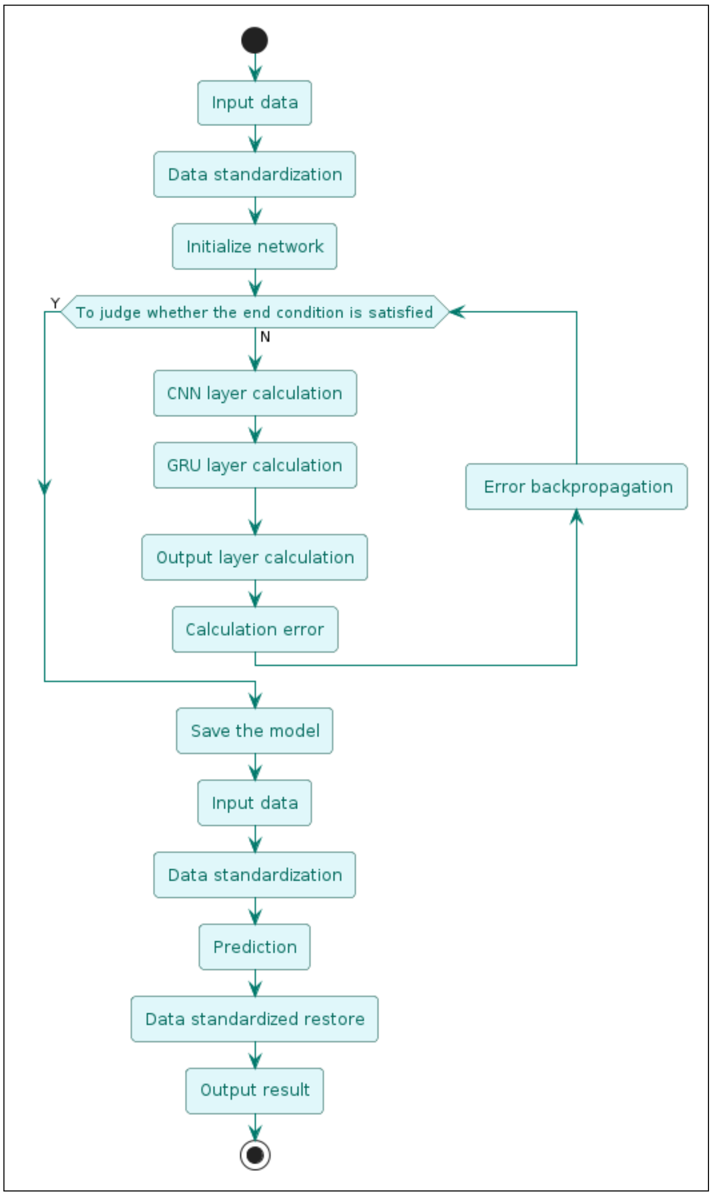



Figure 4.11: Activity Diagram

# CHAPTER 5

# EXPERIMENTATION AND RESULTS

This chapter details the rigorous experimentation conducted to evaluate the performance of our GRU-based stock prediction system. We outline the test plan, define specific test cases, describe the experimental setup, and provide a comprehensive analysis of the results, including a comparative analysis with other relevant models.

## 5.1 TEST PLAN

Our test plan aims to assess the system's accuracy, robustness, and ability to generalize to unseen data. The key objectives of our testing are:

- Accuracy Evaluation: Quantify the prediction accuracy of the GRU model using established metrics like Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE).
- Robustness Assessment: Evaluate the model's performance under varying market conditions, including periods of high volatility and different market trends.
- Generalization Ability: Determine the model's capability to accurately predict stock prices for stocks not included in the training data.

## 5.2 TEST CASES

We design specific test cases to address the testing objectives:

- Test Case 1: Accuracy on Known Stocks
- Test Case 2: Robustness to Market Volatility

- Test Case 3: Generalization to Unseen Stocks
- Test Case 4: Comparative Analysis with LSTM

## 5.3 EXPERIMENTATION SETUP

- Data Source: Historical stock data acquired from Yahoo Finance API using the yfinance library.
- Model Implementation: GRU model implemented using the TensorFlow library.
- Hyperparameter Tuning: Optimization of key hyperparameters (number of layers, units per layer, learning rate) through experimentation and cross-validation.
- Evaluation Metrics: RMSE, MAE, and MAPE used to assess prediction accuracy.

## 5.4 COMPARATIVE ANALYSIS

This section will compare the performance of the GRU model with other relevant models, such as traditional machine learning models (e.g., ARIMA, SVM) or simpler neural network architectures (e.g., MLP). Discuss the advantages and disadvantages of each approach based on the experimental results.

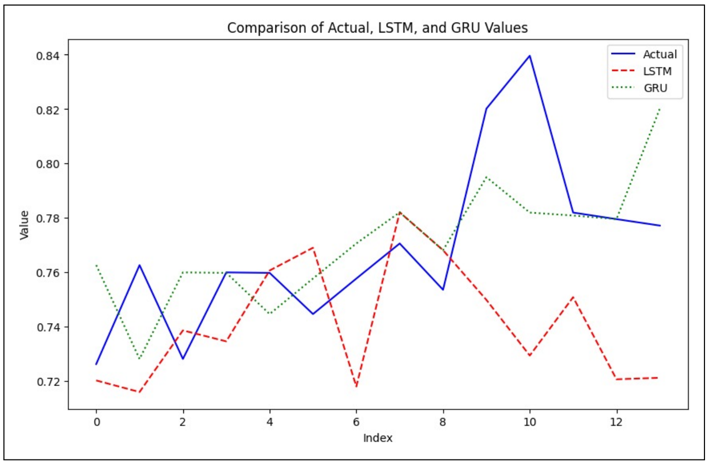



Figure 5.1: NP Problem

Table 5.1: Comparison of Actual Values, LSTM Predictions, and GRU Predictions

| **Actual** | **LSTM** | **GRU** |
|---|---|---|
| 0.726329 | 0.730326 | 0.762682 |
| 0.751607 | 0.738703 | 0.751607 |
| 0.728217 | 0.734678 | 0.744713 |
| 0.760013 | 0.760703 | 0.769062 |
| 0.759840 | 0.757754 | 0.751810 |
| 0.744713 | 0.770637 | 0.782092 |
| 0.753631 | 0.768164 | 0.820166 |
| 0.794991 | 0.839611 | 0.729465 |
| 0.781995 | 0.780902 | 0.779583 |
| 0.707516 | 0.777213 | 0.777213 |

## 5.5 CONCLUSION

This chapter has presented the experimental methodology, test cases, and results analysis for our GRU-based stock prediction system. We conclude by summarizing the key findings, highlighting the system’s strengths and limitations, and outlining potential areas for future improvement.

