Stock price prediction using Generative adversarial network

Project-II (MA57012) report submitted to

Indian Institute of Technology,

in partial fulfilment for the award of the degree of

Integrated Master of Science

in

Mathematics and Computing

by

Harshit Yadav

(18MA20016)

Under the supervision of

Professor Geetanjali Panda

Department of Mathematics

Indian Institute of Technology,

Spring Semester, 2022-2023

DECLARATION

I certify that

(a) The work contained in this report has been done by me under the guidance of my supervisor.

(b) The work has not been submitted to any other Institute for any degree or diploma.

(c) I have conformed to the norms and guidelines given in the Ethical Code of Conduct of the Institute.

(d) Whenever I have used materials (data, theoretical analysis, figures, and text) from other sources, I have given due credit to them by citing them in the text of the thesis and giving their details in the references. Further, I have taken permission from the copyright owners of the sources, whenever necessary.

Date: (Harshit Yadav)
Place: Kharagpur (18MA20016)

DEPARTMENT OF MATHEMATICS
INDIAN INSTITUTE OF TECHNOLOGY,
KHARAGPUR - 721302, INDIA

CERTIFICATE

This is to certify that the project report entitled "**Stock price prediction using Generative adversarial network**" submitted by **Harshit Yadav** (Roll No. 18MA20016) to Indian Institute of Technology, towards partial fulfilment of requirements for the award of degree of Integrated Master of Science in Mathematics and Computing is a record of bona fide work carried out by him under my supervision and guidance during Spring Semester, 2022-2023.

Date:
Place: Kharagpur

Professor Geetanjali Panda
Department of Mathematics
Indian Institute of Technology,
Kharagpur - 721302, India

Abstract

Name of the student: **Harshit Yadav** Roll No: **18MA20016**

Degree for which submitted: **Integrated Master of Science**

Department: **Department of Mathematics**

Thesis title: **Stock price prediction using Generative adversarial network**

Thesis supervisor: **Professor Geetanjali Panda**

Month and year of thesis submission:

The subject of deep learning is intriguing. Due to its broad potential, it has been applied in many different contexts. Examples include high-frequency trading, portfolio optimisation, fraud detection, and risk management, all of which are crucial to society's financial sector. One of the most popular and valuable fields in finance is stock market forecasting. In this study, we propose a stock prediction model that uses a convolutional neural network (CNN) as a discriminator to distinguish between the real stock price and generated stock price and a generative adversarial network (GAN) with gated recurrent units (GRU) used as a generator that inputs historical stock price and generates future stock price. We select the closing price of Apple Inc. stock as the target price, taking into account factors like the SP 500 index, NASDAQ Composite index, U.S. Dollar index, etc. As an additional prediction tool, we also use FinBert to produce a news sentiment index for Apple Inc. Finally, we contrast the outcomes of our GAN model with the baseline model.

Keywords: Stock prediction; Generative adversarial network; WGAN-GP; Natural language processing

Acknowledgements

For the work done in this thesis, I would like to thank Professor Geetanjali Panda (Department of Mathematics, IIT Kharagpur) for all her guidance, support, and freedom to choose the topic of interest throughout the project. Most of the ideas in this work result from the weekly brainstorming and reading of papers/articles from the internet. I am also thankful to my family and friends, who constantly supported me throughout the semester in such difficult times. Without them, this work would not have been possible. I am highly thankful to the Department of Mathematics, IIT Kharagpur, for supporting this project to successful completion.

