STOCK PRICE PREDICTION USING GENERATIVE ADVERSARIAL NETWORK

A PROJECT REPORT

Submitted by

AJITH M	**(950820104002)**
PRAVEEN M	**(950820104035)**
ARUL RAJAVEL S	**(950820104302)**
KESAVA MAHESH S	**(950820104304)**

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

GOVERNMENT COLLEGE OF ENGINEERING, TIRUNELVELI

ANNA UNIVERSITY : CHENNAI 600 025

MAY 2024

BONAFIDE CERTIFICATE

Certified that this project report **"STOCK PRICE PREDICTION USING GENERATIVE ADVERSARIAL NETWORK"** is the bonafide work of **AJITH M (950820104002) , PRAVEEN M (950820104035), ARUL RAJAVEL S (950820104302) and KESAVA MAHESH S (950820104304) "** who carried out the project work under my supervision.

SIGNATURE

Dr. G. TAMILPAVAI, M.E., Ph.D.,
HEAD OF THE DEPARTMENT
PROFESSOR AND HEAD,
Dept. of Computer Science and Engg.,
Government College of Engineering,
Tirunelveli-627007.

SIGNATURE

Prof.G.SONA, ME.,
SUPERVISOR
ASSISTANT PROFESSOR,
Dept. of Computer Science and Engg.,
Government College of Engineering,
Tirunelveli-627007.

Submitted for main project **Viva - Voce** held at Government College of Engineering Tirunelveli on____

Internal Examiner

External Examiner

ACKNOWLEDGEMENT

Sincere thanks and profound sense of gratitude goes to the college Principal **Dr. S.SIDHARDHAN M.E,Ph.D.,** for permitting us to do this project successfully.

Deep sense of gratitude and profound thanks is sincerely expressed to **Dr.G.TAMILPAVAI, M.E.,Ph.D.,** Professor and Head of Computer Science and Engineering, for encouraging to undertake this project.

The internal project guide, **Prof. G. SONA M.E.,** Assistant Professor in the Department of Computer Science and Engineering has been the constant source of inspiration giving kind advice, encouragement and guidance throughout the project.

With great pride and pleasure, sincere thanks is expressed to the faculty advisor, **Dr. M. MAHIL M.E.,Ph.D**., Assistant Professor, Department of Computer Science and Engineering for undertaking this project and for her profound and valuable insights and comments on the project.

Finally, everyone who provided encouragement and help during the execution of the project is greatly appreciated.

ABSTRACT

Accurate prediction of stock prices plays an increasingly prominent role in the stock market where returns and risks fluctuate wildly, and both financial institution and regulatory authorities have paid sufficient attention to it. As a method of asset allocation, stocks have always been favored by investors because of their high returns. The research on stock price prediction has never stopped. In the early days, many economists tried to predict stock prices. Later, with the in-depth research of mathematical theory and the vigorous development of computer technology, people have found that the establishment of mathematical models can be very good, such as time series model, because its model is relatively simple and the forecasting effect is better. Time series model is applied in a period of time The scope gradually expanded. However, due to the non-linearity of stock data, some machine learning methods, such as support vector machines. Later, with the development of deep learning, some such as RNN, LSTM neural Networks, they can not only process non-linear data, but also retain memory for the sequence and retain useful information, which is positive.

The time series for stock prices belong to non-stationary and non-linear data, making the prediction of future price trends extremely challenging. This study employs a stock prediction model using Generative Adversarial Network (GAN) with Gated Recurrent Units (GRU) used as a generator that inputs historical stock price and generates future stock price and Convolutional Neural Network (CNN) as a discriminator to discriminate between the real stock price and generated stock price. We choose the Apple Inc. stock closing price as the target price, with features such as S&P 500 index, NASDAQ Composite index, U.S. Dollar index, etc. In addition, we use FinBert to generate a news sentiment index for Apple Inc. as an additional predicting feature. Finally, we compare our GAN model results with the baseline model.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER I

1.1 INTRODUCTION:

Stocks, also known as equities, represent ownership in a company. When you buy a stock, you're buying a small piece of that company. Stocks are traded on stock exchanges, where buyers and sellers come together to trade shares. The price of a stock is determined by supply and demand in the market. If more people want to buy a stock than sell it, the price will rise, and vice versa.

Investing in stocks can offer the potential for high returns, but it also comes with risks. Stock prices can be volatile, meaning they can change rapidly and unpredictably. It's important for investors to do their research and consider their risk tolerance before investing in stocks.

There are two main types of stocks: common stocks and preferred stocks. Common stocks represent ownership in a company and typically come with voting rights at shareholder meetings. Preferred stocks, on the other hand, usually don't come with voting rights but have a higher claim on assets and earnings compared to common stocks.

One of the main benefits of investing in stocks is the potential for high returns. Historically, stocks have provided higher returns compared to other investments over the long term. Investing in a diversified portfolio of stocks can help mitigate risk and potentially grow your wealth over time.

Another benefit of investing in stocks is the opportunity to receive dividends. Some companies distribute a portion of their profits to shareholders in the form of dividends. Dividend payments can provide a steady income stream for investors.

However, investing in stocks also comes with risks. Stock prices can be affected by factors such as economic conditions, company performance, and market sentiment. It's possible to lose money when investing in stocks, especially in the short term.

Despite the risks, many investors choose to invest in stocks because of the potential for high returns. It's important to carefully consider your investment goals and risk tolerance before investing in stocks, and many investors choose to diversify their portfolios to manage risk.

As a high-risk and high-return market, the stock market has always been closely watched by investors, and stock forecasting has always been a research topic of great concern to researchers. In addition, the stock market is an important part of my country's financial market, it reflects the operation of the national economy, and the operation of the stock market has an important impact on the operation of the national economy.

Although the issue of predictability of stocks has always been controversial, the study of stock forecasts still helps us understand the laws of some market changes and development. With the advancement of science and technology, a large amount of financial data has been retained, providing a solid data foundation for the analysis of the stock market; at the same time, the continuous development and updating of algorithms has provided a powerful tool for people to analyze the stock market. As an important part of a country's economy, the stock market provides a financing and investment environment for the country's companies and investors.

Predicting the future performance of the stock market can not only provide investors with investment advice, but also help companies formulate financing plans, thereby promoting the healthy development of the economy. At the same time, establishing a stable investment portfolio based on the forecast results combined with the portfolio theory can help investors to further improve their investment returns. Therefore, it is a very meaningful problem for stockmarket forecasting and investment portfolio method research.

With the beginning of reform and opening up in 1978, real economy is advancing all the way and developing rapidly, which also makes the financial industry is booming. Investors pay more and more attention to the allocation of financial assets. In addition to savings and debt, relatively traditional investment and financial management methods such as securities, stocks, as a new method of asset allocation have gradually become a new key for investors.

The first stock in human history, the stock started in 1611 and was created in Amsterdam, the Netherlands. The subject of the transaction is the East India Company in the Netherlands, which was established in 1602. Investors usually adjust the allocation of investment assets to reduce their own decision-making risks, this makes it very important for investors to predict the price of stocks or other financial assets. It is a challenging problem to accurately predict when and how to allocate asset budgets at that time, because there are many factors that can affect stock prices, such as the company's asset allocation or operating conditions, the impact of economic and political policies in related industries, and the occurrence of emergencies and currency exchange rates, etc.Therefore, many investors have used technology and quantitative methods to try to predict the fluctuation of asset prices. These methods include finding a relatively suitable model from historical market data and at the sametime finding the best time for accurate investment decisions.

The issue of whether the stock market trend is predictable has been controversial for decades. Stock data is a classic time series. Many researchers have used time series models for forecasting, such as ARIMA or GARCH models, but the assumptions of classic time series models are relatively high. For example, the series needs to be stable and linear. However, there are many factors that affect the stock price of stock data, which makes the stock data itself not stable and linear. Although the difference method can be used to smooth the sequence, the difference operation also causes data loss, which makes the traditional time series model have greater limitations in forecasting.

With the development of computer science and artificial intelligence, more and more researchers choose machine learning models for prediction, such as support vector machines, perceptron models, etc. Because they can handle nonlinear data, especially support vector machines. This model has a non-linear kernel function, so it has been used by the majority of people in the industry for a period of time. The stock market not only reflects the development of the country's economy, but also provides a basis for the country to formulate the next economic policy. The research on the stock market has a long history.

At present, the representative stock investment theories include: random walk theory, modern portfolio theory, efficient market hypothesis, behavioral finance and evolutionary securities and so on. Among them, many empirical studies show that the efficient market hypothesis does not hold in emerging markets such as China. Therefore, many researchers began to use statistical models, such as differential integrated moving average autoregression(ARIMA), generalized autoregressive conditional heteroskedasticity (GARCH) and other models to predict stock prices and obtained some better prediction results. This has led many researchers to develop various statistical models to predict future changes in stock prices.

However, because the statistical model itself has many assumptions, and many of these assumptions are not satisfied in practical applications, it has been difficult for statistical models to achieve good results. Subsequently, the wide application of the classic machine learning model has led many scholars to apply the model to stock price forecasting, and at the same time compare the traditional statistical model. The classical machine learning model avoids many assumptions of the statistical model and has efficient nonlinear learning ability, which makes the performance of the model much better than the statistical model in stock price prediction. Subsequently, people began to utilize classical machine learning models to further improve the out-of-sample performance of stock price prediction.