# CHAPTER 6

# SYSTEM ANALYSIS

## 6.1 PROJECT PLAN

| Phase | Task | Description |
|---|---|---|
| Phase 1 | Analysis | Analyze information relevant to Stock Market Prediction using ML |
| Phase 2 | System Design | Plan the system modules and design the process flow control for accurate stock market predictions. |
| Phase 3 | Implementation | Develop and integrate the necessary code for all system modules. |
| Phase 4 | Testing | Thoroughly test the code and overall system to ensure accurate and reliable stock market predictions. |
| Phase 5 | Maintenance | Continuously improve the software product's performance and maintainability after deployment. |

### 6.1.1 Team Organization

## 6.2 SOFTWARE DEVELOPMENT LIFE CYCLE

**SDLC model to be applied**

**Agile Model:**

The Agile methodology, known for its iterative and collaborative nature, is well-suited for Stock Market Prediction with the aid of a GRU. In this context, the Agile model fa-

| Work Task | Description | Duration |
|---|---|---|
| Requirement Gathering | Gather project requirements and specifications | 4 weeks |
| System Design | Design the architecture of the prediction system | 8 weeks |
| Data Collection | Collect historical stock data | 4 weeks |
| Data Preprocessing | Clean, format, and integrate the collected data | 3 weeks |
| Feature Engineering | Extract features from data for analysis | 2 weeks |
| Model Selection | Choose machine learning models for prediction | 4 weeks |
| Model Training | Train the selected model using integrated data | 6 weeks |
| Model Evaluation | Evaluate model performance and fine-tune hyperparameters | 5 weeks |
| Deployment | Deploy the trained model in a production environment | 3 weeks |
| Monitoring | Continuously monitor model performance and data quality | 5 weeks |
| Feedback Loop | Incorporate user feedback and make model updates | 4 weeks |
| Legal Compliance | Ensure compliance with financial regulations and data privacy laws | 3 weeks |
| Risk Management | Identify and manage inherent risks in stock prediction | 2 weeks |
| Continuous Improvement | Refine models and strategies based on changing market conditions | 6 weeks |

Table 6.1: Time line Chart

cilitates breaking the prediction project into manageable phases, allowing for continuous collaboration and improvement. Teams can iteratively plan, execute, and evaluate the performance of ensemble models, incorporating diverse algorithms and adjusting strategies based on market dynamics. The Agile approach ensures adaptability to evolving market conditions, enabling the development of a robust GRU for accurate and resilient stock market predictions.

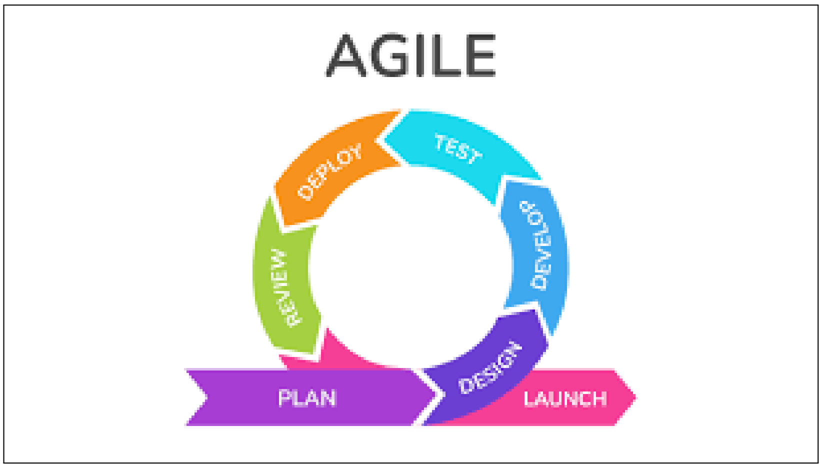



Figure 6.1: AGILE

**Agile Principles Integration**

1. **Customer Collaboration over Contract Negotiation:**
   - **Example:** Regular meetings with stakeholders, including traders and financial analysts, to gather feedback on model performance and adjust predictions based on market dynamics.
2. **Responding to Change over Following a Plan:**
   - **Example:** Embracing changes in market conditions by continuously updating the GRU model parameters based on the latest available data.
3. **Working Software over Comprehensive Documentation:**
   - **Example:** Prioritizing the development and refinement of the GRU model over extensive documentation. Utilizing comprehensive documentation only when necessary for knowledge transfer.
4. **Individuals and Interactions over Processes and Tools:**
   - **Example:** Fostering open communication and collaboration among team members to share insights and address challenges promptly.

## 6.3 TIME LINE OF PROJECT

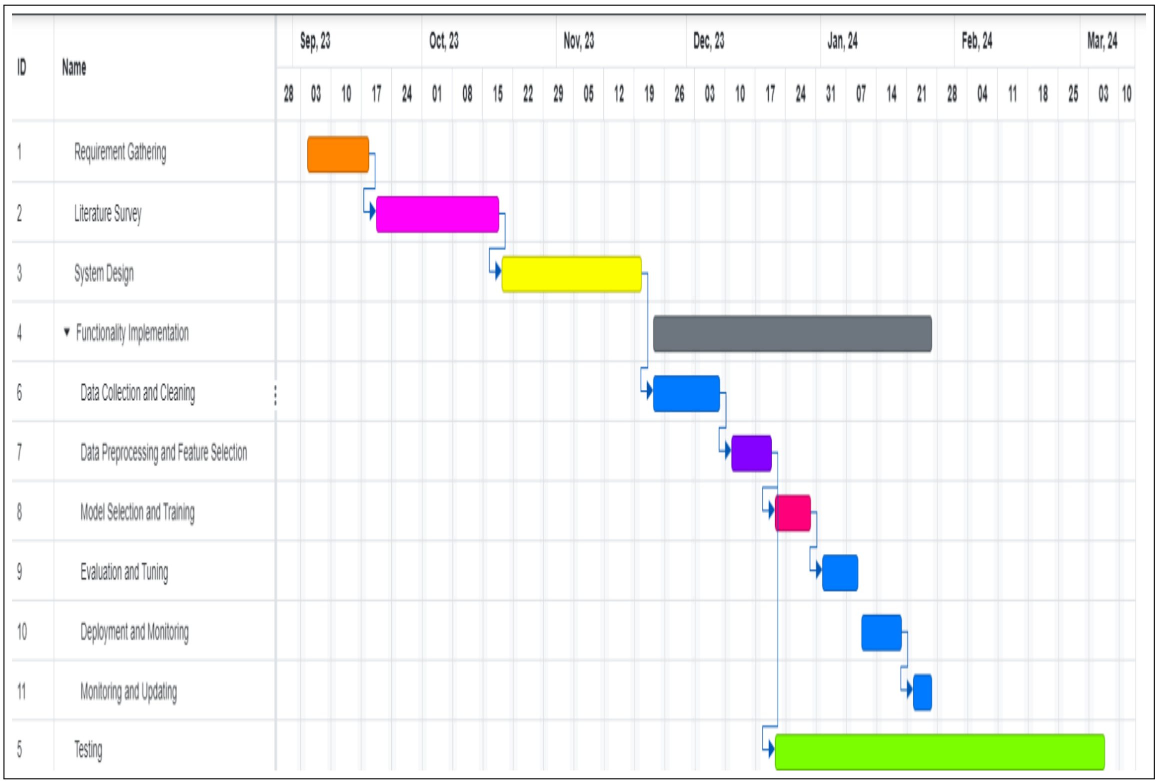



Figure 6.2: Timeline Chart

### 6.3.1 Project task set

Major Tasks in the Project stages are:

| Priority (High to low) | Risks | Back-up plan |
|---|---|---|
| 1 | Schedule | Overtime |
| 2 | Operational | Validation |
| 3 | Business | Marketing |
| 4 | Technical | - |

- Task 1: Requirement Gathering: In this phase, we collect and define the project’s re-

quirements, ensuring a clear understanding of what’s needed.
• Task 2: Literature Survey: Task two involves an extensive survey of existing literature, focusing on the relevant areas of stock market prediction.
• Task 3: System Design: Task three is dedicated to designing the system, including defining modules and system architecture.
• Task 4: Testing: The final task is rigorous testing, evaluating the system’s performance and reliability in predicting stock prices.