Date: (Harshit Yadav)

Place: Kharagpur (18MA20016)

Contents

Chapter 1

Introduction

Predicting stock prices is an intriguing and difficult topic. Numerous studies have demonstrated the predictability of stock price. Time series forecasts, like stock price forecasting, utilise a number of well-known algorithms like Long Short-Term Memory (LSTM) and ARIMA. One of the most potent models is the generative adversarial network (GAN), which uses an adversarial generator and discriminator to produce output that is more correct. GAN is frequently employed in the creation of images but not in the prediction of time series. Their conclusion is contradictory given the paucity of studies on time series prediction using GAN and the findings of their study. Therefore, in this paper, we choose to use GAN to forecast stock prices while also determining whether an adversarial system can enhance time series prediction. We will contrast the conventional models, Wasserstein GAN with Gradient Penalty (WGAN-GP), LSTM, and GRU with the fundamental GAN. The ability to accurately anticipate stock price complexities depends on a variety of factors. Therefore, we must include as much information as we can. In order to gather additional data, we combine a variety of conventional technical indicators, such as MACE and MAE, as well as a typical mix of underlying assets, such as commodities, currencies, indices, VIX, and more. Additionally, we employ Fourier transform methods to identify the general trend of price changes.Additionally, we analyse the market's financial news sentiment using the most recent Natural language processing (NLP) model, FinBert. We make an effort to forecast Apple.Inc.'s price alterations. We will utilise daily closing prices for the calculation from July 1, 2010, to June 30, 2020 (seven years for training and two years for validation).

Chapter 2

Theoretical background

2.1 Long short-term memory

The long short-term memory (LSTM) recurrent neural network (RNN) architecture was first proposed by Hochreiter and Schmidhuber in 1997 to address the vanishing gradient problem in conventional RNNs. LSTM incorporates feedback connections and can be applied to data sequences. Its fundamental parts are an input gate, an output gate, and a forget gate, allowing it to modify the state of the cell by deleting or adding information. LSTM has a unique internal structure that overcomes the vanishing gradients and exploding gradients issues of RNN. Today, LSTM is a powerful technique that can process, categorize, and make predictions based on time series data.

2.2 Gated recurrent unit

The gated recurrent unit (GRU) is a type of RNN that evolved from LSTM and uses gating methods to regulate the flow of information between cells in the network. GRU was introduced in 2014 by Kyunghyun Cho et al. and uses an update gate and reset gate to separate information to be kept and discarded. GRUs overcome the disappearing and exploding gradient issues in conventional RNNs, and have fewer parameters than LSTM due to the absence of one gate. However, GRUs cannot store long-term or short-term memories in the concealed state as they lack the cell state from LSTM. Recent studies have shown that GRUs outperform LSTM on some smaller and less frequent datasets.

2.3 Generative adversarial network

Generative adversarial network (GAN) is a minimax problem based on noncooperative games with zero-sum outcomes. GAN consists of a generator and a discriminator. The generator aims to produce instances that appear as realistically as possible, while the discriminator's task is to determine if the examples are genuine or fabricated. GAN has received significant attention in the deep learning community, with various strategies being created to improve outcomes by modifying the structure and loss function. Different types of GANs, such as Conditional GAN, Wasserstein GAN, WGAN with Gradient Penalty, Cycle GAN, PGGAN, and SAGAN, have been proposed by altering the structure or adding regularization to the loss function.

2.4 Basic Generative adversarial network

The cross-entropy loss is utilized by the GAN model to minimize the difference between two distributions during training, which is similar to minimizing the KL-JS divergence. In contrast, the loss function for the original GAN is based on KL-JS divergence. In this project, the objective function for the discriminator is to maximize the probability of correctly labeling the samples.

$$\hat{\mathbf{V}} = \frac{1}{m}\sum_{i=1}^{m} logD(y^i) + \sum_{i=1}^{m}(1 - logD(G(x^i))) \tag{2.1}$$

and after that, we train the generator to minimise the following objective function:

$$\hat{\mathbf{V}} = \frac{1}{m}\sum_{i=1}^{m}(1 - logD(G(x^i))) \tag{2.2}$$

where x is the input data for generator, y is the target from the real dataset, $G(x^i)$ is the generated data (fake target) from the generator. For present the calculating through the training process in GAN, the loss function of discriminator is:

$$-\frac{1}{m}\sum_{i=1}^{m} logD(y^i) - \frac{1}{m}\sum_{i=1}^{m}(1 - logD(G(x^i))) \tag{2.3}$$