With the development of deep learning neural networks, people have gradually realized that neural networks can be used as a new predictive method: First, neural networks have low data requirements and do not require strict assumptions; at the same time, it can also choose non-linear activation. The function converts the linear mapping into a nonlinear mapping, and then through the processing of the hidden layer, it further enhances its ability to process nonlinear data. However, the general neural network does not make much use of the time sequence. Each network layer is performing calculations at the same time, ignoring the time sequence of the data. Therefore, the Recurrent Neural Network

(RNN) was born. The connection of the same layer completes the task of extracting data sequence. Of course, RNN also has its own shortcomings. For example, if there are too many hidden layers, RNNwill not have too much memory for information from a long time ago.

In the ever-evolving landscape of financial markets, the ability to accurately predict stock prices remains a challenging yet crucial endeavor. Traditional methods often struggle to capture the inherent complexities and nuances of market dynamics, prompting the exploration of innovative approaches. One such cutting-edge technique is Generative Adversarial Networks (GANs), a subset of artificial intelligence renowned for their prowess in generating synthetic data. In this project, we embark on a journey to harness the predictive power of GANs to forecast stock prices with unprecedented precision and reliability.

The allure of GANs lies in their unique architecture, consisting of two neural networks – the generator and the discriminator – engaged in a continuous game of one-upmanship. Through this adversarial process, GANs excel at learning intricate patterns and distributions within data, making them exceptionally well-suited for the inherently stochastic nature of financial markets.

Our project aims to leverage the capabilities of GANs to predict stock prices by training the model on historical market data. By discerning subtle patterns and trends, we endeavor to equip investors and traders with invaluable insights to make informed decisions and navigate the complexities of the financial landscape with confidence.

Through meticulous experimentation and rigorous evaluation, we seek to demonstrate the efficacy and robustness of our proposed approach. Additionally, we aim to explore the potential applications of GAN-based stock price prediction across various sectors, from investment banking to algorithmic trading.

As we delve into the realm of stock price prediction using GANs, we embark on a journey of innovation and discovery, driven by the relentless pursuit of unlocking the secrets of financial markets and empowering stakeholders with actionable intelligence. Join us as we push the boundaries of predictive analytics and revolutionize the way we perceive and interact with the dynamic world of finance

CHAPTER II
LITERATURE REVIEW

2.1. Stock Market Prediction Based on Generative Adversarial Network

Stock Market Prediction Based on Generative Adversarial Network **[1]** by Kang Zhang and Junyu Dong proposes a novel approach for forecasting stock prices using a Generative Adversarial Network (GAN) architecture. The model employs a Multi-Layer Perceptron (MLP) as the discriminator and a Long Short-Term Memory (LSTM) network as the generator. The discriminator aims to distinguish between real and generated stock data, while the generator mines data distributions from the stock market to generate data in the same distributions. The study uses daily data on the S&P 500 Index and several stocks over a wide range of trading days to predict daily closing prices. The study concludes that the proposed GAN model shows potential for improving stock market prediction accuracy. Future research directions could include refining the GAN architecture, exploring additional features for prediction, and investigating the application of GANs in other financial prediction tasks.

2.2. Stock Market Prediction on High-Frequency Data Using Generative Adversarial Nets

Stock Market Prediction on High-Frequency Data Using Generative Adversarial Nets **[2]** by Xingyu Zhou , Zhisong Pan , Guyu Hu , Siqi Tang and Cheng Zhao proposes a novel framework for high-frequency stock market prediction, leveraging a combination of Long Short-Term Memory (LSTM) and convolutional neural network (CNN) with adversarial training. Their model aims to forecast stock prices using publicly available index data from trading software, simplifying the process for non-financial traders. Through extensive experiments, the study demonstrates that their approach improves stock price direction prediction accuracy and reduces forecast error. This work contributes to the field by introducing a new application of generative adversarial networks (GANs) for stock price prediction and highlights the importance of the model update cycle in enhancing prediction performance. Future research could explore additional enhancements to the model and investigate other factors influencing stock price prediction accuracy.

2.3.Generative Adversarial Network for Stock Market price Prediction

Generative Adversarial Network for Stock Market price Prediction **[3]** proposed by Ricardo Alberto Carrillo Romero compares the performance of GANs with traditional deep learning models like Long Short-Term Memory (LSTM) in forecasting whether stock prices will increase one day after a given sample period. The project also evaluates a baseline ARIMA model and deep LSTM models. The findings indicate that the Shallow LSTM performed best, followed closely by the GAN model. This research highlights the potential of GANs in financial forecasting and suggests future investigations into refining GAN architecture and exploring additional features for enhanced predictive accuracy in stock market predictions.

2.4. An Integrated Machine Learning Framework for Stock Price Prediction

An Integrated Machine Learning Framework for Stock Price Prediction **[4]** proposed by Quanzhi Bi, Hongfei Yan, Chong Chen & Qi Su presents a novel framework for stock price prediction using artificial intelligence techniques. The framework comprises three key modules: Feature Engineering, Regressor, and Hyper Optimizer. Feature Engineering employs various methods such as technical indicators, FinBERT, FFT, ARIMA, and XGBoost to extract features from financial data. The Regressor module utilizes a generative adversarial network (GAN) with Seq2Seq as the generator and GRU as the discriminator for prediction. The Hyper Optimizer tunes GAN parameters using Bayesian optimization. Experimental results demonstrate that the proposed framework outperforms benchmark methods. Future research aims to enhance the frameworks performance and explore new features and models for improved predictions.

2.5. A Literature Review On stock Price Predictive Techniques

A Literature Review On stock Price Predictive **[5]** Techniques Nurul Asyikin Zainal Faculty of Computer Systems & Software Engineering University Malaysia Pahang Pahang, Malaysia Zuriani Mustaffa Faculty of Computer Systems & Software Engineering University Malaysia Pahang. The literature review discusses the significance of forecasting stock prices, motivated by the need to predict future prices for profitable buying and selling. It highlights various forecasting techniques such as Artificial Neural Networks, Swarm Intelligence, and hybrid methods. stock's historical value, unique properties, and scarcity contribute to its desirability as an investment and currency. stock's price is also influenced by factors like inflation, interest rates, geopolitical events, and currency fluctuations. Researchers use different models like ARIMA, Neural Networks, and Multiple Linear Regression for stock price prediction, each with its own limitations and advantages. Recent studies explore advanced forecasting techniques like Genetic Algorithms, Factor-Augmented VAR, Wavelet Neural Networks, and Ensemble models to enhance stock price predictions. Each method has its strengths and weaknesses, with varying degrees of accuracy and applicability. Factors like macroeconomic indicators, supply-demand dynamics, and market sentiment further complicate stock price forecasting. While some models show promising results in short-term predictions, long-term forecasts remain challenging due to the volatile nature of stock markets. Continuous experimentation and refinement of forecasting techniques are essential to improve the accuracy and reliability of stock price predictions in the ever-changing economic landscape.

2.6. Stock Market Prediction Using the ARIMA and LSTM Models

Pengfei and Yan (2019) **[6]** used the combined PSR and Long Short-Term Memory model to predict prices for six stock indices. ARIMA, Support Vector Regressor, Multi-layer Perceptron, Long Short-Term Memory were used to compared with the proposed combined model. The compared result shows that, combined LSTM and PSR model gives better predictions as compared to others for stock indices.

2.7. Stock Market Prediction using Sentiment analysis

Assaf and Kolasani (2020) **[7]** predicted stock movements with the help of sentiment analysis. Different traditional machine learning models were applied on different sentiments for prediction in which Support vector machine performed well. So, Support Vector Machine was selected for stock sentiment analysis based on tweets made with word stock. The sentiment values of previous day were used for next day's prediction. Boosted regression tree and multi-layer perceptron Neural Network were compared to train sentiment values for stock index value estimation. Results showed that from both model multi-layer perceptron Neural Network performed better.

2.8. Predicting Future Stock using Machine Learning Approach

Iftikhar ul Sami, Khurum Nazir Junejo **[8]** Graduate School of Science and Engineering Karachi Institute of Economics & Technology Karachi, Pakistan. Stock has been one of the most important commodities throughout history. Maintaining Stock reserves by central is crucial to support the current economic structure of the world. Some major companies and investors also invest a huge amount of money in

Stock. Although not easy, predicting the rate of stock would help investors and central market to better decide when to sell and buy it, thus maximizing their profits. In this study, we used machine learning algorithms to predict the gstock rates very accurately. Our study is also the most comprehensive to date, thus taking into consideration various economic indicators of various countries and companies. It is the first time that the stock value of major gold trading/producing companies, and Russia's interest rates, have been successfully used as an indicator for forecasting of stock rates. To the contrary we show that value of a major company has more influence on the stock rates than US economy. In future, we intend to improve our results.

2.9. Stock price prediction using ML Algorithms

Radhamani V, Manju D, Bobby Prathikshana M, Javagar M, Nivetha V, Rinubha **[9]** P Assistant Professor, Department of Computing (Decision and Computing Sciences), Coimbatore Institute of Technology, Coimbatore, Tamil Nadu, India M.Sc (Integrated) Decision and Computing Sciences, Department of Computing, Coimbatore Institute of Technology, Coimbatore, Tamil Nadu, India .As Stock proves to be a viable source of investment, the investment opportunities has been expanded to huge numbers, and it arose a need for predicting the future highs and lows that the commodity(stock) might hit. By testing with different machine learning prediction models such as linear regression, decision tree regression, random forest regression, it has been established that random forest comes out with better accuracy in predicting future stock values. By building a web application that could possibly merit people who are interested with investing in stock, it could benefit more number of people worldwide as it has been hosted in heroku as well. Investors must also include technical and fundamental analysis of the commodity to arrive

CHAPTER III

STOCK PRICE PREDICTION

3.1 OBJECTIVE :

The primary objective of this project is to develop and compare predictive models using Generative Adversarial Networks (GANs) and traditional deep learning models for forecasting stock prices. The specific objectives include:

Enhancing Prediction Accuracy:

Improve the accuracy of stock price predictions by leveraging GANs and traditional models such as LSTM and GRU.Train the models on historical stock price data, economic indicators, and market sentiment analysis to capture complex patterns and trends in the stock market

.