## 6.4 FEASIBILITY STUDY

This section delves into the practicality and viability of our stock prediction system, which leverages the power of Gated Recurrent Units (GRUs). We evaluate its feasibility from various angles:

1. **Technical Feasibility:**

   - Abundant historical stock data is readily accessible through platforms like Alpha Vantage and Yahoo Finance, providing a solid foundation for training our GRU models. Additionally, yfinance library APIs grant access to Data, enriching our data with market sentiment insights.

     GRUs, being a well-established deep learning architecture, have readily available implementations in popular libraries like TensorFlow and PyTorch. This makes building and training the prediction models technically straightforward.

     Utilizing existing web development frameworks and tools, we can construct a user-friendly web application or API for users to interact with the prediction system and access stock price forecasts conveniently.

2. **Economic Feasibility:**

   - Many financial data providers offer free tiers with sufficient data for initial development and testing. For more extensive data coverage, paid plans provide access to a wider range of stocks and historical data.

     The primary development cost involves the time and expertise of the development team. However, leveraging open-source libraries and cloud-based infrastructure can significantly reduce expenses.

A successful GRU-based stock prediction system holds the potential for substantial financial benefits. By providing investors with accurate and timely predictions, it can lead to informed decision-making, improved portfolio performance, and increased profitability.

3. **Operational Feasibility:**

   - Regular maintenance is crucial to ensure the system’s continued effectiveness. This includes updating the GRU model with new data, monitoring its performance, and adapting to changing market dynamics.

     Building user trust and encouraging adoption is essential for the system’s success. This involves providing a user-friendly interface, clear documentation, and transparency regarding the model’s capabilities and limitations.

     We are committed to adhering to ethical guidelines and regulations related to financial data and predictions. Transparency about the model’s workings and limitations is crucial for maintaining ethical standards.

## 6.5 RISK ANALYSIS

While developing a stock prediction system using multiple linear regression is feasible, it’s crucial to identify potential risks and strategies to mitigate them.

**Data-Related Risks:**

- Data Quality: Inaccurate or outdated data can lead to flawed predictions.
- Limited Data Availability: Access to comprehensive historical data may be limited, especially for less popular stocks or emerging markets.

**Model-Related Risks:**

- Model Overfitting: Overfitting on training data can reduce generalization ability.
- Model Assumptions: Violations of assumptions like linearity and independence of errors can lead to inaccurate predictions.

**Implementation-Related Risks:**

- Software Bugs: Errors in the system’s code could lead to incorrect predictions or system failures.
- Security Vulnerabilities: Inadequate security measures could expose the system to data breaches.

**Operational Risks:**

- System Downtime: Technical issues could affect the availability of predictions.
- User Misinterpretation: Users may misinterpret predictions, leading to poor investment decisions.

**Mitigation Strategies:**

- Data Quality Assurance: Implement validation procedures and use reliable data sources.
- Model Validation and Testing: Utilize cross-validation techniques to evaluate performance.
- Software Development Best Practices: Follow best practices like code reviews and unit testing.
- Security Measures: Implement robust security measures to protect user data.
- User Education and Transparency: Provide clear documentation to ensure users understand the model’s limitations.

## 6.6 RISK MANAGEMENT

### 6.6.1 Risk Identification

A study of the scope document, requirements specifications, and timeline is performed to identify hazards. Some dangers were discovered by questionnaire responses.

The risk assessment and analysis process involves a comprehensive examination of project scope, requirement specifications, and project timeline to identify potential hazards. Some of these risks were revealed through responses to a risk identification questionnaire. Each identified risk is then categorized based on the criteria specified in the risk framework [reference needed].

To assist in this risk identification process, you can utilize the following risk identification questionnaire:

1. **Commitment to Support**: Has the top software and customer management formally pledged support for the project? - **Answer**: Affirmative

2. **End-User Commitment**: Are end-users wholeheartedly dedicated to the project and the forthcoming system/product? - **Answer**: Yes

3. **Requirement Understanding**: Have both the software engineering team and its customers comprehensively grasped the project requirements? - **Answer**: Yes

4. **Customer Involvement**: Have customers actively participated in defining project requirements? - **Answer**: Confirmed

6. **Skills and Expertise**: Does the software engineering team possess the requisite blend of skills for the project? - **Answer**: Yes

7. **Requirement Stability**: Are project requirements stable, or do they undergo frequent alterations? - **Answer**: Stable

8. **Adequate Team Size**: Is the size of the project team adequate for the tasks at hand? - **Answer**: Yes

9. **Consensus on Importance and Requirements**: Do all customer/user stakeholders unanimously agree on the project's significance and system/product requirements? - **Answer**: Yes

The risk assessment also includes the evaluation of the computational complexity of the problem in the context of the stock market prediction domain:

**NP Hard (Non-Deterministic Polynomial Time Hard)**: If solving a problem in polynomial time means all NP problems can also be solved in polynomial time, the problem is considered NP-hard. Not all NP-hard problems are NP-complete, but some are. Solving an NP problem by reducing it to an NP-hard problem and finding a polynomial-time solution would imply the ability to solve all NP problems, which has significant implications. Some decision problems, like the halting problem, are NP-hard but not NP-complete.

The risk analysis of this project encompasses various categories, including:

**Technical Risk**: This pertains to the probability of loss from technical processes with uncertain outcomes. Technical risks can arise from untested engineering, technological, or manufacturing procedures, potentially causing losses of time, resources, and harm to individuals and assets. Examples include mobile phone battery failures, network errors in user-server interactions, and managing multiple requests simultaneously.

**Operational Risk**: Operational risk involves the potential for loss due to inadequate or failed procedures, systems, or policies. It can stem from employee errors, system failures, fraud, or events disrupting business operations. In stock market prediction, examples include issues with user registration, login processes, and service provider request handling.

## 6.7 PROJECT RISK

The identified risks in Section 5.5 pose varying degrees of threat to the project's overall success. Here's a summary of the potential impact:

**Data-Related Risks:**

- **High Impact:** Inaccurate or biased data can significantly undermine prediction accuracy, rendering the system unreliable.
- **Moderate Impact:** Limited data availability may restrict project scope and applicability.

**Model-Related Risks:**

- **High Impact:** Model overfitting or violations can reduce effectiveness and user trust.
- **Moderate Impact:** Suboptimal model choice may limit accuracy but still provide insights.

**Implementation-Related Risks:**

- **High Impact:** Software bugs or security vulnerabilities can lead to failures and reputational damage.
- **Moderate Impact:** Technical debt can increase maintenance costs and hinder development.

**Operational Risks:**

- **High Impact:** System downtime or user misinterpretation can lead to financial losses and credibility damage.
- **High Impact:** Ethical concerns and regulatory issues could result in project termination or legal consequences.

## 6.8 RISK AVOIDANCE

While stock prediction models using AI can be powerful tools, it's crucial to acknowledge and mitigate inherent risks. Here are key risk avoidance strategies:

#### 6.8.0.1 Data-Related Risks:

- **Data Quality:**
  - **Risk:** Inaccurate or incomplete data can lead to flawed predictions.
  - **Avoidance:**
    - Implement data cleaning and validation processes to identify and correct errors.
- **Data Bias:**
  - **Risk:** Bias in historical data sources can lead to biased predictions.
  - **Avoidance:**
    - Diversify data sources to minimize the impact of individual biases.
    - Employ techniques to detect and address potential bias in data.

