Stock Forecasting with Hybrid Deep Learning and Anomaly Detection.

Report Submitted in partial fulfillment of requirements for the B.Tech degree in Computer Science and Engineering

By:

NAME OF THE STUDENT	ROLL NUMBER
J MUNIYANDI	2021UCS1577
RAMANJOT SINGH	2021UCS1598
MUDIT JAIN	2021UCS1604

Under the Supervision

of

(Dr. Savita Yadav)

Department of Computer Science and Engineering

NETAJI SUBHAS UNIVERSITY OF TECHNOLOGY (NSUT)
NEW DELHI , INDIA - 110078

MAY 2025

CERTIFICATE

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

This is to certify that the work embodied in project thesis titled, "Enhancing Stock forecasting with Anomaly Detection and Hybrid deep learning models" by Ramanjot Singh (2021UCS1598), Mudit Jain(2021UCS1604) and J Muniyandi (2021UCS1577) is the bonafide work of the group submitted to Netaji Subhas University of Technology for consideration in 8th Semester B.Tech. Project Evaluation.

The original Research work was carried out by the team under my/our guidance and supervision in the academic year 2024-25. This work has not been submitted for any other diploma or degree of any university. On the basis of a declaration made by the group, we recommend the project report for evaluation.

Dr.Savita Yadav
(Assistant Professor)
Department of Computer Science & Engineering
Netaji Subhas University of Technology

CANDIDATES DECLARATION

I/We, J Muniyandi(2021UCS1577), Ramanjot Singh (2021UCS1598) and Mudit Jain (2021UCS1604) of B. Tech. Department of Computer Science & Engineering, hereby declare that the Project-Thesis titled "Enhancing Stock forecasting with Anomaly Detection and Hybrid deep learning models" which is submitted by me/us to the Department of Computer Science & Engineering, Netaji Subhas University of Technology (NSUT) Dwarka, New Delhi in partial fulfillment of the requirement for the award of the degree of Bachelor of Technology is original and not copied from the source without proper citation. The manuscript has been subjected to plagiarism checks by Turnitin software. This work has not previously formed the basis for the award of any Degree.

Date:

J Muniyandi	Ramanjot Singh	Mudit Jain
2021UCS1577	2021UCS1598	2021UCS1604

CERTIFICATE OF DECLARATION

This is to certify that the Project-Thesis titled "Enhancing Stock forecasting with Anomaly Detection and Hybrid deep learning models" which is being submitted by Ramanjot Singh (2021UCS1598) and Mudit Jain (2021UCS1604) to the Department of Computer Science & Engineering, Netaji Subhas University of Technology (NSUT) Dwarka, New Delhi in partial fulfillment of the requirement for the award of the degree of Bachelor of Technology, is a record of the thesis work carried out by the students under my supervision and guidance. The content of this thesis, in full or in parts, has not been submitted for any other degree or diploma.

Dr.Savita Yadav
(Assistant Professor)
Department of Computer Science & Engineering
Netaji Subhas University of Technology

ACKNOWLEDGEMENT

We would like to express my gratitude and appreciation to all those who make it possible to complete this project. Special thanks to our project supervisor(s) Dr. Savita Yadav whose help, stimulating suggestions and encouragement helped us in writing this report. We also sincerely thank our colleagues for the time spent proofreading and correcting our mistakes.

We would also like to acknowledge with much appreciation the crucial role of the staff in Computer Science & Engineering, who gave us permission to use the lab and the systems and gave permission to use all necessary things related to the project.

PLAGIARISM REPORT

DrillBit

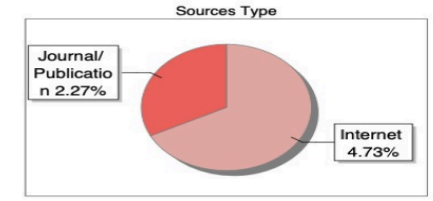
The Report is Generated by DrillBit Plagiarism Detection Software

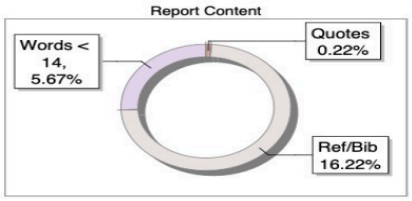
Submission Information

Author Name	Mudit Jain
Title	Stock Forecasting with Hybrid Deep Learning and Anomaly Detection.
Paper/Submission ID	3601258
Submitted by	savita.yadav@nsut.ac.in
Submission Date	2025-05-10 22:07:40
Total Pages, Total Words	35, 5063
Document type	Thesis

Result Information

Similarity **7 %**





Exclude Information

Quotes	Excluded
References/Bibliography	Excluded
Source: Excluded < 14 Words	Excluded
Excluded Source	0 %
Excluded Phrases	Not Excluded

Database Selection

Language	English
Student Papers	Yes
Journals & publishers	Yes
Internet or Web	Yes
Institution Repository	Yes

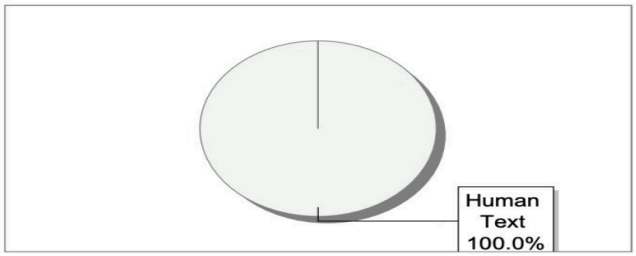
A Unique QR Code use to View/Download/Share Pdf File

Submission Information

Author Name	Mudit Jain
Title	Stock Forecasting with Hybrid Deep Learning and..
Paper/Submission ID	3601258
Submitted By	savita.yadav@nsut.ac.in
Submission Date	2025-05-10 22:07:40
Total Pages	35
Document type	Thesis

Result Information

AI Text: **0 %**



ABSTRACT

This research introduces an innovative approach to stock price prediction that combines several advanced machine learning methods. We start by employing Long Short-Term Memory (LSTM) networks to explore temporal patterns in stock market data sets. Striving to enhance the accuracy of our predictions, we created a new model that employs LSTM with XGBoost to facilitate the integration of both sequence and structured market effects.

One of the key improvements in our approach is the use of advanced anomaly detection methods, which employ both Isolation Forest and Autoencoder methods to identify and remove outlier market trends that can influence our forecasts. In addition, we have broadened our examination by including a wide range of financial metrics, including company-specific data such as quarterly statements, as well as more general economic metrics like GDP growth, inflation rates, and interest rates.

Holistic methodology provides an overall image of market trends. A new hybrid model, which is at the core of our research, integrates three key features: an Autoencoder to extract meaningful features, an Attention-based LSTM to improve the understanding of temporal patterns, and XGBoost to make precise predictions. The overall approach allows us to make more precise and reliable predictions on stock prices.

INDEX

LIST OF FIGURES

CHAPTER 4

LIST OF TABLES

CHAPTER 4

CHAPTER 1

INTRODUCTION

Predicting stock prices is a vital area of study in financial markets, providing valuable tools for investors to make better investment decisions and manage risks effectively. Stock markets are complex systems in which prices are decided by numerous factors, including past price movements, company-specific information, and general economic conditions. This paper introduces a novel approach to stock price prediction based on deep learning, ensemble methods, and advanced anomaly detection techniques.

Our approach begins using Long Short-Term Memory (LSTM) networks, which by their nature are suited to learning temporal variations in stock prices. We then developed an even more sophisticated model by combining LSTM with XGBoost, thus developing a robust model that is capable of analyzing temporal patterns and structured data. Another important aspect of our approach is the inclusion of sophisticated anomaly detection systems, both Isolation Forest and Autoencoder, to identify and remove unusual market trends that can affect our predictions. We have also included a wide range of financial indicators in our analysis, including firm-specific data like quarterly earnings, along with more general economic indicators like GDP growth, inflation rates, and interest rates.