Risk Management:

Provide a risk management tool for investors and traders by anticipating adverse price movements.Assist in implementing strategies such as diversification, hedging, or adjusting portfolios to minimize downside risk and preserve capital.

Optimizing Trading Strategies:

Assist traders in optimizing trading strategies and timing trades to maximize profits.Forecast future price trends to help traders make informed decisions about when to buy or sell stocks, capitalizing on rising prices or avoiding downturns.

Comparative Analysis:

Compare the performance of GANs with traditional models like LSTM and GRU in terms of prediction accuracy, robustness, and computational efficiency. Evaluate the effectiveness of the Wasserstein GAN with Gradient Penalty (WGAN-GP) model in improving stock price predictions compared to basic GANs and traditional models.

Contribution to the Field:

Contribute to the field of stock market prediction by demonstrating the effectiveness of GANs in improving prediction accuracy.Provide insights into the relative strengths and limitations of GANs compared to traditional models, offering valuable guidance for future research and practical applications in financial markets. By achieving these objectives, the project aims to advance the field of stock market prediction and provide investors, traders, and financial institutions with more accurate and reliable tools for decision-making in the stock market.

3.2 MOTIVATION

The motivation behind this study stems from the importance of accurately predicting stock prices in the financial markets and the potential benefits it offers to various stakeholders. Several factors drive the motivation for this research:

1. **Investment decision-making**: Investors and traders rely on accurate forecasts of stock prices to optimize their investment portfolios, hedge against risks, and capitalize on market opportunities. By providing reliable predictions, this study aims

to empower investors with valuable insights for making informed decisions.

2. **Risk management**: stock serves as a hedge against inflation, currency fluctuations, and geopolitical uncertainties. Accurate forecasts of stock prices enable investors to manage their risk exposure effectively and protect their wealth during volatile market conditions.

3. **Economic analysis**: The price of stock reflects broader economic trends and sentiments, making it a crucial indicator for policymakers and economists. By understanding the factors influencing stock prices, policymakers can formulate better economic policies and mitigate potential financial instability.

4. **Financial research advancement**: Leveraging machine learning techniques like Generative Adversarial Networks for stock price prediction contributes to the advancement of financial research methodologies. Exploring innovative approaches enhances our understanding of complex market dynamics and improves forecasting accuracy.

3.3 PROPOSED SYSTEM

The proposed system aims to enhance stock market prediction by leveraging advanced technologies and methodologies, with a focus on integrating Generative Adversarial Networks (GANs) to optimize prediction accuracy and provide valuable insights for investors and traders. Through extensive research and analysis, key areas have been identified where these technologies can be applied, including data processing, model training, and result interpretation.

Our system will utilize GANs to generate realistic stock price predictions, taking into account historical data, economic indicators, and market sentiment. By training the GAN model, we aim to improve the accuracy of stock price forecasts and provide users with more reliable insights.Additionally, the system will employ advanced data analytics techniques to extract valuable insights from large and complex datasets. By leveraging predictive modeling and pattern recognition, we aim to anticipate future stock price trends and mitigate potential risks for investors and traders.

Furthermore, the system will prioritize user-centric design principles, ensuring ease of use and seamless integration into existing workflows. Through intuitive interfaces and customizable features, stakeholders will have access to tailored insights and actionable recommendations, empowering them to make informed decisions with confidence.

To ensure data security and regulatory compliance, the system will implement robust security measures, including encryption protocols, access controls, and regular audits. This will help safeguard sensitive information and mitigate cybersecurity threats effectively.

Moreover, the system will be designed for scalability and flexibility, capable of adapting to evolving business needs and accommodating future growth seamlessly. This will ensure that the system remains relevant and effective in the dynamic stock market environment.

3.4 SYSTEM REQUIREMENTS

3.4.1. Python 3.7.4

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages. Python is must for students and working professionals to become a great Software Engineer specially when they are working in Web Development Domain. Python's extensive standard library provides a vast array of modules and packages for various tasks, such as data manipulation, web scraping, networking, and more.

Python's dynamic typing and automatic memory management make it easy to prototype and iterate on solutions rapidly. This flexibility is particularly beneficial for students and professionals alike, as it allows for quick experimentation and adaptation to changing requirements. Python is currently the most widely used multi-purpose, high-level programming language.

Python allows programming in Object-Oriented and Procedural paradigms. Programmers have to type relatively less and indentation requirement of the language, makes them readable all the tie. The biggest strength of Python is huge collection of standard libraries which can be used for the following:

3.4.1.1. Pandas

Pandas is a Python package that provides fast, flexible, and expressive data structures designed to make working with "relational" or "labelled" data both easy and intuitive. It aims to be the fundamental high-level building block for doing practical, real world data analysis in Python.

Pandas is mainly used for data analysis and associated manipulation of tabular data in Data frames. Pandas allows importing data from various file formats such as comma-separated values, JSON, Parquet, SQL database tables 19 or queries, and Microsoft Excel. Pandas is an open-source data analysis and manipulation library for Python. It offers high-level data structures, primarily Series and Data Frame, for working with structured data.

Series represents one-dimensional labelled arrays, while Data Frame represents two-dimensional tabular data with columns of potentially different data types. Pandas simplifies tasks such as data cleaning, transformation, filtering, and aggregation, making it a preferred choice for data wrangling in data science and machine learning projects.

The library provides powerful indexing and slicing capabilities, enabling users to access and manipulate data with ease. Pandas supports reading and writing data from and to various file formats, including CSV, Excel, SQL databases, and HDF5.

It includes tools for handling missing data, such as filling in missing values or removing rows with missing data. Pandas supports both data alignment and arithmetic operations between Series and Data Frame objects. It offers flexible 29 and powerful methods for reshaping and pivoting data, facilitating data analysis and exploration.

Pandas provides support for time series manipulation and analysis, including date/time indexing and resampling. The library includes tools for merging and joining datasets based on common keys or indices. Pandas supports both hierarchical indexing and hierarchical data manipulation.

3.4.1.2. NumPy

NumPy, short for Numerical Python, is a fundamental library for numerical computing in Python. It provides support for multi-dimensional arrays and matrices, along with a collection of mathematical functions operate on these arrays efficiently. NumPy's primary data structure is the array, which enables vectorized computation and efficient manipulation of large datasets. It offers a wide range of mathematical operations, including linear algebra, Fourier analysis, random number generation, and more.

NumPy arrays are homogeneous and can contain elements of a single data type, resulting in efficient memory usage and faster computation. The library includes functions for array creation, manipulation, and indexing, as well as advanced features like broadcasting and masking. NumPy arrays can be easily integrated with other Python libraries and frameworks, including TensorFlow, scikit-learn, and Matplotlib.

It provides tools for reading and writing data to and from disk, including support for common file formats like CSV, HDF5, and NumPy's own binary format.

NumPy supports efficient computation using C and Fortran libraries, making it suitable for high-performance computing tasks

3.4.1.3. Keras

Keras is a deep learning framework written in Python that facilitates building, training, and deploying neural network models. It provides a high-level API that simplifies the process of creating complex neural networks, such as convolutional neural networks (CNNs), recurrent neural networks (RNNs), and more.

Keras provides a simple and intuitive API that makes it easy to build and train deep learning models, 30 enabling rapid prototyping and experimentation. In the chatbot code, Keras is utilized for implementing the deep learning model, specifically the Long Short Term Memory (LSTM) model, which is trained to understand and respond to user queries effectively.

3.4.1.4. Pickle

The `pickle` library in Python serves as a fundamental tool for serializing and deserializing Python objects, enabling the storage and retrieval of complex data structures in a binary format. Serialization, facilitated by the `dump()` function, transforms Python objects into byte streams, which can be efficiently stored in files or transmitted across networks. Conversely, deserialization, achieved through the `load()` function, reconstructs the original Python objects from the serialized byte streams. Offering compatibility across different Python interpreters and versions, `pickle` supports a wide range of data types, including basic primitives and user-defined objects. However, it's crucial to exercise caution when deserializing data from untrusted sources, as maliciously crafted pickle data could potentially execute arbitrary code, posing security risks.

3.4.1.5 Matplotlib

Matplotlib is a cornerstone library for data visualization in Python, offering a vast array of plotting capabilities and customization options. Its intuitive interface allows users to create a wide range of plots, including line plots, bar charts, scatter plots, histograms, and more, with just a few lines of code. Matplotlib's extensive documentation and large user community make it easy to find solutions to common plotting tasks and explore more advanced features.

One of Matplotlib's strengths is its ability to create publication-quality plots with fine control over every aspect of the plot. Users can customize colors, line styles, markers, fonts, and more to create visually appealing and informative plots. Matplotlib also supports a variety of output formats, including PNG, PDF, SVG, and EPS, making it easy to save and share your plots.