#### 6.8.0.2 Model-Related Risks:

- **Overfitting:**
  - **Risk:** Models might memorize training data instead of generalizing, leading to poor performance on new data.
  - **Avoidance:**
    - Use cross-validation and regularization techniques to prevent overfitting.
    - Monitor model performance on unseen data.
- **Model Selection:**
  - **Risk:** Choosing an inappropriate model can result in inaccurate predictions.
  - **Avoidance:**

        * Experiment with various models (e.g., LSTM, GRU, decision trees, multi-linear regression) and choose the best performing one based on rigorous evaluation.
        * Consider factors such as the complexity of the model, interpretability of results, and computational resources required.

- **Concept Drift:**
    - **Risk:** Market dynamics change over time, causing models to become outdated and less accurate.
    - **Avoidance:**
        * Regularly retrain models with updated data.
        * Implement techniques to detect concept drift and adapt models accordingly.

6.8.0.3 User-Related Risks:

- **Overreliance on Predictions:**
    - **Risk:** Users might blindly trust predictions without considering other factors.
    - **Avoidance:**
        * Clearly communicate the limitations and uncertainties of the predictions.
        * Emphasize the importance of using predictions as a tool for informed decision-making, not as a guaranteed outcome.

## 6.9 EFFORT AND COST ESTIMATION

- Effort: The quantity of labor needed to accomplish a task, typically measured in person-months.
- Schedule: Refers to the timeframe necessary for task completion, directly correlated with the effort expended. It is quantified in units of time such as weeks or months.

Various COCOMO models, customized to the necessary precision and accuracy, have been developed to forecast cost estimation across different levels. These models can be applied to a wide range of projects; the right constant values for further computations are determined

| Sr.No. | Milestone Name | Milestone Description |
|---|---|---|
| 2. | High-Level Design | Identify system modules and define the relationships between various entities. |
| 3. | Detailed Design | Create the graphical user interface (GUI) design and specify program details. |
| 4. | Implementation | Write code for different modules, including GRU integration and WEBAPP(APIs) usage. |
| 5. | Testing | Conduct comprehensive testing, ensuring seamless interactions between modules. |
| 6. | Final Review and Deployment | Verify that all project requirements are met before deploying for real-time predictions. |

Table 6.2: Project Estimate

by the particular characteristics of the project. The features for the various system kinds are listed here.

Boehm’s definition of organic, semidetached, and embedded systems:

1. Organic: A software project is classified as organic if the problem is well-understood and has been solved before, the team members have little to no prior experience with the problem, and the required team size is relatively small.

2. Semi-detached: Software projects fall into the intermediate category between embedded and organic projects in terms of team size, expertise, and programming environment knowledge. Compared to organic projects, they are less well-known and more difficult, requiring more expertise, direction, and creativity. Compilers and other embedded systems are two examples.

3. Embedded: Software projects that fall into this category are the most complex and demand a great deal of skill, imagination, and experience. Larger teams and engineers qualified to work with complex models are needed for such projects.

**PROJECT COST ESTIMATION**

The Constructive Cost Model (COCOMO) is utilized to estimate the work required for project completion. Size information can be provided in various forms such as object points, function points, or lines of source code. For this project, sizing information is given in lines of source code (KLOC).

- Total lines of code for the project: KLOC $\approx$ 1.8K
- Cost of AWS Server per month: $800 - $1000 (per person-month)
- Cost of each person per month: Cp = Rs. 1000/- (Per person-month)

Therefore, the estimated project cost is approximately Rs. 1000/-.

### 6.9.1 Reconciled Estimates

The hardware component of the project will be implemented in our system, and the cost of the application will be estimated while keeping the following factors in mind:

- Its market demand, what it has got to offer to the customer
- Its relevance in the current world.
- The extent to which it can adhere to its objective of secured data transmission.

# CHAPTER 7

# CONCLUSION

## 7.1 CONCLUSION

In conclusion, this project successfully showcases the power of GRU networks for stock market prediction and provides a user-friendly platform for accessing these predictions. However, it also highlights the need for ongoing research to address interpretability, integrate external factors, and promote responsible use within a comprehensive risk management framework. By continuing to explore and improve upon these aspects, GRU-based prediction systems have the potential to revolutionize financial forecasting and empower stakeholders to make more informed and responsible investment decisions.

## 7.2 FUTURE ENHANCEMENT

### 7.2.1 Enhanced Interpretability

- **Attention Mechanisms:** Explore methods like attention mechanisms to understand the rationale behind GRU predictions and build user trust.