The significant innovation in our research is an innovative hybrid model that integrates three central elements: an Autoencoder for determining significant features, an Attention-based LSTM for enhanced understanding of time-based patterns, and XGBoost for precise prediction. The integrated process helps our model to provide more credible predictions on new data without allowing cases of overfitting. The findings prove significant improvement in prediction credibility, providing valuable insight for investment choices and portfolio management strategies.

MOTIVATION

The dynamic nature of stock markets presents a complex challenge for investors and analysts, as market movements are shaped by an intricate web of factors ranging from company-specific developments to global economic trends. Conventional techniques of forecasting stock prices, which rely to a great degree on past price information and technical analysis, are typically not able to cover the whole range of market variables, particularly under conditions of extreme market turbulence.

Economic instability, as evidenced in events like market downturns or inflationary pressures, tends to generate erratic stock price fluctuations that defy standard forecasting methods. Similarly, important corporate events like quarterly earnings announcements, management changes, and strategic initiatives can generate extreme variations in stock valuations that prove difficult to forecast using standard analytical tools.
This research is driven by the appreciation that modern stock market analysis requires a more nuanced and combined approach. The goal is to create an integrated forecasting model that encompasses fundamental financial data, broader economic metrics, and advanced machine learning techniques.

Our research tries to solve three key issues: improving data quality with more sophisticated anomaly detection, improving the accuracy of prediction with novel hybrid modeling approaches, and offering more trustworthy analytical tools for investment decision-making. Through the combination of all these features, we attempt to provide market participants with a more solid platform on which to deal with the complexities of today's financial markets and make wiser investment decisions.

LITERATURE SURVEY

1. Traditional Approaches for Stock Price Prediction

- Early methods for stock price forecasting primarily relied on statistical and econometric models:
  - **Autoregressive Integrated Moving Average (ARIMA)**: Extensively used for time-series forecasting, leveraging past stock prices to make future predictions. However, these models assume stationarity in data, making them less effective in volatile stock markets [3, 7, 12].
  - **Generalized Autoregressive Conditional Heteroskedasticity (GARCH)**: Introduced to model stock market volatility, capturing heteroskedasticity in financial data. Although these models work well for predicting market volatility, they struggle to capture complex, nonlinear market patterns [9, 14].
  - **Support Vector Regression (SVR)**: Applied in financial prediction due to its ability to handle high-dimensional data and nonlinear relationships. However, it struggles with time-dependent features and sequential dependencies in stock price data [5, 10, 15].

2. Machine Learning and Ensemble Techniques

- Researchers have developed different models to enhance stock price prediction by understanding complex relationships and interactions between multiple market factors
  - **Random Forest and Decision Trees**: Commonly used for analyzing organized financial data and preventing overfitting issues. However, they lack sequential memory, limiting their performance in time-series forecasting [5, 9, 14].
  - **Extreme Gradient Boosting (XGBoost)**: Applied in financial markets for feature-based prediction. Studies have shown its ability to capture relationships between stock prices and macroeconomic indicators [2, 8, 11, 17, 20]. However, it naturally lacks the ability to understand time-based relationships in the data
  - **Hybrid Models (SVR + XGBoost, ARIMA + XGBoost)**: Researchers have combined traditional statistical models with machine learning techniques. For

instance, ARIMA is often used to model linear trends, while XGBoost is applied for nonlinear feature extraction, improving overall accuracy [6, 13, 18].

3. Deep Learning-Based Approaches

- Deep learning has transformed financial forecasting by creating models that can understand complex time-based and nonlinear patterns.
  - **Recurrent Neural Networks (RNNs)**: Widely used for sequential data processing in stock price prediction. However, due to issues like vanishing gradients, they struggle with long-term dependencies [4, 6, 10].
  - **Long Short-Term Memory (LSTM)**: Overcomes RNN limitations by incorporating memory cells that retain information over long sequences. Research has demonstrated that LSTM models work well in identifying long-term trends in stock price movements [1, 4, 6, 10, 15, 18]. However, they sometimes fail to respond to short-term fluctuations and external market factors.
  - **Gated Recurrent Units (GRU)**: A variant of LSTM with fewer parameters, making it computationally efficient. While GRU performs similarly to LSTM in many cases, it may not always capture long-range dependencies as effectively [3, 9, 12].
  - **Attention-Based Mechanisms**: Recent research has enhanced LSTM performance using attention mechanisms, which prioritize significant time steps in sequential data. These approaches have shown better results in predicting financial time-series data [7, 11, 16].

4. Data Preprocessing and Anomaly Detection

- High-quality data is crucial for accurate stock price prediction. Different methods have been developed to deal with noise, missing data, and unusual values in financial information.
  - **Outlier Detection Methods**:
    - **Isolation Forest and One-Class SVM**: Used to detect and remove anomalous stock price fluctuations [8, 13].

- **Z-Score and Interquartile Range (IQR)**: Statistical methods to filter extreme price movements [10, 15].
- **Autoencoder-Based Anomaly Detection**: A deep learning technique that reconstructs normal data patterns and flags deviations [12, 19].

- **Feature Engineering and Selection**:
  - **Technical Indicators**: Moving averages, Bollinger Bands, and Relative Strength Index (RSI) are commonly used features in predictive models [14, 21].
  - **Fundamental and Macroeconomic Data**: Incorporating financial ratios (P/E ratio, earnings per share) and macroeconomic indicators (inflation, GDP growth) improves forecasting accuracy [13, 16, 19, 22, 24].
  - **Sentiment Analysis**: Models have incorporated social media content, news articles, and investor sentiment to understand how market mood affects stock prices [9, 18, 23].

5. Hybrid and Advanced Architectures

- Researchers have tested various advanced combined models to improve stock price prediction.
  - **LSTM + CNN**: Convolutional Neural Networks (CNNs) have been used alongside LSTMs to extract spatial and temporal patterns in stock price data [5, 14, 17].
  - **Wavelet Transform + Deep Learning**: Wavelet transforms break down stock price signals before processing them through deep learning models, leading to better feature identification [3, 11, 16].
  - **Reinforcement Learning for Trading Strategies**: Some studies have applied reinforcement learning to predict stock price movements and optimize trading decisions [7, 15, 20].
  - **Isolation Forest for anomaly detection:** Detected and removed anomalies in high-dimensional data, ensuring cleaner and more reliable input for model

training. This preprocessing step improved overall model robustness and prediction accuracy [8 , 10 , 12 , 13].

6. Challenges and Limitations in Existing Literature

- **Market Volatility and Unpredictability**: Sudden market crashes and economic crises introduce uncertainty that most models struggle to predict [6, 12, 22].
- **Data Availability and Quality**: Reliable financial data is crucial, but missing values, anomalies, and inconsistent reporting affect model performance [9, 14, 19].
- **Overfitting and Generalization Issues**: Many deep learning models tend to overfit historical data, making them less effective in real-world applications [8, 18, 21].
- **Computational Complexity**: Sophisticated combined models need substantial computing power, which can limit their practical application [11, 16, 24].

CHAPTER 2

PROBLEM STATEMENT

To propose a solution that integrates LSTM and XGBoost in an ensemble framework, incorporating attention mechanisms for enhanced temporal pattern recognition , anomaly detection methods for data quality assurance and interpretable predictions of stock price movements based on both macroeconomic and fundamental indicators.

OBJECTIVE

This research aims to develop an innovative hybrid model for stock price forecasting that synergistically combines deep learning and ensemble learning methodologies. The study focuses on creating a unified framework that leverages LSTM networks for time-series analysis and XGBoost for structured data processing. We will incorporate both micro-level company financial data and macro-level economic indicators to improve prediction accuracy. A key component of our approach involves implementing sophisticated anomaly detection systems to ensure data quality by identifying and addressing market outliers and irregularities. This integrated methodology aims to deliver more reliable stock market predictions by effectively handling both temporal dependencies and structural relationships in the data

Specific Objectives:

- Feature Engineering & Anomaly Detection: Develop an feature extraction system using Autoencoder architecture to identify significant patterns in financial and economic data. Implement a dual-layer anomaly detection system combining Isolation Forest and Autoencoder techniques to identify and handle market irregularities.