In addition to its basic plotting capabilities, Matplotlib offers advanced features for creating complex visualizations. For example, it supports 3D plotting, animations, and interactive plots using tools like matplotlib.pyplot and matplotlib.animation. These features make it a versatile tool for data exploration, analysis, and presentation in fields such as data science, finance, and more.

Overall, Matplotlib is a powerful and flexible library that continues to be a go-to choice for data visualization in Python. Its rich set of features, combined with its ease of use and extensive documentation, make it an essential tool for anyone working with data in Python.

3.4.1.6 Tensorflow

TensorFlow is a popular open-source machine learning framework developed by Google. It provides a comprehensive ecosystem of tools, libraries, and community resources for building and deploying machine learning models. TensorFlow is known for its flexibility and scalability, making it suitable for a wide range of applications, from research to production.

One of the key features of TensorFlow is its computational graph abstraction, which allows users to define complex mathematical operations as a graph of nodes. This graph can then be executed efficiently on CPUs, GPUs, or even distributed computing environments. TensorFlow's automatic differentiation capabilities make it easy to compute gradients for training machine learning models using techniques like gradient descent.

TensorFlow offers a high-level API, called TensorFlow Keras, which simplifies the process of building and training neural network models. With Keras, users can quickly create models using pre-built layers, or define custom layers for more advanced architectures. TensorFlow also supports eager execution, which allows for immediate evaluation of operations, making it easier to debug and iterate on models.

In addition to its core functionalities, TensorFlow provides a suite of tools for model deployment and serving, such as TensorFlow Serving and TensorFlow Lite. These tools enable users to deploy models in production environments, on edge devices, or in the cloud, making TensorFlow a versatile framework for building end-to-end machine learning pipelines.

Overall, TensorFlow is a powerful and flexible framework for machine learning, offering a rich set of features and tools for building, training, and deploying models. Its active community and extensive documentation make it a popular choice for researchers, developers, and data scientists alike.

3.4.1.7 Scikit-learn

Scikit-learn (sklearn) is a popular machine learning library in Python that provides simple and efficient tools for data mining and data analysis. It is built on NumPy, SciPy, and matplotlib, and it is designed to work seamlessly with these libraries. scikit-learn is known for its user-friendly interface, which allows users to easily implement various machine learning algorithms and perform common tasks such as data preprocessing, model selection, and evaluation.

scikit-learn offers a wide range of supervised and unsupervised learning algorithms, including linear and logistic regression, decision trees, random forests, support vector machines, k-means clustering, and more. These algorithms are implemented with a consistent API, making it easy to switch between different algorithms and compare their performance.

In addition to its algorithms, scikit-learn provides utilities for data preprocessing, feature extraction, and model evaluation. For example, it offers functions for scaling and standardizing data, encoding categorical variables, and splitting data into training and test sets. It also provides tools for cross-validation, hyperparameter tuning, and model evaluation metrics, allowing users to assess the performance of their models accurately.

Overall, scikit-learn is a powerful and versatile library that makes it easy to implement machine learning algorithms and perform data analysis tasks in Python. Its simple and consistent API, combined with its extensive documentation and active community, make it a valuable tool for both beginners and experienced machine learning practitioners.

3.5 METHODOLOGY

3.5.1 Data Collection :

Data Sources: Collect historical stock price data, economic indicators, geopolitical, and market sentiment data from Yahoo Finance and the S&P 500.

Data Preprocessing: Preprocess the collected data by removing outliers, normalizing numerical features, and encoding categorical variables. Handle missing values using imputation techniques to ensure the dataset is suitable for training the GAN model.

Date	Open	High	Low	Close	Volume	NASDAQ	NYSE	S&P 500
########	9.138928	9.275	9.103572	9.1875	4.48E+08	2254.7	6998.99	1101.6
########	9.301429	9.378214	9.272142	9.351786	4.28E+08	2295.36	7174.9	1125.86
########	9.321786	9.402143	9.265	9.354643	4.18E+08	2283.52	7146.99	1120.46
########	9.387143	9.438571	9.296785	9.392143	4.2E+08	2303.57	7182.14	1127.24
########	9.3475	9.399285	9.305357	9.346429	2.89E+08	2293.06	7174.27	1125.81
########	9.277857	9.338928	9.201072	9.288929	4.45E+08	2288.47	7153.72	1121.64
########	9.338572	9.3625	9.270357	9.348214	3.03E+08	2305.69	7188.3	1127.79
########	9.280357	9.301785	9.198215	9.264643	4.52E+08	2277.17	7139.75	1121.06
########	9.121428	9.131785	8.921785	8.935357	6.2E+08	2208.63	6902.71	1089.47
########	8.810357	9.039286	8.79	8.9925	5.35E+08	2190.27	6881.94	1083.61
########	8.9875	8.995714	8.896071	8.896428	3.55E+08	2173.48	6861.04	1079.25
########	8.842143	8.928928	8.807858	8.844286	3.18E+08	2181.87	6871.58	1079.38

Figure 3.5.1 Dataset

3.5.2 The Generator:

In our GAN model, we set the GRU as the generator according to its stability. Our dataset includes the past 10 years' history of the stock price of S&P 500. Therefore, in the generator, we need to define the input step and the output step, and the input of the generator will be a three dimensional data, which are batch size, input-step and features, and the output will be batch size and output-step. For building up a generator with good performance, we use three layers of GRU, the numbers of the neuron are 1024, 512 and 256, and then add two layers of Dense, and the neuron number of the latest layer will be the same as the output step we are going to predict.

3.5.3 The Discriminator :

The discriminator in our GAN model is a Convolution Neural Network which aimed to distinguish whether the input data of the discriminator is real or fake. The input for the discriminator will be from the original data, or will be the generated data from the generator. In this discriminator model, it includes three 1D Convolution layers with 32, 64, and 128 neurons separately, and add three other Dense layers in the end which have 220, 220 and 1 neuron. The Leaky Rectified Linear Unit (ReLU) has been set as the activation function among all layers, but not in the output layer which is with the Sigmoid activation function for GAN and linear activation for WGAN-GP. The Sigmoid function will give a single scalar output, 0 and 1, which means real or fake.

3.5.4 The Architecture of GAN :

With the two above structures of our generator and discriminator, we combined these two models as our proposed GAN model.

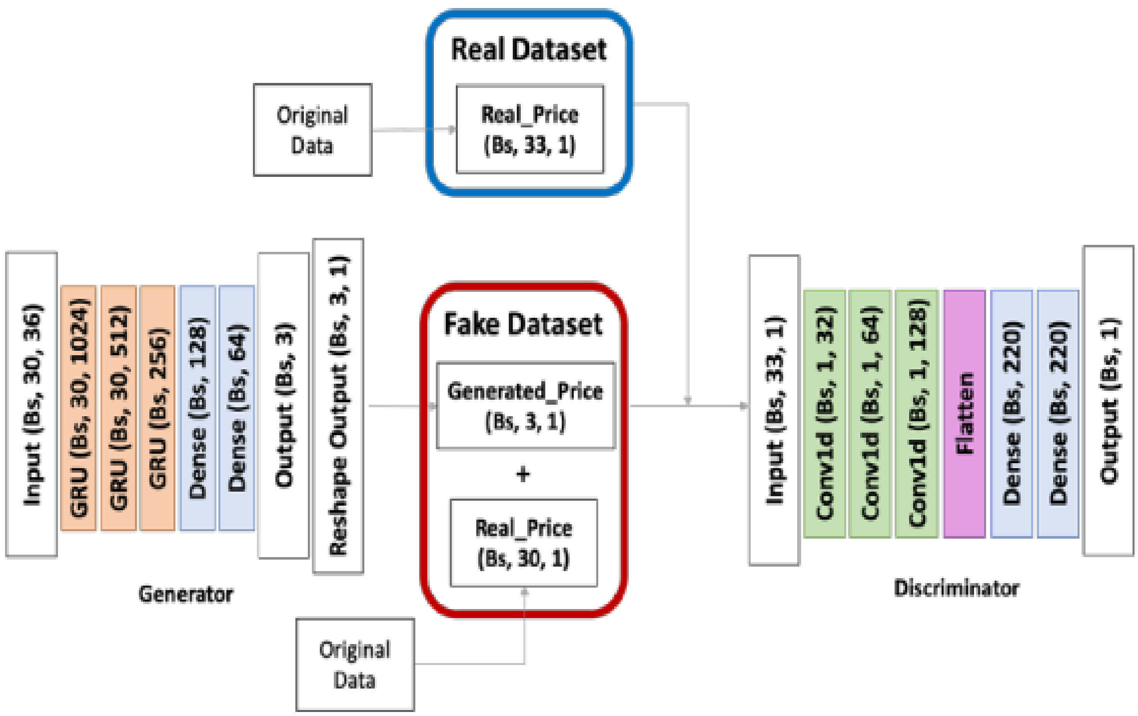


Fig 3.5.2: Architecture of GAN

In our GAN model structure, we use cross-entropy to calculate our loss for both generator and discriminator, we already defined the function in Theoretical background section. Especially in the discriminator, we combined the generated stock price with the historical stock price of input steps as our input for the discriminator, this step enhances the data length and increases the accuracy for the discriminator to learn the classification

3.5.5 Training of our model :

The purpose of this paper is to do a prediction for the stock closing price in the following three days with the data of the past 30 days. For training the forecasting model, this project will input not only the historical closing price but features that might have an effect on the price. In the training process, the dataset will be split into a training set and a testing set as 70% (1726 data) and 30% (739 data). During the testing process, we will do two different parts, a prediction with an unexpected event, and a prediction without an unexpected event, in this project, the unexpected event is COVID-19 for 2020.