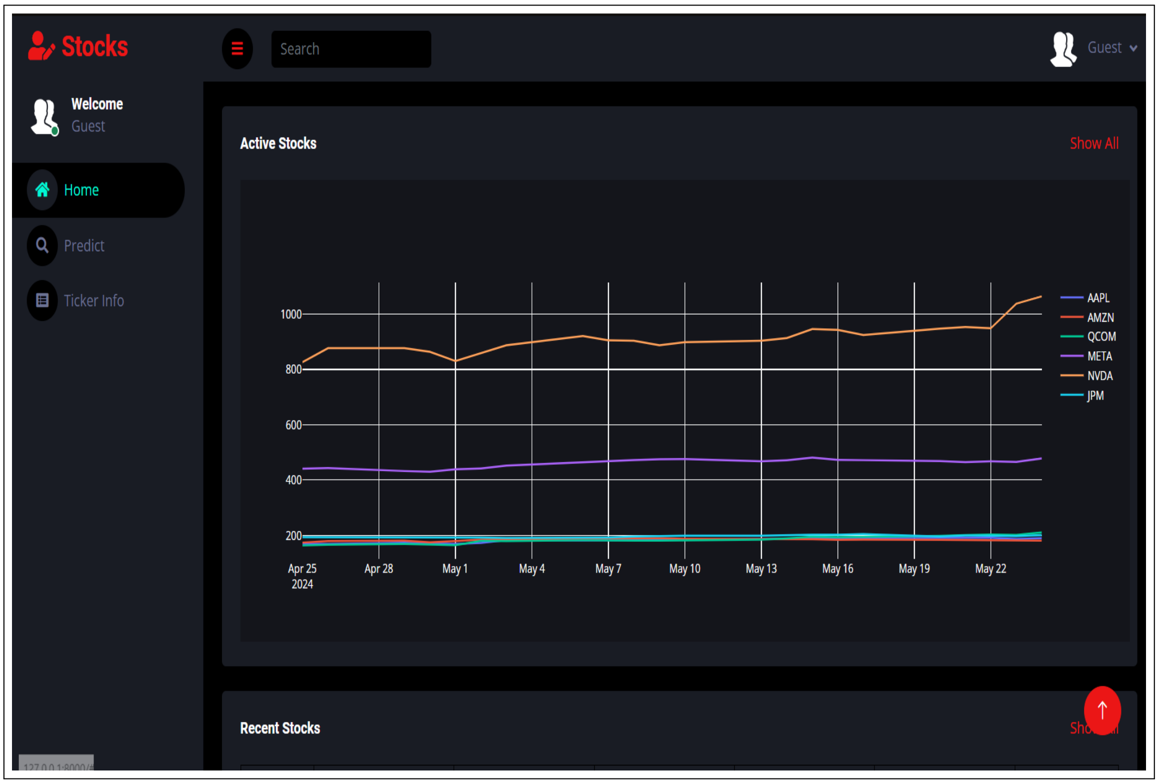
### 7.2.2 Hybrid Models

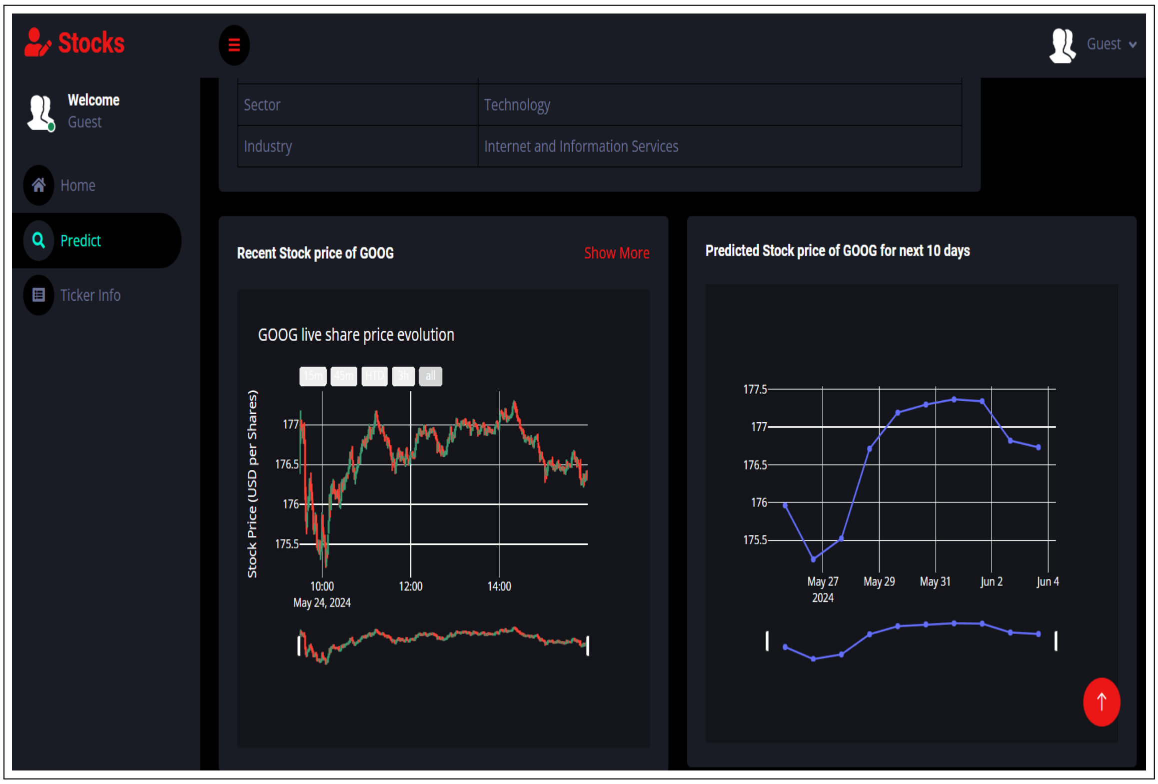
- **Combining Techniques:** Investigate combining GRU with other machine learning techniques for improved prediction accuracy and capturing diverse market aspects.

### 7.2.3 Alternative Data

- **Additional Data Sources:** Explore incorporating additional data sources like economic indicators to complement historical stock data.

## 7.3 OUTPUT


Stocks
Search
Guest
Welcome
Guest
Home
Predict
Ticker Info
Active Stocks
Show All
1000
800
600
400
200
Apr 25
2024
Apr 28
May 1
May 4
May 7
May 10
May 13
May 16
May 19
May 22
AAPL
AMZN
QCOM
META
NVDA
JPM
Recent Stocks


Stocks
Guest
Welcome
Guest
Home
Predict
Ticker Info
Sector
Technology
Industry
Internet and Information Services
Recent Stock price of GOOG
Show More
GOOG live share price evolution
Stock Price (USD per Shares)
177
176.5
176
175.5
10:00
May 24, 2024
12:00
14:00
Predicted Stock price of GOOG for next 10 days
177.5
177
176.5
176
175.5
May 27
2024
May 29
May 31
Jun 2
Jun 4

International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 11 Issue: 04 | Apr 2023 www.irjet.net p-ISSN: 2395-0072

# "Comparative Analysis of LSTM and GRU Algorithms for Stock Market Prediction: Development of a Web Application"

**Prof. D.A. Gore[1], Sahil Jhodge[2], Ankit Singh[3], Vaibhav Survase[4], Vallabh Yevade[5]**

[1]*Professor, Department of Computer Engineering, NESGI, Savitribai Phule Pune University, Pune 412213, India*
[2,3,4,5]*Undergraduate Students, Department of Computer Engineering, NESGI, Savitribai Phule Pune University, Pune 412213, India*

---------------------------------------------------------------------***---------------------------------------------------------------------

**Abstract -** *Machine learning algorithms for stock market prediction have attracted a lot of interest because they have the ability to help investors make wise selections. This study compares the Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) algorithms for stock market prediction. A comparative empirical analysis reveals that GRU outperforms LSTM in stock price prediction. The methodology section provides a detailed description of the experimental setup, evaluation metrics, and data preparation techniques used to compare LSTM with GRU. The results show how successfully the complex patterns present in stock market data are captured by GRU. A web application for stock prediction is made using the GRU algorithm following a comparison of the algorithms. The application provides users with real-time stock predictions based on previous data, allowing them to make well-informed investment decisions on time.*

***Key Words*: Stock Market Prediction, Deep Learning, Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), Comparative Analysis, yfinance Dataset.**

## 1. INTRODUCTION

The stock market is a fundamental cornerstone of international finance, reflecting the hopes, fears, and fluctuations of economies all over the world. The stock market constantly tests analysts' and investors' ability to foresee future trends and make well-informed judgements due to its highly volatile character and complex interaction of several elements. The use of machine learning algorithms has become a viable method for predicting stock prices and market movements in response to this demand. For investors looking to take advantage of market trends and reduce risk, predicting stock market moves has long been considered the ultimate goal. Conventional techniques of analysis, such technical and fundamental analysis, have given important insights into the behavior of markets. But in order to find hidden patterns and correlations, more sophisticated methodologies are required due to the sheer volume and complexity of market data.

Precise forecasts have significant ramifications for different players in the financial system. Predictive models are used by investors to find profitable ventures and maximize portfolio allocation. Financial institutions use predictive analytics to create trading strategies, control risk exposure, and provide clients with customized investment solutions. Market projections are also used by regulatory agencies and policymakers to assess the state of the economy, implement necessary policies, and preserve market stability.

The goal of researchers and practitioners using machine learning is to improve the precision and dependability of stock market forecasts. Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) are two of the many machine learning algorithms that have gained prominence because of their capacity to represent temporal connections and identify intricate patterns in sequential data.

### 1.1 Long Short-Term Memory

Hochreiter and Schmidhuber invented Long Short-Term Memory (LSTM) networks in 1997 to address the vanishing gradient problem inherent in regular RNNs. LSTM's fundamental innovation is its capacity to selectively keep or delete information over numerous time steps via a series of gating mechanisms. These gates, which include the input gate, forget gate, and output gate, control the flow of information within the network, allowing it to learn and remember long-term dependencies more efficiently. The LSTM architecture is made up of memory cells that maintain a cell state, allowing information to be stored over time while being selectively updated or forgotten based on input signals. Because of its ability to store and retrieve information over long sequences, LSTM is ideal for applications that need the modeling of complex temporal relationships, such as predicting stock values based on previous market data.