- Temporal Dependency Modeling: Design and implement an Attention-based LSTM network specifically tailored to capture the complex temporal patterns in stock price movements. This system will focus on understanding both immediate market reactions and longer-term trend developments, providing a comprehensive view of market dynamics.

- Structured Data Analysis: Use XGBoost algorithms to study and understand the complex connections between different market factors, including company financial data and broader economic indicators.

- Hybrid Ensemble Framework Development: Create a sophisticated ensemble model that effectively combines the strengths of LSTM and XGBoost through a Gradient Boosting Regressor. This integration will provide a balanced approach to market prediction, leveraging both temporal and structural aspects of the data.

CHAPTER 3

METHODOLOGY

Our methodology is structured around a four-phase approach to stock price prediction: data acquisition, preprocessing, model architecture design, and validation. This systematic framework allows us to effectively capture market patterns while maintaining model stability and preventing overfitting issues.

Data Preprocessing

Historical Stock Data Collection

- Daily stock price data was collected, including key attributes such as open, high, low, close, and volume.
- The dataset covered 10 years to provide sufficient temporal coverage for training and testing.
- Adjusted close prices were included to account for dividends and stock splits.

Fundamental Financial Data

- Collected quarterly financial metrics such as earnings reports, price-to-earnings (P/E) ratio, revenue, and net income.
- Ensured data consistency by aligning it with macroeconomic indicators to prevent misalignment issues.

Macroeconomic Indicators

- Retrieved key macroeconomic indicators besides GDP growth rate, inflation rate, and interest rates.
- Used interpolation and aggregation techniques to align data frequencies with quarterly financial reports.
- Applied a correlation heatmap to filter out irrelevant features, reducing model complexity and mitigating overfitting risks.
- Final selected features: 'Open', 'High', 'Low', 'Volume', 'Adj Close', 'Total Revenue', 'GDP', 'Net Income', 'Gross Profit', 'Cashflow from Financing'.

Data Merging and Alignment

- Merged historical stock data, fundamental financial data, and macroeconomic indicators based on timestamps.
- Converted daily stock prices to quarterly averages for better alignment with fundamental and macroeconomic data.

Feature Scaling

- Min-Max Scaling was applied to numerical features for compatibility with LSTM models.
- Standardization was used for models like XGBoost and meta-learners to maintain numerical stability.

LSTM Model

Our initial approach utilizes LSTM networks to capture the complex temporal patterns in stock price movements. The model architecture incorporates multiple layers with carefully selected units and activation functions to achieve an optimal balance between model complexity and performance. We conducted preliminary experiments to fine-tune key parameters such as learning rates and batch sizes, using cross-validation to ensure robust model configuration.

Stacked Model

The core of our prediction system combines two complementary models: an LSTM network for temporal pattern recognition and an XGBoost regressor for feature-based analysis. To enhance prediction accuracy, we implemented a multi-level stacking approach where the outputs from both models serve as inputs to a meta-learner. This meta-learner, implemented as a regression model, effectively combines the strengths of both the LSTM and XGBoost models, resulting in a more robust and accurate prediction system.

Anomaly Detection for Data Cleaning

We used several anomaly detection methods to improve data quality and reduce noise.

- Isolation Forest: Detected and removed unusual price movements that could distort predictions.
- Z-Score & IQR Method: Applied Z-Score and IQR techniques to identify and filter extreme market outliers.

- Autoencoder-Based Anomaly Detection: Utilized Autoencoder networks to detect complex patterns of irregularities in time-series data.

Data cleaning significantly improved model generalization, leayding to a more robust predictive model.

Hybrid Model for Greater Prediction Accuracy

To further refine stock price predictions, a hybrid model was developed incorporating multiple advanced techniques:

1. Autoencoder for Feature Extraction

- Reduced dimensionality and extracted key features, minimizing redundancy in input data.
- Enhanced data representation by filtering out irrelevant noise, allowing the model to focus on meaningful information.

2. Attention-Based LSTM for Temporal Learning

- Introduced an attention mechanism to prioritize significant time steps over less relevant historical data.
- Enhanced the model's capability to identify significant stock price movements, leading to better trend recognition.

3. XGBoost for Final Prediction

- Integrated refined features from the Autoencoder and Attention-LSTM layers.
- Used a decision-tree-based learning approach to enhance predictive accuracy, leveraging both structured and unstructured data.

IMPLEMENTATION

Platform and Tools:

- **Programming Language:** Python
- **Libraries:**
  - **Pandas/Numpy**: For data manipulation.
  - **XGBoost**: For building gradient boosting models.
  - **Keras/TensorFlow**: For implementing LSTM networks.
  - **Scikit-learn**: For model evaluation and ensemble methods.
  - **Matplotlib/Seaborn**: For visualizing data trends.
  - **APIs**: yfinance and Alpha Vantage for fetching historical stock and macroeconomic data.

Hardware Requirements:

- Mid-range CPU / GPU
- Minimum of 16GB RAM is recommended for handling large datasets.

Dataset:

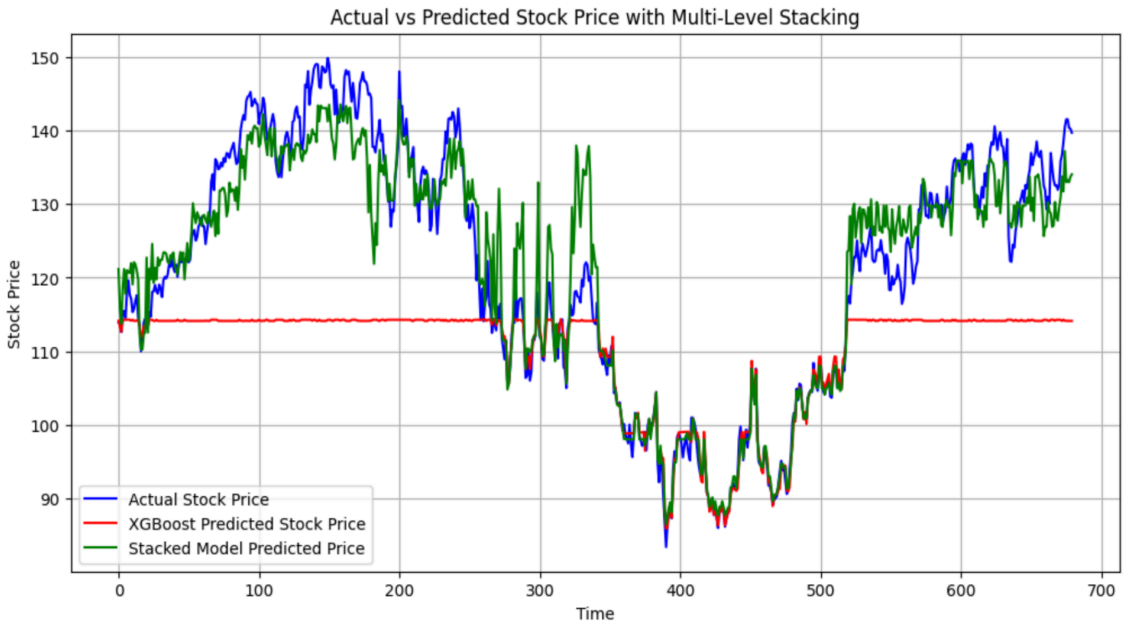
- **Historical Data from yfinance**: Includes fields like date, open, high, low, close, etc.
- **Fundamental Data from Alpha Vantage**: Balance sheet, income statement, cash flow, etc.
- **Macroeconomic Indicators from FRED**: GDP, inflation, unemployment, interest rate, etc.