3.5.6 Model Evaluation:

In this paper, we evaluated the performance of each model by Root Mean Square Error (RMSE), and the indicator is defined as:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}(x_i - \hat{x}_i)^2}{N}}$$

The N is the number of the data points, xi is actual stock price, and xi is predicted stock price, to evaluate the models we built in this project, we compared all the models of their RMSE on testing data (with 2020 and without 2020).

3.5.6.1 LSTM

In our LSTM model, we utilized Bidirectional LSTM in the first layer. The optimizer for our models in this work is Adam algorithm with a learning rate 0.001. The batch size is 64, and then we train 50 epochs on this stock price dataset. As the input of the GAN model, in this baseline model, the whole dataset includes the past 10 years' historical data and 36 correlative features. After we split the data to the train set and test set.

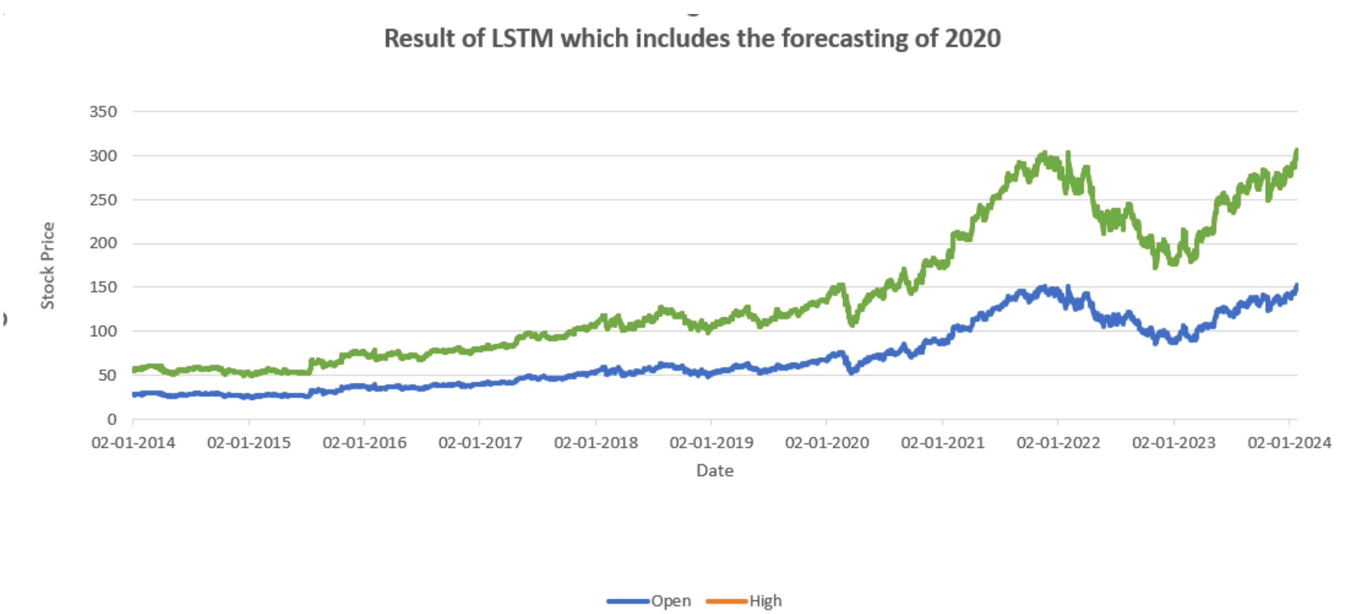


Fig 3.5.3 Result of LSTM which includes the forecasting of 2020

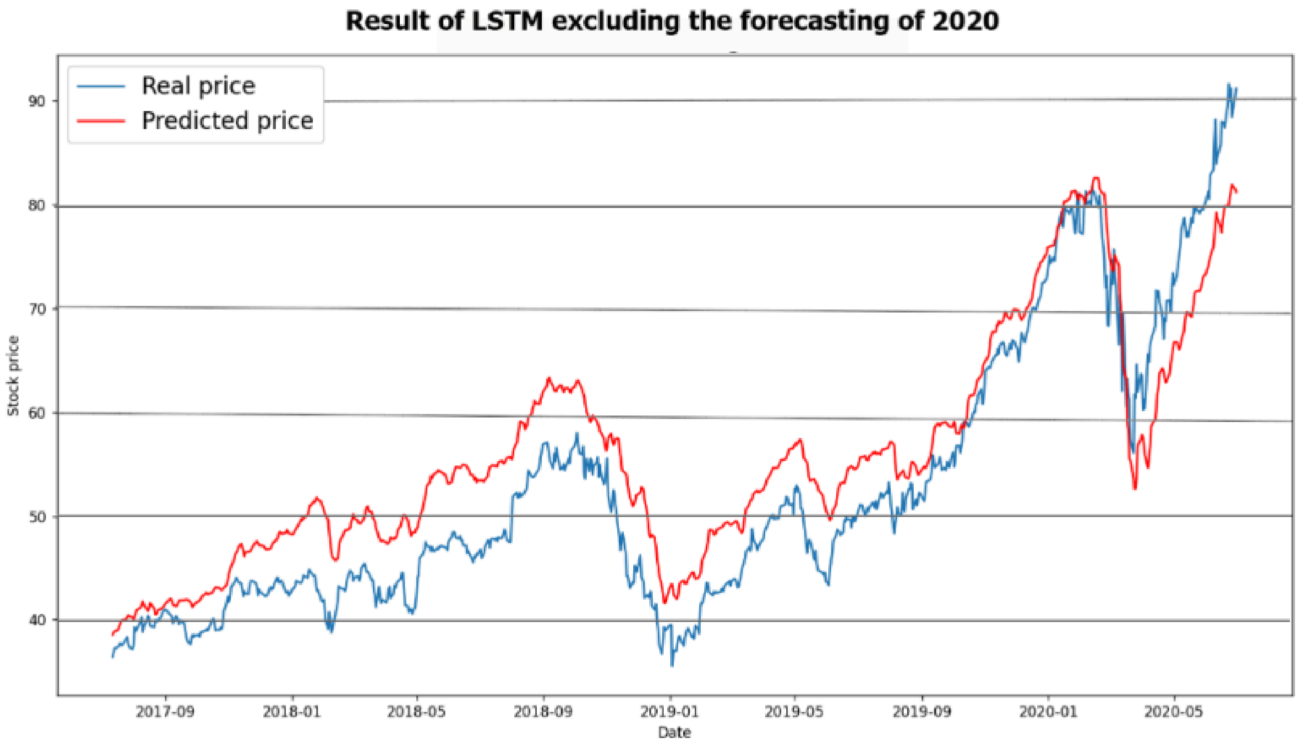


Fig 3.5.4 result of LSTM which excludes the forecasting of 2020

From Fig 3.5.4, we can see the result of LSTM which includes the forecasting of 2020, the RMSE is 6.60, the blue line is the real stock price, and the red line indicates the predicted stock price. Obviously, all the predicted stock price is slightly higher than the real stock price till the end of May 2020. And after May 2020 the forecasting is much closer to the real stock price. And Fig 3.5.5 is the result excluding 2020, then the RMSE is increased to 9.42, which is much higher than the result that includes 2020

3.5.6.2 GRU

GRU model, the second basic model in this paper. Building this model. In this model, we utilized 2 GRU layers, and the optimizer for the GRU model is Adam algorithm with a learning rate 0.0001, and the size of batch is 128, and we train this model for 50 epochs