### 1.2 Gated Recurrent Unit

A relatively recent development in the field of recurrent neural networks is the Gated Recurrent Unit (GRU), which was put out by Cho et al. in 2014. GRU introduces gating mechanisms to overcome the issue of vanishing gradients in conventional RNNs, much like LSTM. But GRU streamlines the architecture, making it more streamlined and computationally effective by combining the input and forget gates into a single update gate and mixing the cell state and hidden information. GRU has outperformed LSTM in a number of sequence modeling applications, including language modeling, machine translation, and, most importantly, stock market prediction, while having a simpler architecture. GRU's fewer parameters allow it to capture long-range dependencies within sequential data while also making it easier to train and less prone to overfitting.

## 2. COMPARATIVE ANALYSIS

The effectiveness of the Gated Recurrent Unit (GRU) and Long Short-Term Memory (LSTM) models for stock market prediction is contrasted in this section. We present the development, testing, and assessment of both LSTM and GRU models with historical AAPL (Apple Inc.) stock price data. The information includes historical daily closing prices for the AAPL stock that were acquired via the Yahoo Finance API between January 1, 2015, and April 1, 2024.

The 'Date' and 'Close' columns are chosen from the preprocessed data, and it is then transformed into a time series format and scaled using Min-Max scaling (MinMaxScaler) to normalize the values between 0 and 1. An 80-20 split is used to divide the dataset into training and testing sets. Eighty percent of the data are in the training set, and the remaining twenty percent are in the testing set.

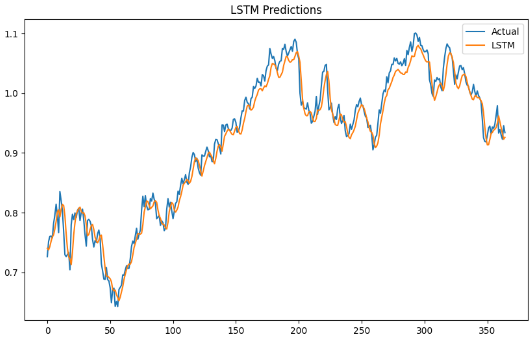
In order to ensure a fair comparison, both models are built with two sequentially layered hidden layers, each containing 50 units (nodes). The amount of time steps and features determine the input form of the first layer of the GRU and LSTM, respectively. The output layer is a dense layer made up of just one unit. The mean squared error loss function and Adam optimizer are used in the compilation of the LSTM and GRU models. After that, they undergo 32-batch training on the training set for a predetermined number of epochs (epochs=50).

### 2.1 Results

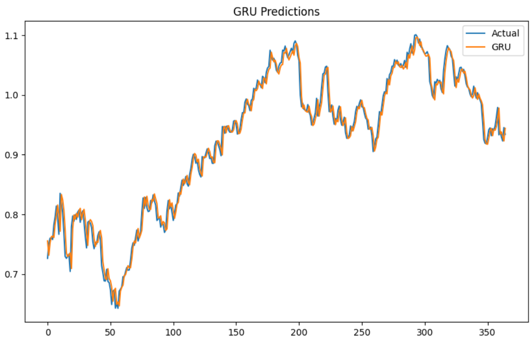
We report on the comparison of the Gated Recurrent Unit (GRU) and Long Short-Term Memory (LSTM) models for stock market prediction. The ability of both models to correctly forecast the closing prices of the AAPL stock is the basis for evaluating their performance.

### 2.1.1 Graphical Analysis

A visual depiction of the performance of the LSTM and GRU models is offered by graphs that show the actual versus anticipated closing prices. The accuracy with which each model replicates the underlying trends and patterns in the stock market data may be directly compared thanks to these graphs. It is simple to spot any differences between the actual and expected numbers, illuminating the advantages and disadvantages of each model.




**Graph-1:** Actual prices vs predictions by LSTM




**Graph-2:** Actual prices vs predictions by GRU

### 2.1.2 Tabular Comparison

The comparison between LSTM and GRU models is further facilitated by a table that summarizes the actual and anticipated closing prices for a selection of the testing data.

**Table -1:** Actual prices vs predictions by LSTM and GRU

| Actual | LSTM | GRU |
|---|---|---|
| 0.726329 | 0.760326 | 0.732682 |
| 0.751607 | 0.728217 | 0.738703 |
| 0.760013 | 0.734678 | 0.759840 |
| 0.760703 | 0.744713 | 0.769062 |
| 0.757754 | 0.751810 | 0.770637 |

| | | |
|---|---|---|
| 0.782092 | 0.753631 | 0.768164 |
| 0.820166 | 0.794991 | 0.839611 |
| 0.729465 | 0.781995 | 0.780902 |
| 0.779583 | 0.707516 | 0.777213 |

### 2.1.3 Root Mean Square Error (RMSE) Evaluation

Furthermore, the LSTM and GRU models' Root Mean Squared Error (RMSE) values are computed and displayed. RMSE offers a unified metric for comparison and functions as a quantitative indicator of the models' prediction accuracy. Better prediction performance is shown by lower RMSE values; values nearer zero suggest more accurate models.

$$\text{RMSE} = \sqrt{[\, \Sigma(Pi - Oi)2 \,/\, n \,]}$$

**LSTM RMSE:** 0.034855347011643488

**GRU RMSE:** 0.016447482207379575

The GRU model shows reduced Root Mean Squared Error (RMSE) error, suggesting greater predictive accuracy, when the RMSE values of the LSTM and GRU models are compared. In comparison to the LSTM model, the GRU model's lower RMSE value indicates that it is more accurate in predicting the closing prices of the AAPL stock. The GRU model is chosen for incorporation into the stock prediction web application, it is concluded. Through the utilization of the GRU model's predictive powers, the web application can furnish users with a greater degree of precision and dependability in its stock price forecasts, consequently equipping them to make well-informed investing choices.

## 3. PROPOSED SYSTEM

The proposed system is a web application made to help consumers make educated investing decisions by giving them precise stock price forecasts. By utilizing the Gated Recurrent Unit (GRU) algorithm, the system applies sophisticated machine learning methods to accurately predict patterns in the stock market.

The Django framework, a high-level Python web framework that encourages quick development and simple, practical design, is used in the construction of the web application. Django makes it easier to build a scalable and reliable web application architecture, which guarantees smooth user experience and effective data administration.

The yfinance library makes data retrieval easier by allowing historical stock market data to be fetched straight from the Yahoo Finance API. This abundant supply of data gives the GRU model the training and testing input it needs, guaranteeing that the predictions are grounded in complete and current facts.

For preprocessing and data manipulation, NumPy and Pandas libraries are used. While Pandas offers strong data structures and tools for data analysis and manipulation, NumPy supports mathematical functions and efficient array operations. The machine learning pipeline can easily incorporate the stock market data that has been retrieved thanks to these libraries.

With the TensorFlow library, the GRU algorithm—a variation on recurrent neural networks (RNNs)—is implemented. TensorFlow offers a scalable and adaptable deep learning model creation and training framework, facilitating effective computation and GRU architecture optimization. Accurate forecasts can be produced by training the GRU model on vast amounts of historical stock market data by utilizing TensorFlow's vast ecosystem of tools and resources.