CHAPTER 4

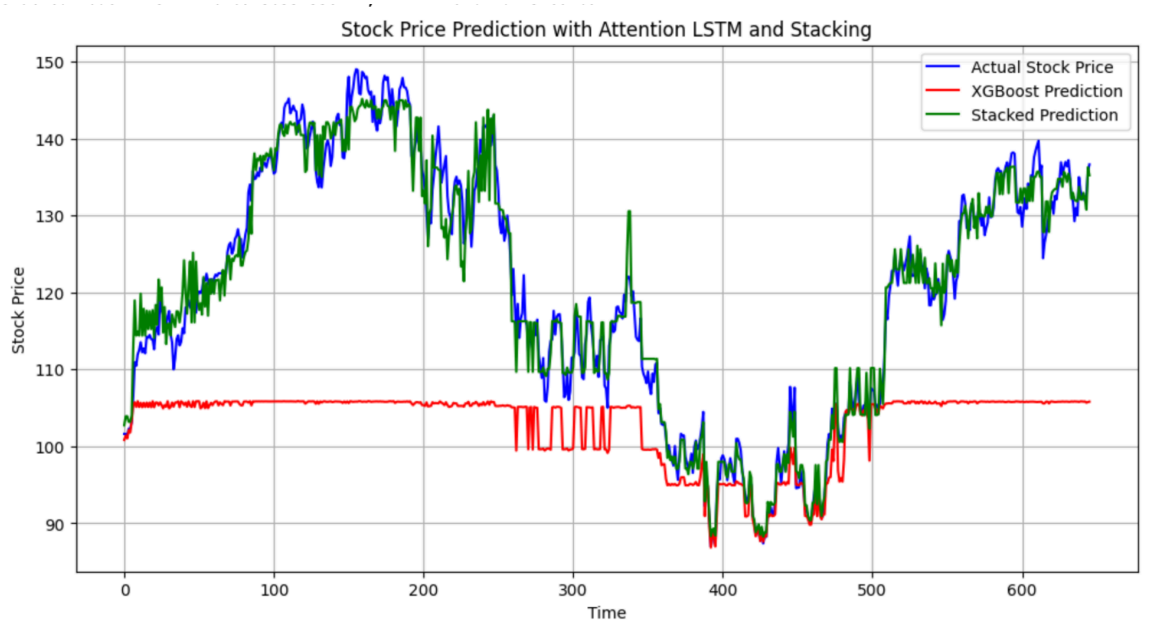
RESULTS AND DISCUSSIONS

Company Name	Hybrid Model - LSTM , XGBoost	Hybrid Model - Anomaly Detection, Attention-LSTM, XGBoost
Amazon	RMSE : 9.99 , MAE : 6.18 , R2 : 0.87	RMSE : 5.5 , MAE : 9.29 , R2 : 0.90
Google	RMSE : 4.74 , MAE : 3.29 , R2 : 0.92	RMSE : 2.64, MAE : 1.97 , R2 : 0.97
Apple	RMSE : 8.01 , MAE : 5.66 , R2 : 0.96	RMSE : 5 , MAE : 3.48 , R2 : 0.99
Tesla	RMSE : 4.35 , MAE : 3.01 , R2 : 0.91	RMSE : 1.41 , MAE : 0.88 , R2 : 0.97
Nvidia	RMSE : 47.75 , MAE : 32.98 , R2 : 0.72	RMSE : 18.8 , MAE : 11.35 , R2 : 0.96

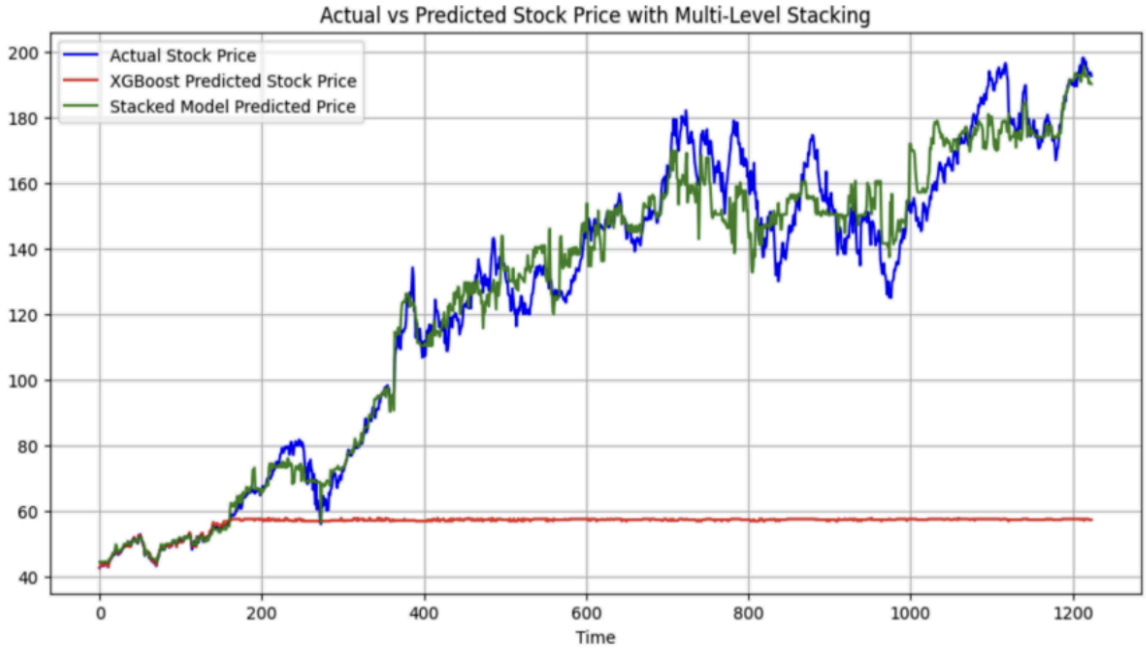
Comparison between Hybrid model and Hybrid Model - Anomaly Detection
Table 1