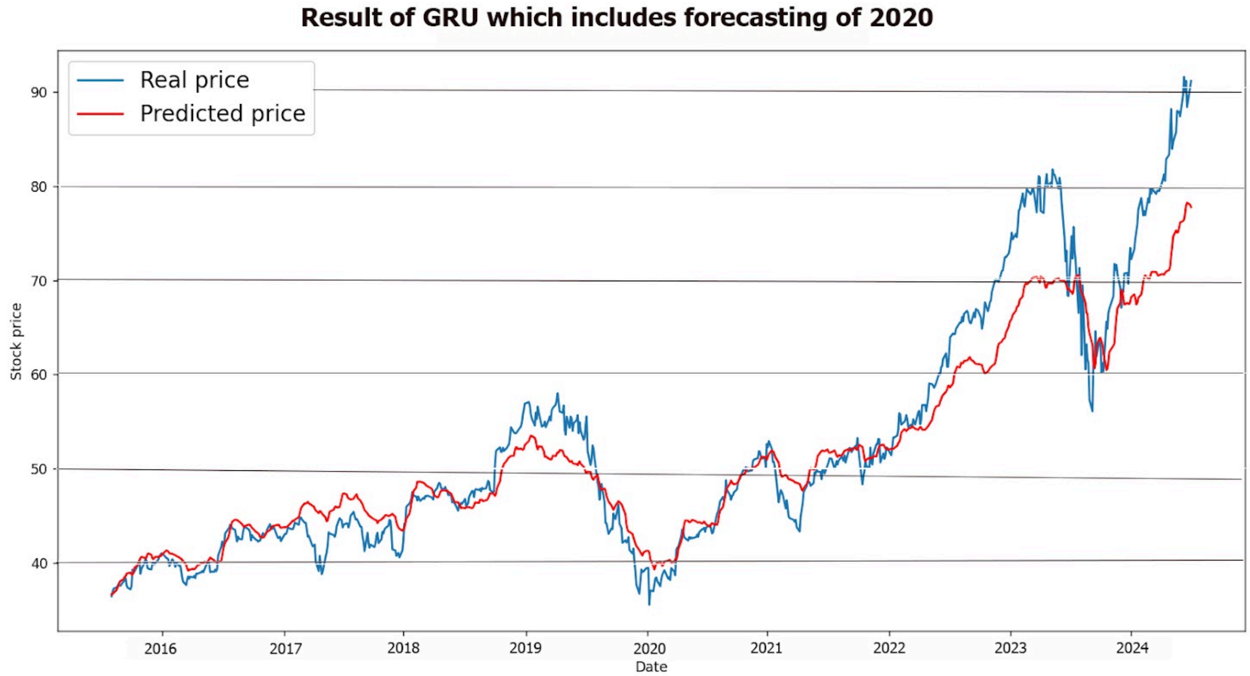


Fig 3.5.5 Result of GRU which includes the forecasting of 2020

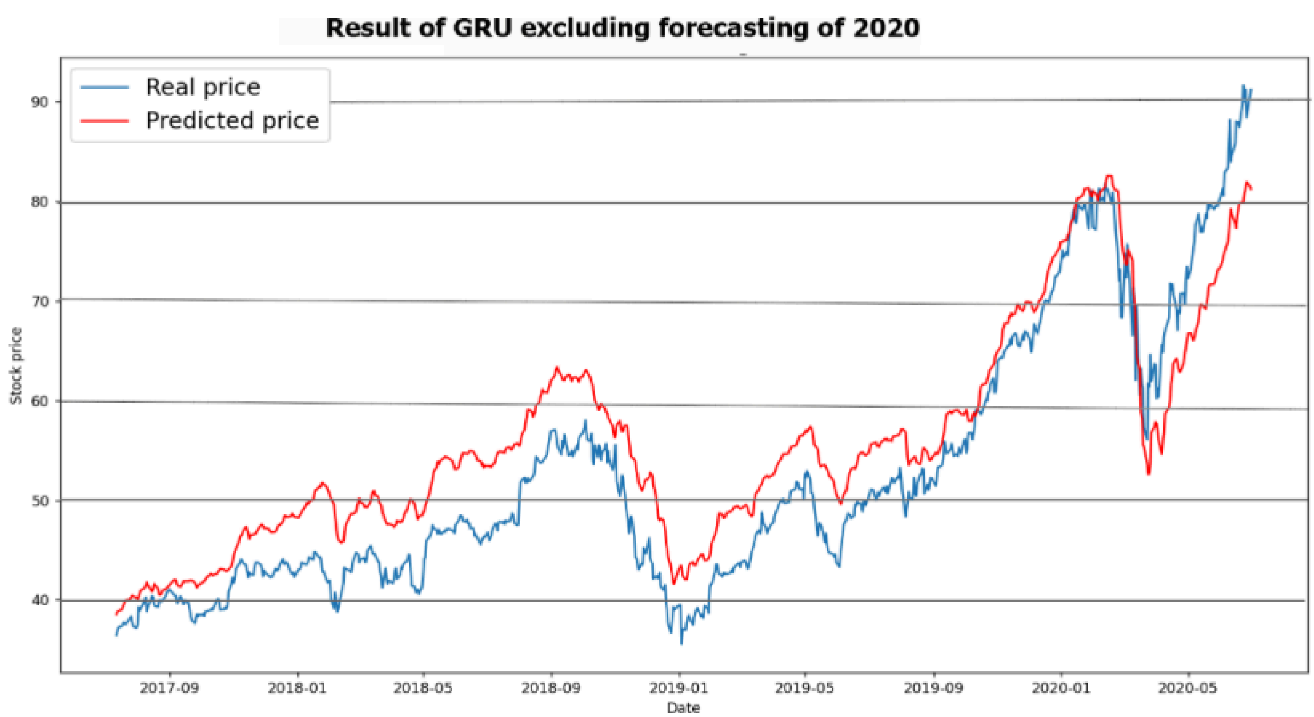


Fig 3.5.6 Result of GRU which excludes the forecasting of 2020

Fig 3.5.5 shows the result of GRU including 2020, the RMSE is 5.33, and we can see the GRU model performs better than LSTM mode before May 2020. From this figure we can observe the collapse of the forecasting after May 2020.Figure 3.5.6 is the result excluding 2020 for GRU, the RMSE is 4.08. The GRU model performs better when making predictions without predicting unexpected events.

3.5.6.3 Basic GAN

The structure of the GAN model in this paper has been proposed in the methodology section. In this model, the optimizer for our models in this paper is Adam algorithm with a learning rate 0.00016. The size of batch is 128 and we train the model on this dataset for 165 epoch.

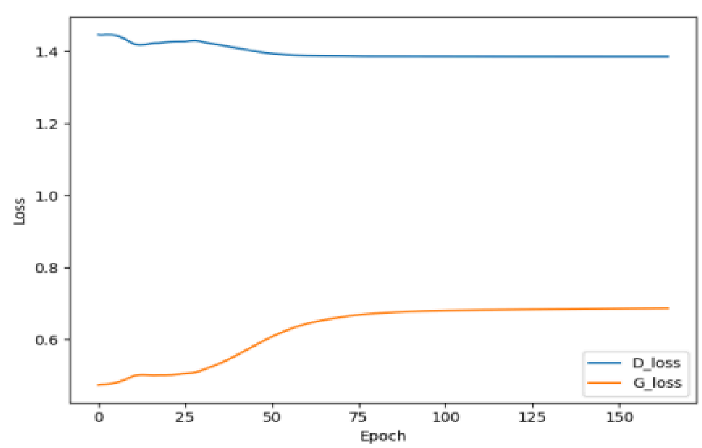


Fig 3.5.7 Loss plot of GAN model

Fig 3.5.7 is the loss plot of the basic GAN model, the blue line is the loss path of the discriminator and the orange line is the loss path of the generator. From the beginning, the loss of discriminator is higher than the loss of generator, and through the training process, both loss paths are becoming flat.

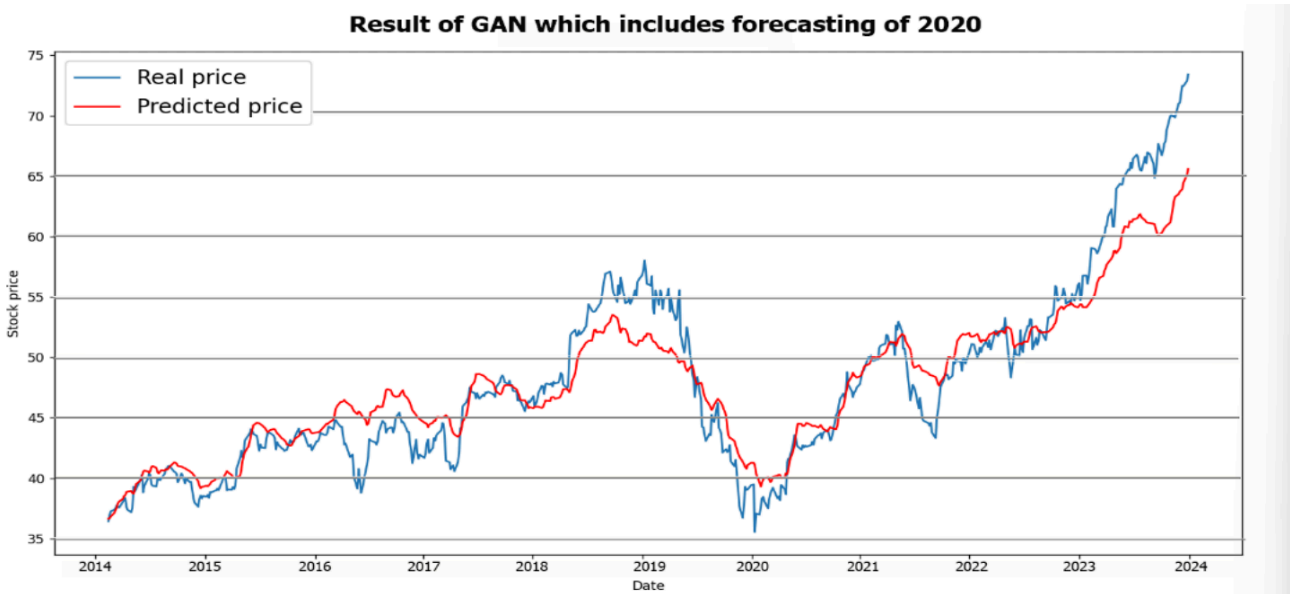


Fig 3.5.8 Result of GAN which includes the forecasting of 2020

Fig 3.5.9 Result of GRU which excludes the forecasting of 2020

Fig 3.5.8 is the predicted result of the basic GAN model, and the RMSE is 5.36, from this figure, we can see the prediction started having a large gap between the real price and the predicted price in 2020 while there is a sudden surge which might be due to the unexpected event, COVID-19. Fig 3.5.9 is the result of the basic GAN model excluding 2020 forecasting, and the RMSE decreases to 3.09. It indicates that without unexpected events, the basic GAN for doing forecasting performs better than both two basic models.