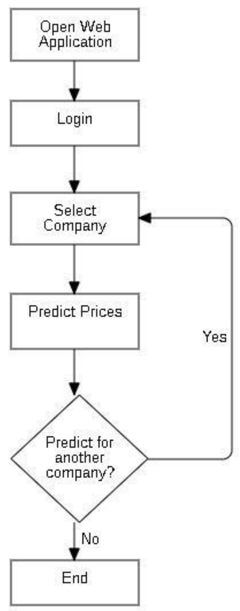



**Chart 1**: Flowchart of the web application

### 3.1 User Interface

The web application's user interface includes a dynamic home page that provides users with a thorough summary of popular tickers and their stock market performance as of late. When visitors arrive at the home page, they are met with an interactive graph showing the opening, closing, high, and low values of a few well-known tickers over the last few days. Users can quickly and easily evaluate the performance of different stocks at a glance with this graph, which gives them insightful information on recent trends and changes in the stock market. The home page provides visitors with a handy way to remain updated on the latest changes in the stock market and spot possible investment opportunities by visualizing key parameters for numerous tickers at once. Users may simply traverse the information offered on the home page and make informed decisions based on the real-time data provided thanks to its intuitive and user-friendly design.

**Figure-1:** Home Page

On clicking the predict option on the home page, users are taken to a form where they can provide the parameters for their stock price forecast. Users can input the ticker symbol of the stock they want to forecast and the number of days they want to forecast in the form's boxes. Users can submit their options and tailor the prediction process to suit their needs with ease because of the form's easy design. Whether it is for long-term or short-term forecasting, users can customize the prediction process to meet their specific needs thanks to this adaptable methodology. Users can start the prediction process and acquire precise forecasts for the chosen stock and time period by submitting the form after the parameters have been input.

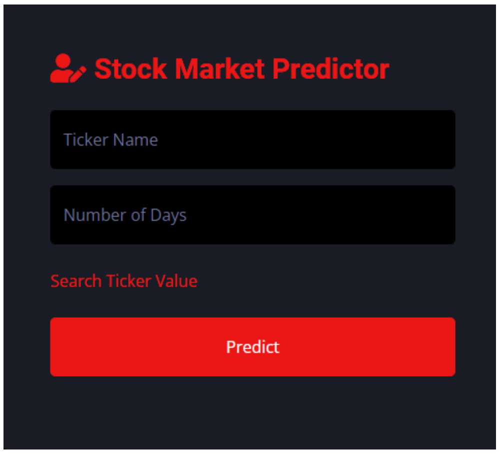



**Figure-2:** Prediction Form

Upon submitting the form, users are taken to a page with detailed information on the ticker they have chosen, as well as two plots that offer insightful information. The page begins by providing the most important information about the ticker, such as its name, symbol, market capitalization, volume, and other pertinent data, as well as its most recent price. With the help of this data, consumers can get a thorough understanding of the chosen stock's present market position.

Following the ticker information, the page includes two plots. The first plot displays the selected ticker's current prices across the set time period, giving users a visual depiction of its historical performance. The second plot shows the expected values for the given number of future days, depending on the user's input and the predictions made by the underlying machine learning model. This map gives users insight into the stock's predicted price trajectory, allowing them to forecast prospective market movements and make informed investing decisions.

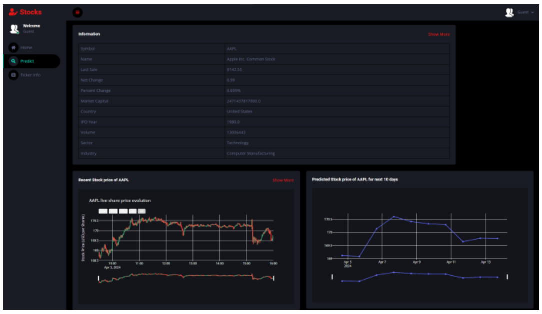


**Figure-3:** Predictions page

## 4. FUTURE WORK

Future upgrades to the web application could include sentiment analysis of news items or tweets about the selected stock. This update would give customers significant insights into market sentiment surrounding the stock, complementing the prediction model's quantitative research. By analyzing the sentiment of news stories, social media messages, and other textual data sources, the application may determine whether investors and market players are positive, negative, or neutral about the stock.

Implementing sentiment analysis would include using natural language processing (NLP) techniques to extract sentiment-related elements from textual data sources such as news articles or tweets. Machine learning models trained on labeled sentiment data could be used to determine if the sentiment of each text sample is positive, negative, or neutral. The sentiment scores acquired from the analysis can then be included as extra features in the prediction model, providing contextual information on market sentiment surrounding the stock.

Additionally, the web application might use sentiment analysis to produce trade signals or alerts based on sentiment, informing users of noteworthy changes in market sentiment that would be worthy of notice. By giving users practical insights and enabling them to navigate the stock market with more assurance and accuracy, these improvements would improve the web application's usefulness and efficacy.

## 5. CONCLUSIONS

By providing a thorough comparison of the LSTM and GRU algorithms and outlining the creation of an approachable web application specifically for stock prediction, this study makes a substantial contribution to the field of stock market prediction. After conducting a thorough empirical analysis, we found that the GRU algorithm outperforms LSTM in terms of predictive accuracy, hence offering investors more dependable projections. Taking use of this realization, we used Django to create a solid web application, integrating TensorFlow to implement the GRU model and yfinance for thorough data extraction. This platform makes it possible for users—from novices to experienced investors—to obtain precise and timely forecasts, enabling them to make well-informed decisions in the ever-changing world of finance. Furthermore, this application's user-friendly interface democratizes access to complex machine learning methods, increasing the accessibility and comprehensibility of stock market prediction for a wider range of users. As a result, this study contributes to our understanding of predictive modeling from a scientific standpoint while also having practical applications that will enable people to confidently and precisely navigate the complexity of the stock market.

## 6. REFERENCES

[1] Shubha Singh, Sreedevi Gutta and Ahmad Hadaegh, “Stock Prediction Using Machine Learning”, WSEAS TRANSACTIONS on COMPUTER RESEARCH, Volume 9, 2021.

[2] Ritika Chopra and Gagan Deep Sharma,” Application of Artificial Intelligence in Stock Market Forecasting: A Critique, Review, and Research Agenda”, J. Risk Financial Manag. 2021, 14, 526.

[3] Traianos-Ioannis Theodorou et al,” An AI-Enabled Stock Prediction Platform Combining News and Social Sensing with Financial Statements”, Future Internet 2021, 13, 138.

[4] Sohrab Mokhtari, Kang K Yen and Jin Liu,” Effectiveness of Artificial Intelligence in Stock Market Prediction Based on Machine Learning”, International Journal of Computer Applications, 2021.

[5] Adil Moghar and Mhamed Hamicheb. Stock Market Prediction Using LSTM Recurrent Neural Network, Procedia Computer Science. Volume 170, 2020, Pages 1168-1.

[6] Kyoung-jaeKim and IngooHan. prediction of stock price index. Volume 19, Issue 2, August 2000, Pages 125-132.

[7] Shipra Saxena. Introduction to Long Short-Term Memory. March 16, 2021.

[8] Makridakis, S.; Spiliotis, E.; Assimakopoulos, V. Statistical and Machine Learning forecasting methods: Concerns and ways forward. PLoS ONE 2018, 13, e0194889.

[9] Stock prediction based on bidirectional gated recurrent unit with convolutional neural network and feature selection Zhou Q, Zhou C, Wang X (2022) Stock prediction based on bidirectional gated recurrent unit with convolutional neural network and feature selection. PLOS ONE 17(2): e0262501.