Google: Hybrid Model (LSTM - XGBoost)
Figure 1.1

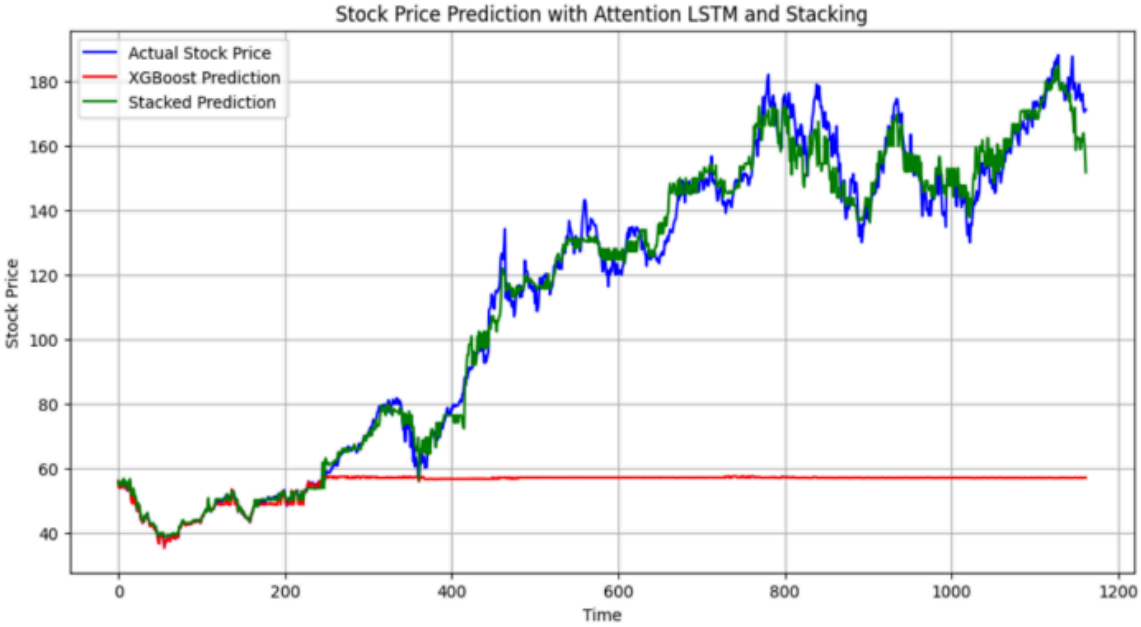


Google: Hybrid Model (LSTM - XGBoost) with anomaly detection
Figure 1.2



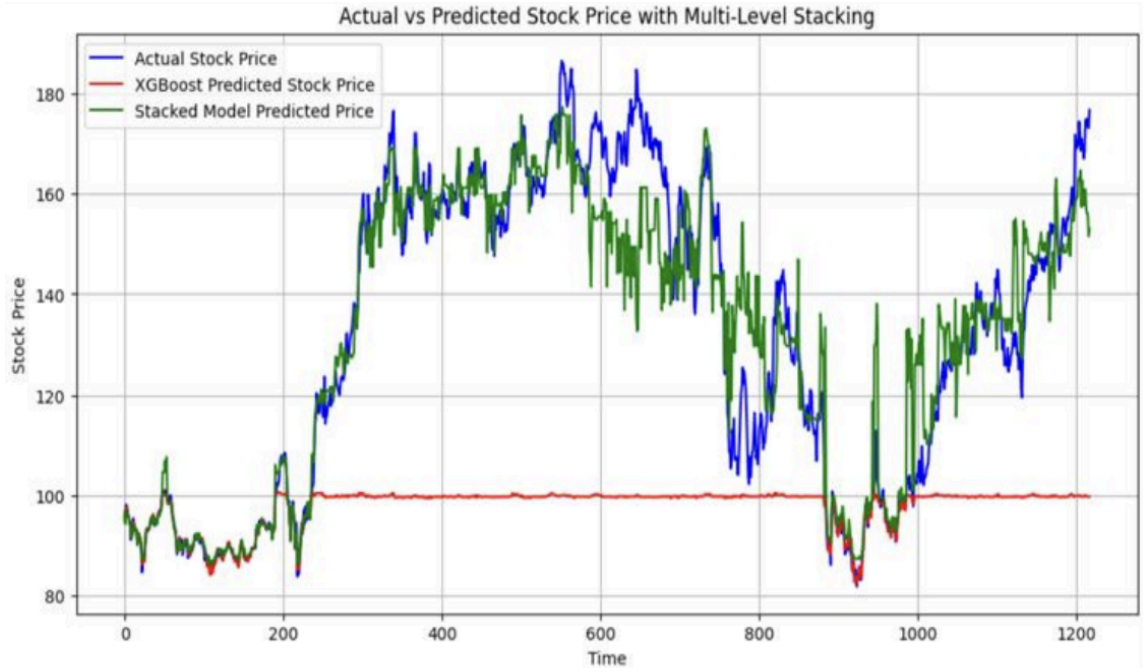
Apple: Hybrid Model (LSTM - XGBoost)

Figure 2.1



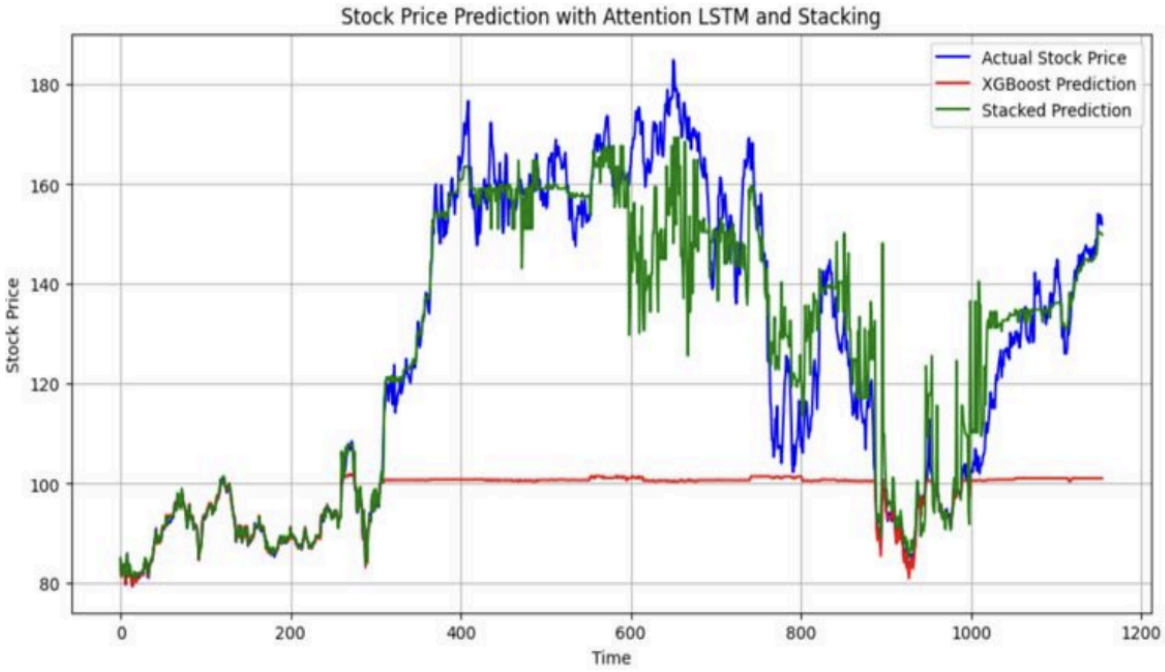
Apple: Hybrid Model (LSTM - XGBoost) with anomaly detection

Figure 2.2



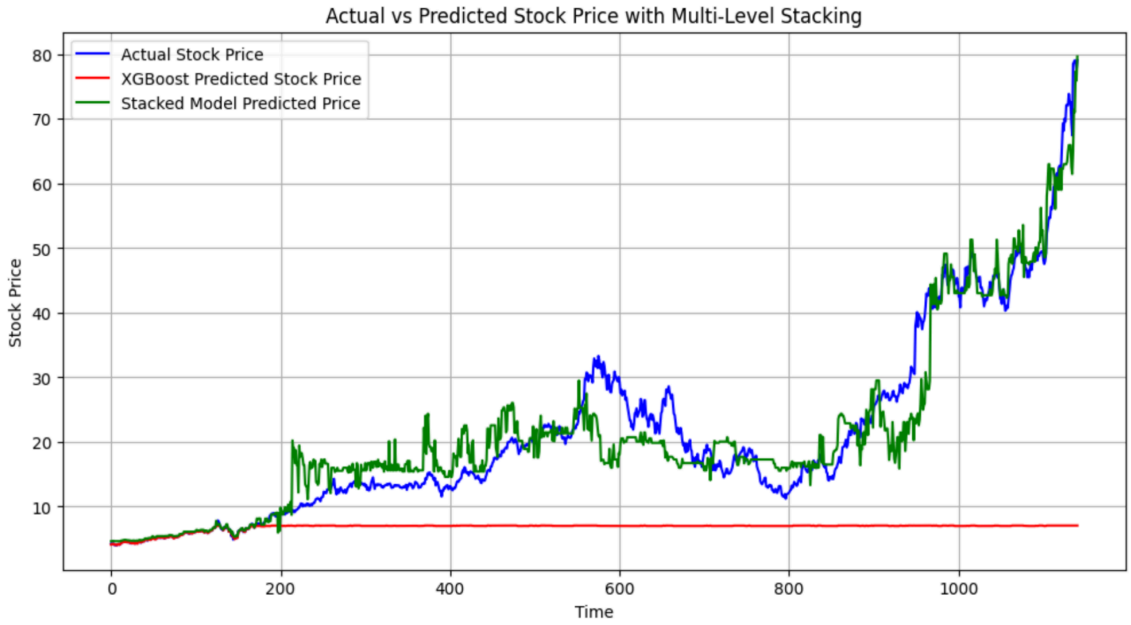
Amazon: Hybrid Model (LSTM - XGBoost)