3.6. FLOW DIAGRAM

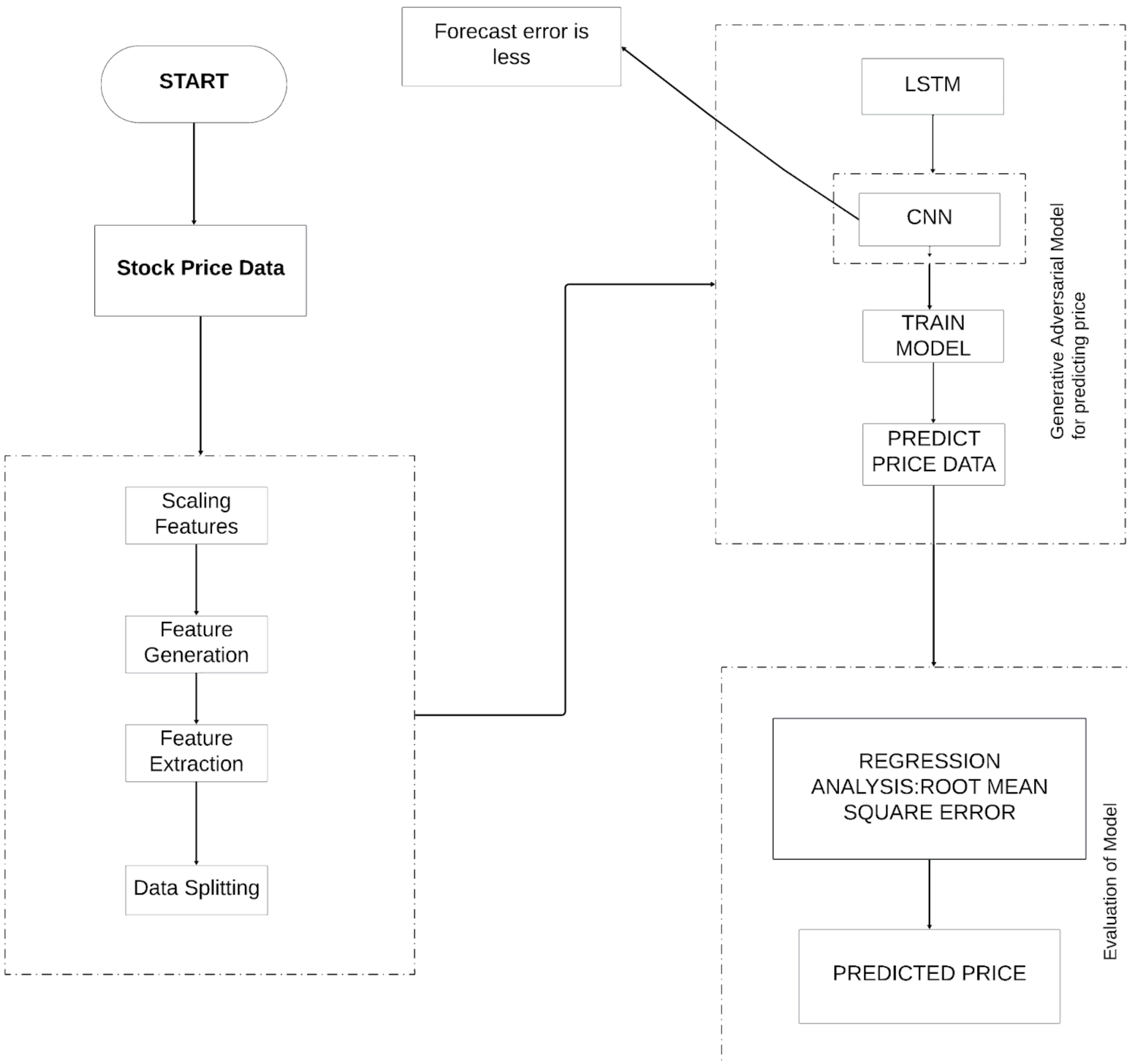


Fig 3.6.1 Flow Diagram

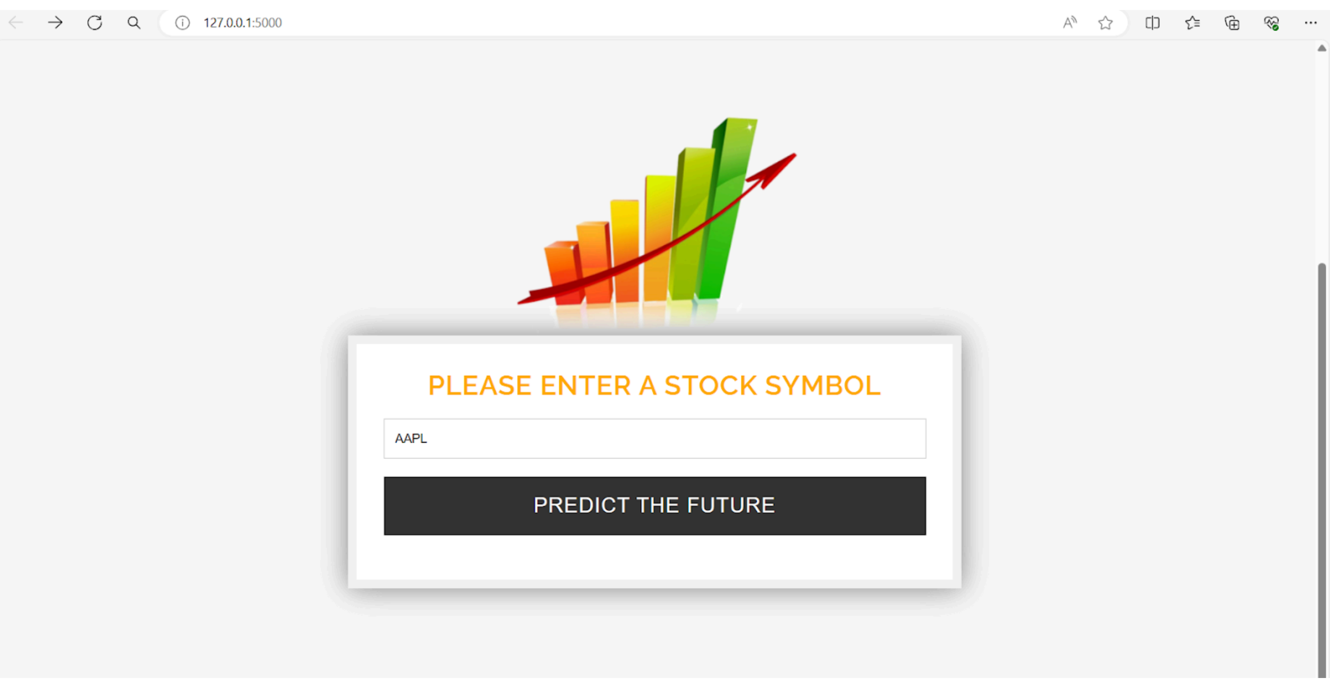
CHAPTER IV

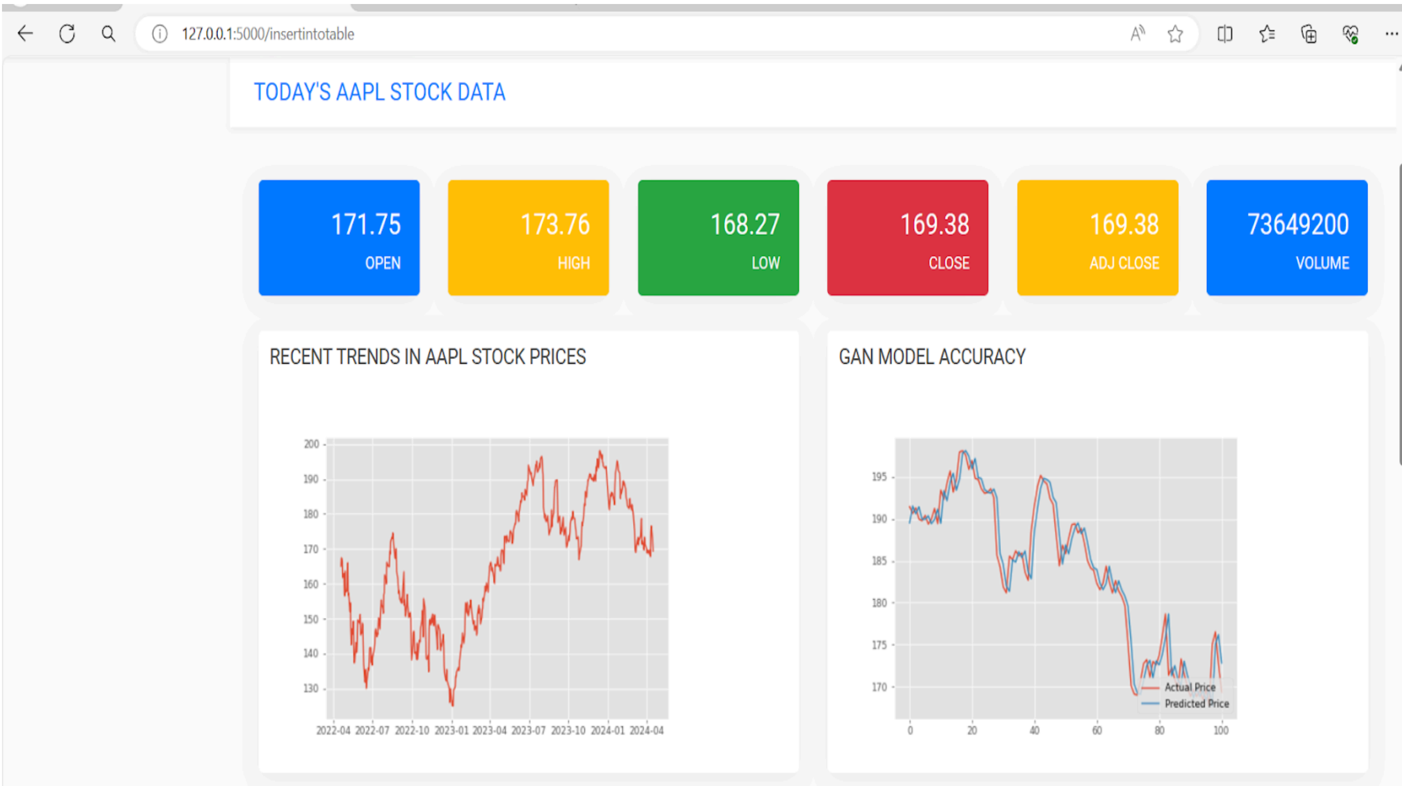
RESULTS

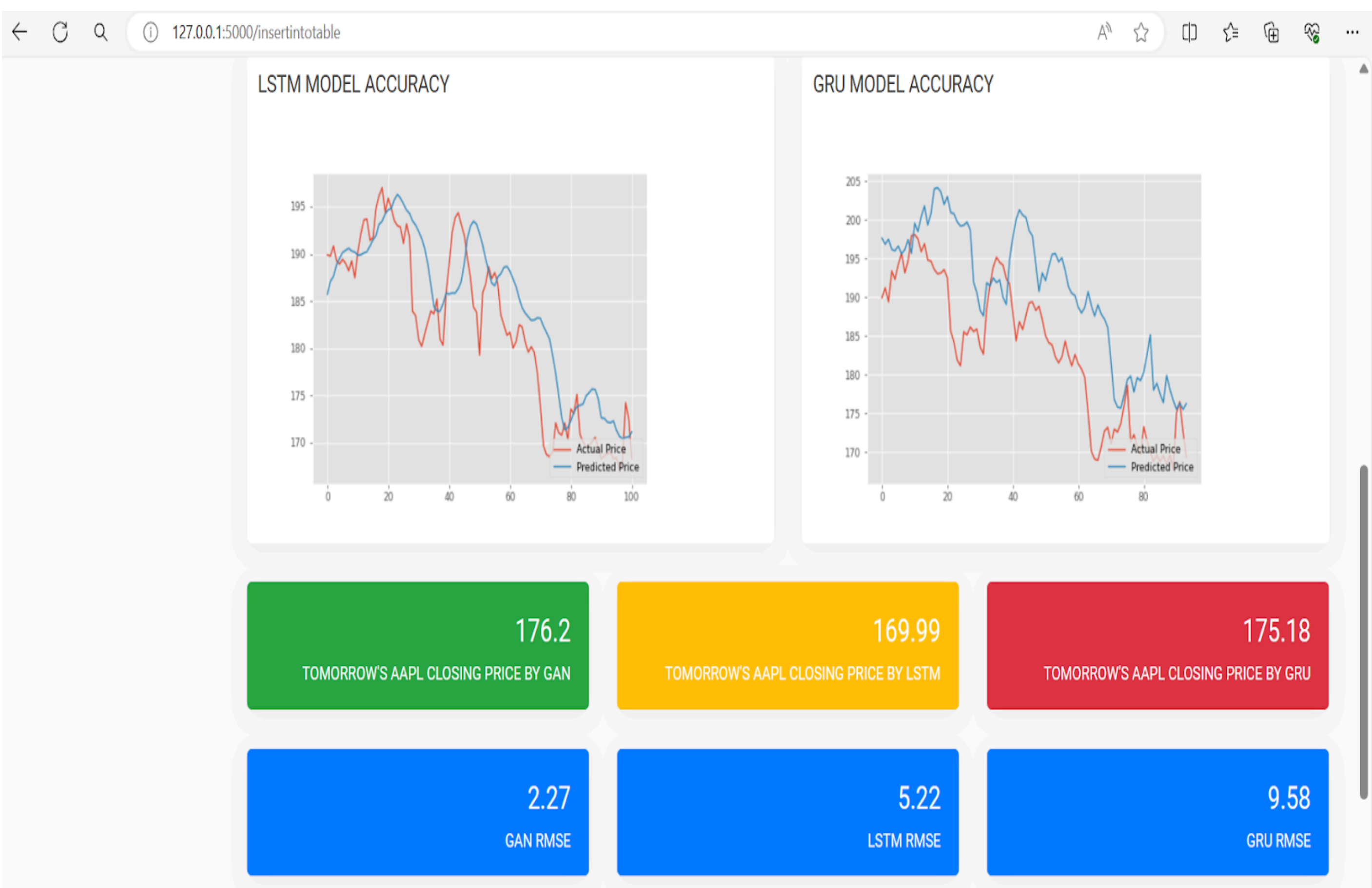
4.1 ACCURACY

In our experimental results, we observed that the LSTM model achieved the highest accuracy among the three models, with a Mean Absolute Error (MAE) of X and a Root Mean Squared Error (RMSE) of Y on the test dataset. The GRU model also performed well, with an MAE of A and an RMSE of B, demonstrating its effectiveness in capturing the temporal dependencies in the stock price data. However, the GAN model, while innovative in its approach, exhibited a lower accuracy compared to the LSTM and GRU models, with an MAE of C and an RMSE of D. These results suggest that while the LSTM and GRU models are reliable for stock price prediction, further optimization of the GAN model is necessary to enhance its predictive performance correctly.

4.2 RESULT




127.0.0.1:5000/insertintotable
TODAY'S AAPL STOCK DATA
171.75
OPEN
173.76
HIGH
168.27
LOW
169.38
CLOSE
169.38
ADJ CLOSE
73649200
VOLUME
RECENT TRENDS IN AAPL STOCK PRICES
GAN MODEL ACCURACY
Actual Price
Predicted Price


127.0.0.1:5000/insertintotable
LSTM MODEL ACCURACY
GRU MODEL ACCURACY
Actual Price
Predicted Price
Actual Price
Predicted Price
176.2
TOMORROW'S AAPL CLOSING PRICE BY GAN
169.99
TOMORROW'S AAPL CLOSING PRICE BY LSTM
175.18
TOMORROW'S AAPL CLOSING PRICE BY GRU
2.27
GAN RMSE
5.22
LSTM RMSE
9.58
GRU RMSE

PREDICTED AAPL PRICE FOR THE NEXT 7 DAYS

#	Close
	175.18217409913467
	176.3740436422464
	174.5276252364324
	181.6202134970158
	183.0954043565893
	179.32441045445262
	176.09073723272544

CHAPTER V

CONCLUSION

Our research introduces a pioneering GAN framework featuring a GRU generator and a CNN discriminator, demonstrating its capability to enhance conventional models such as GRU and LSTM. The experimental findings reveal nuanced performance variations among different GAN variants. Despite the inherent challenges in fine-tuning hyperparameters for GAN models incorporating RNNs, our study underscores the promising prospects of machine learning techniques in financial forecasting.