The loss function of generator is:

$$-\frac{1}{m}\sum_{i=1}^{m} logD(G(x^i)) \tag{2.4}$$

2.5 WGAN-GP

The basic GAN's discriminator is not strong enough, and the training process can be unreliable and slow. To address this issue, it is suggested to use WGAN-GP to stabilize and improve GAN training. WGAN-GP proposes the Wasserstein distance as a solution. The Wasserstein distance, also known as the Earth-Mover Distance (EMD), measures the minimal amount of mass required to convert one data distribution into another. The infinium for any transport plan, or the cost of the cheapest plan, is what is referred to technically as the Wasserstein distance for the real data distribution P_r and the produced data distribution P_g:

$$W(P_r, P_g) = inf_{\gamma \in \prod(P_r,P_g)} E_{(x,y)\sim y}[||x - y||] \tag{2.5}$$

Where $\prod(P_r, P_g)$ denotes the set of all joint distributions between P_r and P_g, $\prod$ contains all the possible transport plan γ. Using the Kantorovich-Rubinstein duality, we can simplify the calculation to:

$$W(P_r, P_g) = sup_{\|f\|_L \leq 1} E_{x\sim P_r}[f(x)] - E_{x\sim P_g}[f(x)] \tag{2.6}$$

where sup is the least upper bound and f is a 1-Lipschitz function following Lipschitz constraint:

$$|f(x_1) - f(x_2)| \leq |x_1 - x_2| \tag{2.7}$$

WGAN-GP uses gradient penalty to enforce the Lipschitz constraint. A differentiable function f is 1-Lipschitz if and only if it has gradients with norm at most $1(\|\nabla f\|_2 \leq 1)$ everywhere. The model is penalized if the gradient norm moves away from its target norm value 1.

Compared with Basic GAN, the network is without the sigmoid function and outputs a scalar score rather than a probability. This score can be interpreted as how real the input data are. In addition, a gradient penalty is used in the discriminator.

	Discriminator	Generator
GAN	$-\frac{1}{m}\sum_{i=1}^{m}[logD(y^i) + log\left(1 - D\left(G(x^i)\right)\right)]$	$-\frac{1}{m}\sum_{i=1}^{m} log\ (D(G(x^i)))$
WGAN-GP	$\frac{1}{m}\sum_{i=1}^{m}\left[D(y^i) - D\left(G(x^i)\right) + \lambda(\|\nabla D_{y^i \sim x^i}\|_2 - 1)^2\right]$	$-\frac{1}{m}\sum_{i=1}^{m} D(G(x^i))$

Table 2.1: The comparison of Basic GAN and WGAN-GP loss function.

Chapter 3

Dataset and Features

3.1 Dataset Descriptions

The data used in the model includes the dollar index from Fred, stock index and stock price data from Yahoo Finance, and news data scraped from SeekingAlpha. The model's target stock price is the closing price of Apple Inc. stock, and this price is used to calculate statistical statistics. The dataset consists of 36 variables and a total of 2497 observations, with a 7:3 split between the test and train data.

Feature Name	Feature explanation	Feature Name	Feature explanation
Open	Opening price in the trading day	Amazon	Amazon company stock Closing Price
High	Highest price in the trading day	Goolgle	Google company stock Closing Price
Low	Lowest price in the trading day	Microsoft	Microsoftcompany stock Closing Price
Close	Closing price in the trading day	MA7	7-day simple moving average
Volume	Volume in the previous trading day	MA21	21-day simple moving average
NASDAQ	NASDAQ Composite Index Closing Price	20SD	Bollinger bands mid-rail
NYSE	NYSE Composite Index Closing Price	MACD	Moving average convergence/divergence
S&P500	S&P 500 Index Closing Price	upper	Bollinger band upper track
FTSE100	FTSE100 Index Closing Price	Lower	Bollinger Band lower track
Nikki225	Nikki Average Closing Price	EMA	Exponential moving average
BSE SENSEX	BSE Sensitive Index Closing Price	logmomentum	Logarithmic momentum indicator
RUSSELL2000	RUSSELL2000 Index Closing Price	absolute of 3 comp	3-order reconstruction(absolute)
HENGSENG	Hong Kong Hang Seng Index Closing Price	angle of 3 comp	3-order reconstruction(angle)
SSE	SSE Composite Index Closing Price	absolute of 6 comp	6-order reconstruction(absolute)
CrudeOil	Crude Oil Closing Price	angle of 6 comp	6-order reconstruction(angle)
Gold	Gold Closing Price	absolute of 9 comp	9-order reconstruction(absolute)
VIX	CBOE Volatility Index	angle of 9 comp	93-order reconstruction(angle)
USD index	The US dollar index	News	Sentiment value of financial news