Figure 3.1



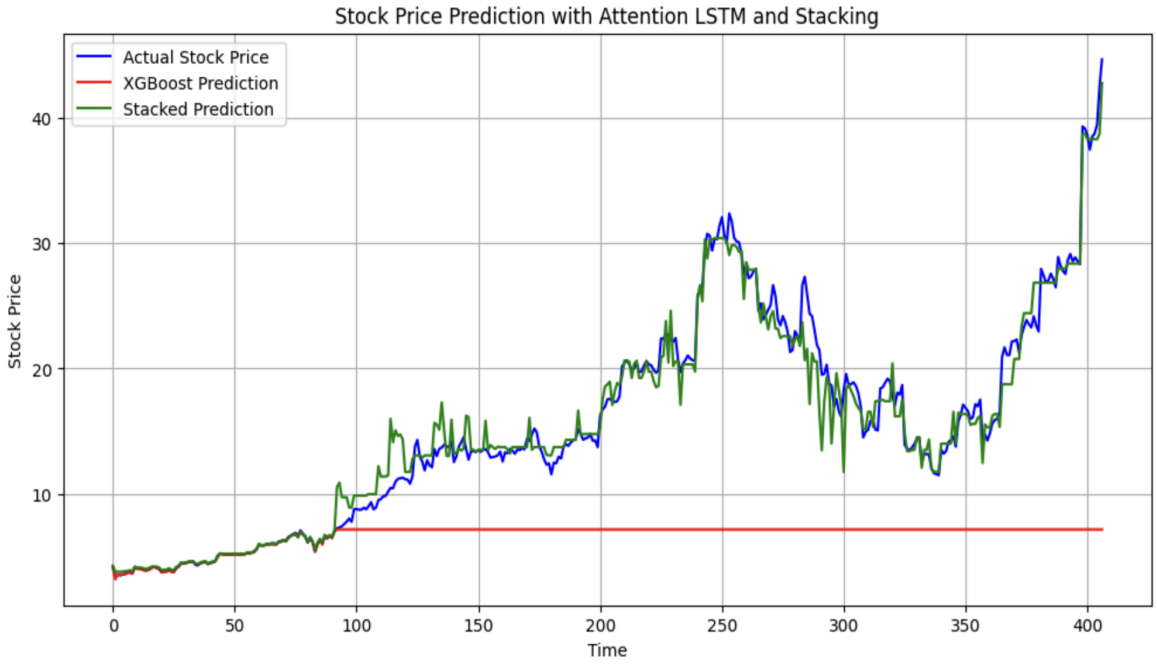
Amazon: Hybrid Model (LSTM - XGBoost) with anomaly detection

Figure 3.2



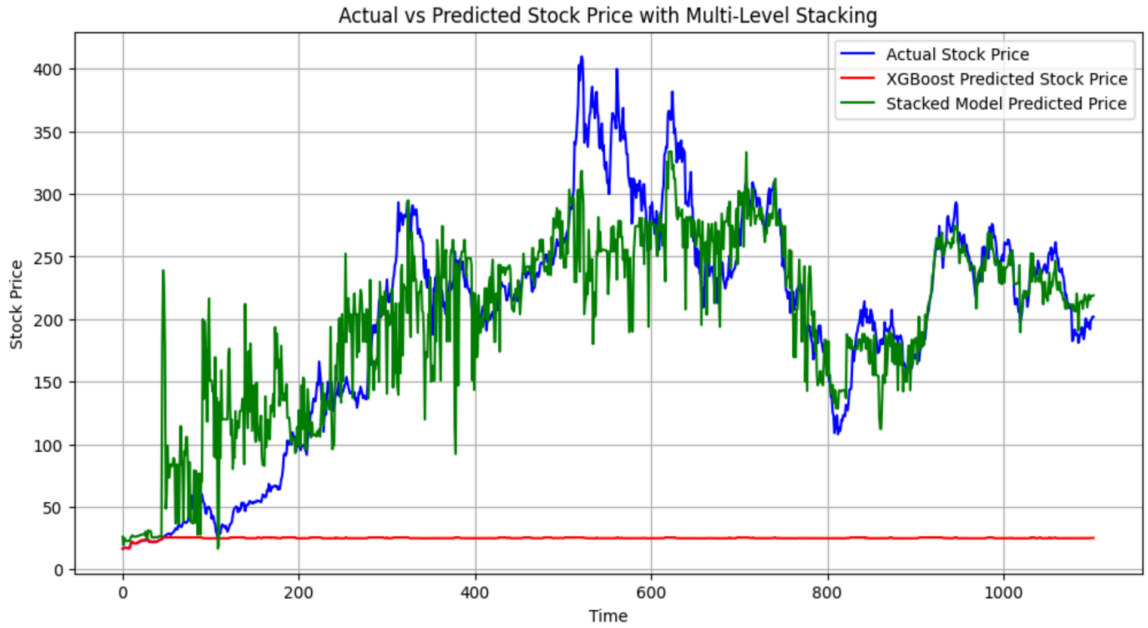
Nvidia: Hybrid Model (LSTM - XGBoost)

Figure 4.1



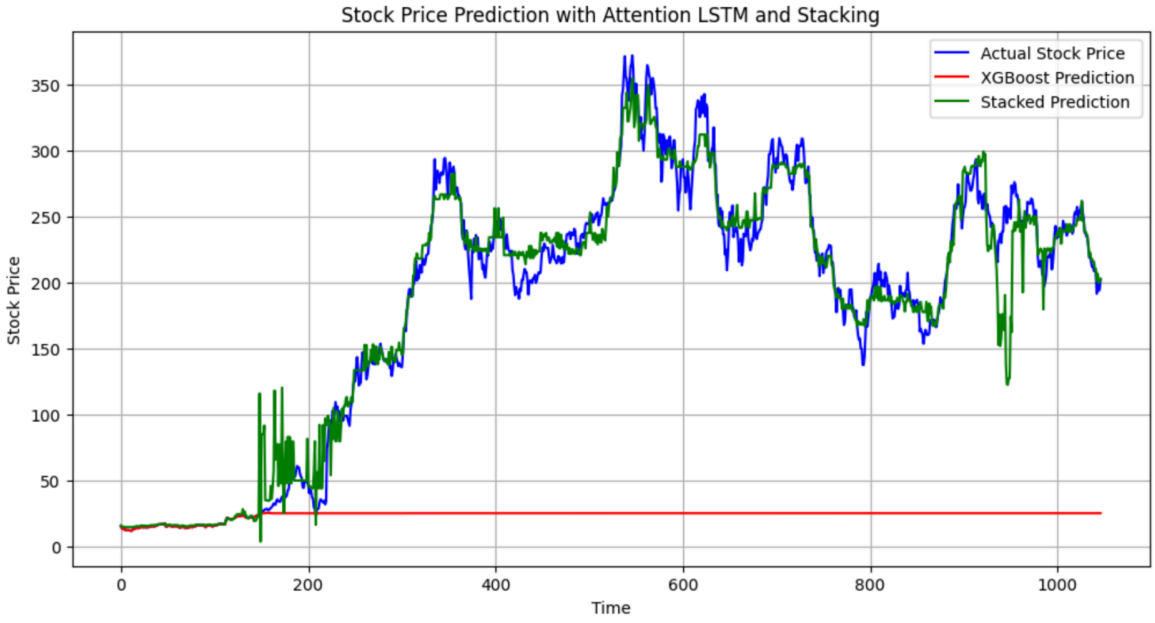
Nvidia: Hybrid Model (LSTM - XGBoost) with anomaly detection

Figure 4.2



Tesla: Hybrid Model (LSTM - XGBoost)

Figure 5.1



Tesla: Hybrid Model (LSTM - XGBoost) with anomaly detection

Figure 5.2

Our enhanced hybrid model, which combines Anomaly Detection, Attention-LSTM, and XGBoost, showed remarkable improvements in stock price prediction across various companies. We evaluated the model's performance using three key metrics: Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and R^2 Score. Table 1 presents a detailed comparison between our initial hybrid model and the enhanced version.

The improved model showed better performance in most cases, with especially significant improvements in RMSE and R^2 scores for volatile stocks like Tesla and Nvidia. The biggest improvement was in Nvidia's predictions, where the R^2 score rose from 0.72 to 0.96, showing how well our anomaly detection method works with highly variable data.