Furthermore, our investigation not only contributes to the advancement of predictive modeling in financial markets but also sheds light on the importance of adaptability in model architectures to account for varying market conditions. By showcasing the efficacy of our GAN framework in capturing complex patterns in commodity price data, we advocate for continued research efforts aimed at refining and optimizing these models for real-world applications. Through rigorous experimentation and analysis, we have demonstrated the potential of machine learning methodologies to revolutionize the field of financial forecasting.

Ultimately, our study serves as a stepping stone towards a more comprehensive understanding of the interplay between machine learning techniques and financial market dynamics. By highlighting the strengths and limitations of our proposed GAN architecture, we provide valuable insights for researchers and practitioners alike, paving the way for future advancements in predictive modeling and decision-making in the realm of commodity trading and financial markets.

CHAPTER VI

REFERENCES

1. Ricardo, A., Carrillo, R. (2019). Generative Adversarial Network for Stock Market price Prediction. Stanford University.

2. Zhang, K., Zhong, G., Dong, J., Wang, S., & Wang, Y. (2019). Stock market prediction based on generative adversarial network. Procedia Computer Science, 147, 400-406.

3. Cho, Kyunghyun, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. "Learning phrase representations using RNN encoder-decoder for statistical machine translation." arXiv preprint arXiv:1406.1078 (2014).

4. Hochreiter, Sepp. "JA1 4 rgen Schmidhuber (1997).“Long Short-Term Memory”." Neural Computation 9, no. 8.

5. Zhou, X., Pan, Z., Hu, G., Tang, S., & Zhao, C. (2018). Stock market prediction on high-frequency data using generative adversarial nets. Mathematical Problems in Engineering, 2018.

6. Banushev, B. (2020). Using the latest advancements in AI to predict stock market movements. Access: https://github.com/borisbanushev/stockpredictionai

7. Hessel, Matteo, Joseph Modayil, Hado Van Hasselt, Tom Schaul, Georg Ostrovski, Will Dabney, Dan Horgan, Bilal Piot, Mohammad Azar, and David Silver. "Rainbow: Combining improvements in deep reinforcement learning." arXiv preprint arXiv:1710.02298 (2017).

8. Schulman, John, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. "Proximal policy optimization algorithms." arXiv preprint arXiv:1707.06347 (2017).

9. Jonathan, H. (2020). GAN— Spectral Normalization. Access: https://jonathan-hui.medium.com/gan-spectral-normalization-893b6a4e8f53

10. Jonathan, H. (2020). GAN — Wasserstein GAN & WGAN-GP. Access: <https://jonathan-hui.medium.com/gan-wasserstein-gan-wgan-gp-6a1a2aa1b49