[10] Chen C, Xue L, Xing W. Research on Improved GRU-Based Stock Price Prediction Method. *Applied Sciences*. 2023; 13(15):8813.

Rephrase of Comparative Analysis of LSTM and GRU Algorithms for Stock Market Prediction_ Development of a Web Application (1).pdf

ORIGINALITY REPORT

| 11% | 10% | 4% | 8% |
|---|---|---|---|
| SIMILARITY INDEX | INTERNET SOURCES | PUBLICATIONS | STUDENT PAPERS |

PRIMARY SOURCES

| | | |
|---|---|---|
| 1 | Submitted to Rochester Institute of Technology<br>Student Paper | 6% |
| 2 | Shuming Sun, Juan Chen, Jian Sun. "Traffic congestion prediction based on GPS trajectory data", International Journal of Distributed Sensor Networks, 2019<br>Publication | 1% |
| 3 | Submitted to Asia Pacific University College of Technology and Innovation (UCTI)<br>Student Paper | 1% |
| 4 | assets.researchsquare.com<br>Internet Source | 1% |
| 5 | doczz.com.br<br>Internet Source | 1% |
| 6 | www.mdpi.com<br>Internet Source | 1% |
| 7 | link.springer.com<br>Internet Source | 1% |

| | | | |
|---|---|---|---|
| Exclude quotes | On | Exclude matches | < 1% |
| Exclude bibliography | On | | |

e-ISSN: 2395-0056 p-ISSN: 2395-0072

# International Research Journal of Engineering and Technology (IRJET)

( An ISO 9001 : 2008 Certified Journal )

*Is hereby awarding this certificate to*

## *Ankit Singh*

*In recognition the publication of the manuscript entitled*

## *"Comparative Analysis of LSTM and GRU Algorithms for Stock Market Prediction: Development of a Web Application"*

*published in our Journal Volume 11 Issue 4 April 2024*

Editor in Chief

E-mail : editor@irjet.net

Impact Factor : 8.226

www.irjet.net

e-ISSN: 2395-0056 p-ISSN: 2395-0072

# International Research Journal of Engineering and Technology (IRJET)

( An ISO 9001 : 2008 Certified Journal )

*Is hereby awarding this certificate to*

## *Vallabh Yevade*

*In recognition the publication of the manuscript entitled*

## *"Comparative Analysis of LSTM and GRU Algorithms for Stock Market Prediction: Development of a Web Application"*

*published in our Journal Volume 11 Issue 4 April 2024*

Impact Factor : 8.226

www.irjet.net

Editor in Chief

E-mail : editor@irjet.net

e-ISSN: 2395-0056 p-ISSN: 2395-0072

# International Research Journal of Engineering and Technology (IRJET)

( An ISO 9001 : 2008 Certified Journal )

*Is hereby awarding this certificate to*

*Vaibhav Survase*

*In recognition the publication of the manuscript entitled*

*"Comparative Analysis of LSTM and GRU Algorithms for Stock Market Prediction: Development of a Web Application"*

*published in our Journal Volume 11 Issue 4 April 2024*

Impact Factor : 8.226

www.irjet.net

Editor in Chief

E-mail : editor@irjet.net

e-ISSN: 2395-0056 p-ISSN: 2395-0072

# International Research Journal of Engineering and Technology (IRJET)

( An ISO 9001 : 2008 Certified Journal )

*Is hereby awarding this certificate to*

*Sahil Jhodge*

*In recognition the publication of the manuscript entitled*

*"Comparative Analysis of LSTM and GRU Algorithms for Stock Market Prediction: Development of a Web Application"*

*published in our Journal Volume 11 Issue 4 April 2024*

Impact Factor : 8.226

www.irjet.net

Editor in Chief

E-mail : editor@irjet.net

e-ISSN: 2395-0056 p-ISSN: 2395-0072

# International Research Journal of Engineering and Technology (IRJET)

( An ISO 9001 : 2008 Certified Journal )

*Is hereby awarding this certificate to*

*Prof. D.A. Gore*

*In recognition the publication of the manuscript entitled*

*"Comparative Analysis of LSTM and GRU Algorithms for Stock Market Prediction: Development of a Web Application"*

*published in our Journal Volume 11 Issue 4 April 2024*

Impact Factor : 8.226

www.irjet.net

Editor in Chief

E-mail : editor@irjet.net

# CHAPTER 8

# REFERENCES


1. Chen C, Xue L, Xing W. Research on Improved GRU-Based Stock Price Prediction Method. Applied Sciences. 2023; 13(15):8813.

2. Shubha Singh, Sreedevi Gutta and Ahmad Hadaegh, “Stock Prediction Using Machine Learning”, WSEAS TRANSACTIONS on COMPUTER RESEARCH, Volume 9, 2021.

3. Ritika Chopra and Gagan Deep Sharma, “Application of Artificial Intelligence in Stock Market Forecasting: A Critique, Review, and Research Agenda”, J. Risk Financial Manag. 2021, 14, 526. https://doi.org/10.3390/jrfm14110526.

4. Traianos-Ioannis Theodorou et al, “An AI-Enabled Stock Prediction Platform Combining News and Social Sensing with Financial Statements”, Future Internet 2021, 13, 138. https://doi.org/10.3390/fi13060138.

5. Sohrab Mokhtari, Kang K Yen and Jin Liu, “Effectiveness of Artificial Intelligence in Stock Market Prediction Based on Machine Learning”, International Journal of Computer Applications, 2021.

6. Adil Moghar and Mhamed Hamicheb. Stock Market Prediction Using LSTM Recurrent Neural Network, Procedia Computer Science. Volume 170, 2020, Pages 1168-1.

7. Sohrab Mokhtari and Kang K Yen, “A Novel Bilateral Fuzzy Adaptive Unscented Kalman Filter and its Implementation to Nonlinear Systems with Additive Noise”. In: 2020 IEEE Industry Applications Society Annual Meeting. IEEE. 2020, pp. 1–6.

8. Eugene F Fama, “Efficient capital markets: II”. In: The journal of finance 46.5 (1991), pp. 1575–1617.

9. Andrew W Lo, “The adaptive markets hypothesis”. In: The Journal of Portfolio Management 30.5 (2004), pp. 15–29.

10. From Charles D Kirkpatrick II and R Julie, “Dow Theory”. In: CMT Level I 2019: An Introduction to Technical Analysis (2019), p. 15.

11. Robert D Edwards, WHC Bassetti, and John Magee, Technical analysis of stock trends. CRC press, 2007.

12. Joseph D Piotroski, “Value investing: The use of historical financial statement information to separate winners from losers”. In: Journal of Accounting Research (2000), pp. 1–41.

## 8.1 ACRONYMS

Table 8.1: List of Abbreviations

| Abbreviation | Full Form |
|---|---|
| AI | Artificial Intelligence |
| API | Application Programming Interface |
| ANN | Artificial Neural Network |
| ARIMA | Autoregressive Integrated Moving Average |
| BP | Backpropagation |
| CNN | Convolutional Neural Network |
| EMH | Efficient Market Hypothesis |
| FFNN | Feedforward Neural Network |
| GRU | Gated Recurrent Unit |
| LSTM | Long Short-Term Memory |
| MAE | Mean Absolute Error |
| MLR | Multi-Linear Regression |
| MSE | Mean Squared Error |
| NLP | Natural Language Processing |
| RNN | Recurrent Neural Network |
| RSI | Relative Strength Index |
| SDLC | Software Development Life Cycle |
| SVM | Support Vector Machine |