When compared with previous research:

1. Amin et al. (2024) reported ensemble model accuracy of 95.67% for Microsoft, outperforming traditional models like Gradient Boosting (74%) and Random Forest (82%). Our model follows this pattern, particularly achieving R^2 scores near or above 0.97 for companies like Apple, Google, and Tesla.
2. Nti et al. (2019) found LSTM-RNN and ARIMA to yield accuracies of 98.03% and 95.88% respectively. Our model achieved 96.83% accuracy for Apple, validating its strong predictive capability.
3. Alshammari et al. (2022) focused on Gulf market data with a limited set of indicators. Our approach differs by using a more comprehensive set of global market indicators, enhanced by anomaly detection and attention mechanisms

Unlike previous research that focused on specific regional markets with a limited set of indicators, our model uses a wider range of global market data. This more comprehensive approach, along with our advanced anomaly detection and attention mechanisms, helps our model perform better across different companies and economic situations.

The use of attention mechanisms and anomaly detection has been especially helpful in dealing with market noise and volatility. These improvements make our model more reliable and useful for real-world applications in today's constantly changing global financial markets.

CHAPTER 5

CONCLUSION

The aim of this research was to predict the stock prices for the largest tech companies (Google, Apple, Tesla, Nvidia, and Amazon), utilizing company financials and traditional economy signals (GDP growth, inflation rate, unemployment rate, and interest rates) as input. The study examined a new hybrid model (combination of LSTM and XGBoost) which integrated anomaly detection methodology to improve the robustness of the model.

The experimental results were very strong in all situations. The hybrid model predicted the stock prices for Google with R^2 scores of .97, Apple - .99, Tesla - .97, and Nvidia - .96. These results were in addition to historical LSTM models, the inputs also produced lower RMSE and MAE metrics, as well as providing an economic rationale and therefore the capability of recognizing complex market behaviour.

In conclusion, the results provide evidence of the value of traditional company financials, as well as macro data, to predict stock prices in the future.

Future Work

1. **Incorporation of Additional Financial Indicators** : Future iterations of the model could incorporate a wider range of technical indicators and market sentiment data, including MACD, Bollinger Bands, and social media analytics. These additions would enhance the model's ability to identify and predict market trends.

2. **Integration of Multi-Modal Data Sources** : The model could be enhanced to include different types of data, such as detailed financial reports, economic indicators, and social media sentiment analysis. This broader data integration would provide a more complete picture of market conditions.

3. **Extension to Portfolio-Level Predictions** : The current individual stock prediction model could be extended to analyze entire portfolios, enabling more sophisticated investment strategies based on predicted returns across multiple assets.

REFERENCES

1. Amin, M.S., E. H. Ayon, B. P. Ghosh, M.S. Chowdhury, M.S. Bhuiyan, R.M. Jewel, and A.A. Linkon. Harmonising Macro-Financial Factors and Twitter Sentiment Analysis in Forecasting Stock Market Trends. Journal of Computer Science and Technology Studies, pages 250-265, 2024.

2. Nti, I.K., A. F. Adekoya, and B. A. Weyori. Random Forest-Based Feature Selection of Macroeconomic Variables for Stock Market Prediction. American Journal of Applied Sciences, 16(1):1–12, 2019.

3. Botunac, I., J. Bosna, and M. Mateti´c. Optimization of Traditional Stock Market Strategies Using the LSTM Hybrid Approach. Electronics, 10(9):1–15, 2021.

4. Bukhari, K., A. K. Jadoon, M. Iqbal, and A. Arshad. Stock market prediction with time series data and news headlines: a stacking ensemble approach. Applied Sciences, 11(15):1–20, 2021.

5. Vora, S., R. Shaikh, K. Bhanushali, and P. Patil. BERT-LSTM model for sarcasm detection in code-mixed social media posts. International Journal of Advanced Research in Science, Engineering and Technology, 8(12):18754–18761, 2021.

6. Kumar, S., P. Singh, and R. Kumar. Stock price prediction: comparison of different moving average techniques using deep learning models. Journal of Ambient Intelligence and Humanized Computing, 12(1):1–12, 2021.

7. Khan, M.A., N. Abid, and N. Afridi. Predicting Stock Market Trends Based on Macroeconomic Indicators through Machine Learning Approach: A Case Study of KSE 100 INDEX. RASD Journal of Economics, 4(1):1–15, 2022.

8. Li, Y., Z. Yang, and X. Li. Coupling Macro-Sector-Micro Financial Indicators for Learning Stock Representations with Less Uncertainty. IEEE Transactions on Knowledge and Data Engineering, 33(8):3198–3211, 2021.

9. Patil, P.S., S.D. Pawale, and D.A. Borikar. Stock Price Prediction using LSTM. International Journal of Advanced Intelligence and Neural Networks, 2(1):1–8, 2021.

10. Agarwal, S., S. Kumar, and U. Goel. Social media and the stock markets: an emerging market perspective. Journal of Business Economics and Management, 22(1):1–23, 2021.

11. Ampomah, E.K., Z. Qin, and G. Nyame. Evaluation of Tree-Based Ensemble Machine Learning Models in Predicting Stock Price Direction of Movement. Information, 11(6):332, 2020.

12. Alshammari, B.M., F. Aldhmour, Z.M. AlQenaei, and H. Almohri. Stock market prediction by applying big data mining. Arab Gulf Journal of Scientific Research, 40(2):139-152, 2022.

13. Almehmadi, A. COVID-19 Pandemic Data Predict the Stock Market. Computer Systems Science & Engineering, 36(2):323–335, 2021.

14. Bagga, A.R., and H. Patel. Stock Market Forecasting using Ensemble Learning and Statistical Indicators. Journal of Engineering Research, 10(2):1–12, 2022.

15. Eachempati, P., and P.R. Srivastava. Training Multilayer Perceptron with Genetic Algorithms and Particle Swarm Optimization for Modeling Stock Price Index Prediction. Journal of Database Management, 34(1):1–20, 2023.

16. Liao, L., and T. Huang. The Impact of Social Media Sentiment on Stock Market Based on User Classification. Information Sciences, 580:620–638, 2023.

17. Mendoza-Urdiales, R.A., J.A. Núñez-Mora, R.J. Santillán-Salgado, and H. Valencia-Herrera. Twitter Sentiment Analysis and Influence on Stock Performance Using Transfer Entropy and EGARCH Methods. Entropy, 24(2):277, 2022.

18. Maqsood, H., M. Maqsood, S. Yasmin, I. Mehmood, J. Moon, and S. Rho. Analyzing the Stock Exchange Markets of EU Nations: A Case Study of Brexit Social Media Sentiment. Systems, 10(5):166, 2022.

19. Nti, K.O., A. Adekoya, and B. Weyori. Random Forest-Based Feature Selection of Macroeconomic Variables for Stock Market Prediction. American Journal of Applied Sciences, 16(2):27–41, 2019.

20. Pourroostaei Ardakani, S., N. Du, C. Lin, J.C. Yang, Z. Bi, and L. Chen. A federated learning-enabled predictive analysis to forecast stock market trends. Journal of Ambient Intelligence and Humanized Computing, 14(1):1–15, 2023.

21. Sarkar, M., E.H. Ayon, M.T. Mia, R.K. Ray, M.S. Chowdhury, B.P. Ghosh, M. Al-Imran, M.T. Islam, M. Tayaba, and A.R. Puja. Optimizing E-Commerce Profits: A Comprehensive Machine Learning Framework for Dynamic Pricing and Predicting Online Purchases. Journal of Computer Science and Technology Studies, 5(2):45-60, 2023.

22. Bhuiyan, M.S., E.H. Ayon, B.P. Ghosh, and A.A. Linkon. Transforming Customer Experience in the Airline Industry: A Comprehensive Analysis of Twitter Sentiments Using Machine Learning and Association Rule Mining. Journal of Computer Science and Technology Studies, 5(3):78-95, 2023.

23. Vlah Jerić, S. Comparing classification algorithms for prediction on CROBEX data. Business and Social Statistics, 6(2):45–58, 2020.

24. Yin, L., B. Li, P. Li, and R. Zhang. Research on stock trend prediction method based on optimized random forest. CAAI Transactions on Intelligence Technology, 8(1):100–115, 2023.

25. Zhang, H., Y. Chen, W. Rong, et al. Effect of social media rumors on stock market volatility: A case of data mining in China. Frontiers in Psychology, 13:1–15, 2022

APPENDIX

Appendix A: Technical Specifications

LSTM Configuration:

- Number of layers: 3
- Hidden units per layer: [128, 128, 64]
- Dropout rate: [0.3, 0.3, 0.3]
- Batch size: 32
- Learning rate: 0.001

XGBoost Parameters:

- n_estimators: 200
- Learning Rate: 0.05
- Max Depth: 6

Appendix B: Dataset Details

- Stock companies used (Amazon, Google, Apple, Tesla, Nvidia)
- Time period
- Source of data (e.g., Yahoo Finance, Kaggle)

Appendix C: Feature Description

- List of features used after preprocessing: 'Open', 'High', 'Low', 'Volume', 'Adj Close', 'Total Revenue', 'GDP', 'Net Income', 'Gross Profit', 'Cashflow from Financing'

Appendix D: Code

1) Importing Required Libraries

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import MinMaxScaler
from sklearn.metrics import mean_squared_error, mean_absolute_error
import tensorflow as tf
from tensorflow.keras.models import Model
from tensorflow.keras.layers import LSTM, Dense, Dropout, Input, Attention
import xgboost as xgb
import matplotlib.pyplot as plt
from sklearn.ensemble import GradientBoostingRegressor
from sklearn.ensemble import IsolationForest
from scipy import stats
```

2) Data Loading and Preprocessing

```
    def compute_rsi(series, window=14):
        delta = series.diff(1)
        gain = delta.where(delta > 0, 0).rolling(window=window).mean()
        loss = (-delta.where(delta < 0, 0)).rolling(window=window).mean()
        rs = gain / loss
        rsi = 100 - (100 / (1 + rs))
        return rsi

    data = pd.read_csv('/content/Processed_Data.csv')


    features = ['Open', 'High', 'Low', 'Volume','totalRevenue',
                'totalAssets',  'Unemployment Rate',
                'GDP', 'Inflation']
    data = data[features + ['Close']]
    data.dropna(inplace=True)

    data['MA7'] = data['Close'].rolling(window=7).mean()
    data['MA21'] = data['Close'].rolling(window=21).mean()
    data['EMA'] = data['Close'].ewm(span=20, adjust=False).mean()
    data['RSI'] = compute_rsi(data['Close'], window=14)
    data.dropna(inplace=True)

    def add_lagged_features(data, lag=1):
        for i in range(1, lag + 1):
            data[f'lag{i}'] = data['Close'].shift(i)
        return data

    data = add_lagged_features(data, lag=5)
    data.dropna(inplace=True)

    scaler = MinMaxScaler()
    scaled_data = scaler.fit_transform(data)
```

3) Anomaly Detection Using Isolation Forest

```
iso_forest = IsolationForest(contamination=0.05, random_state=42)
preds = iso_forest.fit_predict(data[features])
data = data[preds == 1]
```

4) Autoencoder for Feature Extraction

```
def build_autoencoder(input_dim):
    input_layer = Input(shape=(input_dim,))
    encoded = Dense(64, activation='relu')(input_layer)
    encoded = Dense(32, activation='relu')(encoded)
    encoded = Dense(16, activation='relu')(encoded)
    decoded = Dense(32, activation='relu')(encoded)
    decoded = Dense(64, activation='relu')(decoded)
    output_layer = Dense(input_dim, activation='linear')(decoded)

    autoencoder = Model(input_layer, output_layer)
    encoder = Model(input_layer, encoded)
    autoencoder.compile(optimizer='adam', loss='mse')
    return autoencoder, encoder

scaler = MinMaxScaler()
scaled_features = scaler.fit_transform(data[features])
autoencoder, encoder = build_autoencoder(scaled_features.shape[1])
autoencoder.fit(scaled_features, scaled_features, epochs=50, batch_size=32, verbose=1)
encoded_features = encoder.predict(scaled_features)

data_encoded = pd.DataFrame(encoded_features)
data_encoded['Close'] = data['Close'].values
```

5) Sequence Generation for LSTM Input

```
def create_sequences(data, seq_length=60):
    X, y = [], []
    for i in range(seq_length, len(data)):
        X.append(data[i-seq_length:i, :-1])
        y.append(data[i, -1])
    return np.array(X), np.array(y)

seq_length = 100
data_encoded.columns = data_encoded.columns.astype(str)
scaler = MinMaxScaler()
scaled_data = scaler.fit_transform(data_encoded)
X, y = create_sequences(scaled_data, seq_length)

train_size = int(len(X) * 0.7)
X_train, X_test = X[:train_size], X[train_size:]
y_train, y_test = y[:train_size], y[train_size:]
```

6) Attention-based LSTM Model Architecture and Training

```
def build_attention_lstm(input_shape):
    inputs = Input(shape=input_shape)
    x = LSTM(128, return_sequences=True)(inputs)
    attention = Attention()([x, x])
    x = LSTM(64)(attention)
    x = Dropout(0.3)(x)
    output = Dense(1)(x)
    model = Model(inputs, output)
    model.compile(optimizer='adam', loss='mean_squared_error')
    return model

lstm_model = build_attention_lstm((X_train.shape[1], X_train.shape[2]))
lstm_model.fit(X_train, y_train, epochs=50, batch_size=32, validation_split=0.1)

lstm_predictions = lstm_model.predict(X_test)
lstm_predictions = scaler.inverse_transform(np.concatenate((np.zeros((lstm_predictions.shape[0], scaled_data.shape[1]-1)), lstm_predictions), axis=1))[:,-1

data_encoded['LSTM_Prediction'] = np.nan
data_encoded.iloc[seq_length + len(X_train):, data_encoded.columns.get_loc('LSTM_Prediction')] = lstm_predictions
data_encoded.dropna(inplace=True)
```

7) XGBoost Regression Model

```
X_xgb = data_encoded.iloc[:, :-1].values
y_xgb = data_encoded['Close'].values
train_size_xgb = int(len(X_xgb) * 0.7)
X_train_xgb, X_test_xgb = X_xgb[:train_size_xgb], X_xgb[train_size_xgb:]
y_train_xgb, y_test_xgb = y_xgb[:train_size_xgb], y_xgb[train_size_xgb:]

xgb_model = xgb.XGBRegressor(objective='reg:squarederror')
xgb_model.fit(X_train_xgb, y_train_xgb)
y_pred_xgb = xgb_model.predict(X_test_xgb)

data_encoded['XGBoost_Prediction'] = np.nan
num_predictions = min(len(y_pred_xgb), len(data_encoded) - (seq_length + len(X_train)))
data_encoded.iloc[seq_length + len(X_train):seq_length + len(X_train) + num_predictions,
                  data_encoded.columns.get_loc('XGBoost_Prediction')] = y_pred_xgb[:num_predictions]
data_encoded.dropna(inplace=True)
```

8) Stacked Ensemble Model using Gradient Boosting

```
X_final = np.column_stack((lstm_predictions[:len(y_pred_xgb)], y_pred_xgb))
y_final = y_test_xgb[:len(y_pred_xgb)]

stacked_model = GradientBoostingRegressor()
stacked_model.fit(X_final, y_final)
final_predictions_stacked = stacked_model.predict(X_final)

rmse_stacked = np.sqrt(mean_squared_error(y_final, final_predictions_stacked))
mae_stacked = mean_absolute_error(y_final, final_predictions_stacked)
print(f"Stacked Model RMSE: {rmse_stacked}, MAE: {mae_stacked}")
```

9)Visualization of Results

```
plt.figure(figsize=(12, 6))
plt.plot(y_final, color='blue', label='Actual Stock Price')
plt.plot(y_pred_xgb, color='red', label='XGBoost Prediction')
plt.plot(final_predictions_stacked, color='green', label='Stacked Prediction')
plt.title('Stock Price Prediction with Attention LSTM and Stacking')
plt.xlabel('Time')
plt.ylabel('Stock Price')
plt.legend()
plt.grid()
plt.show()
```