TABLE 3.1: Feature Name and Feature explanation

3.2 Feature Engineering

We employed historical data for multiple assets and extracted technical indicators and trend features, including exponential moving average, momentum, Bollinger bands, and MACD. News sentiment analysis was conducted using FinBert to classify news related to Apple.Inc as positive, neutral, or negative. Fourier transformations were also employed to identify long- and short-term trends in Apple's stock. The combination of technical indicators, trend features, and news sentiment analysis were used to predict the closing price of Apple's stock using a GRU network.

3.3 Data Structure

The process of preparing a dataset for supervised learning by dividing it into equal portions using the rolling window method, as shown in figure 1. To account for timesteps, the original 2-dimensional dataset is transformed into 3 dimensions. Figure 2 illustrates the output generated, which has one unit and can be modified by changing the timestep and output step in the model. The paper being referred to uses a many-to-many model with a timestep of 30 and output step of 3 to predict 3 days of stock price using 30 days of historical price data.

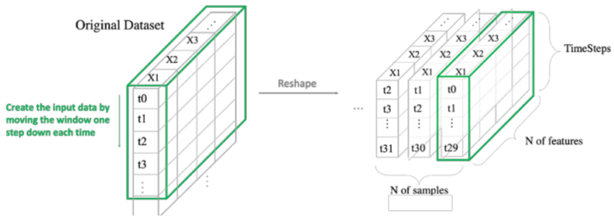

Fig. 1: Divison of datasets

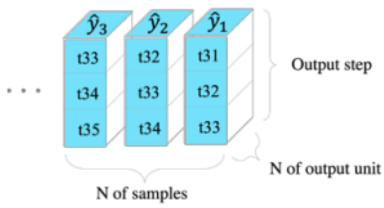

Fig. 2: Generator Output

Chapter 4

Methodology

4.1 The Generator

We chose the GRU as the generator in our GAN model based on its stability. Our dataset covers the stock price history for the previous 10 years as well as 36 variables, including Open, High, and Low, Close, Volume; NASDAQ, NYSE; SP 500; FTSE100; NIKKI225; BSE SENSEX; HENG SENG; SSE; Crude Oil; Gold; VIX; USD index; upper_band; lower_band; EMA; log momentum; absolute of 3 comp; angle of 3 comp; absolute of 6 comp; angle of 6 comp; absolute of 9 comp; angle of 9 comp; and News. This project will perform multi-step ahead prediction, so we must define the input step and the output step in the generator. The generator will receive three-dimensional data as input, consisting of batch size, input-step, and features, and will produce batch size and output step as output. We employ three layers of GRU, with neuron counts of 1024, 512, and 256, followed by two layers of Dense, with the neuron count of the last layer matching the output step we intend to predict, in order to construct a generator with good performance.

4.2 The Discriminator

Convolution neural networks were used as the discriminator in our GAN model with the goal of determining if the input data was authentic or fraudulent. The original data or newly created data from the generator will serve as the discriminator's input. This discriminator model has three 1D Convolution layers with 32, 64, and 128 neurons each, plus three additional Dense layers with 220, 220, and 1 neuron at the very end. All layers' activation functions have been set to the Leaky Rectified Linear Unit (ReLU), except for the output layer, which has the Sigmoid activation function for GAN and the linear activation function for WGAN-GP. The Sigmoid function will give a single scalar output, 0 and 1, which means real or fake.

4.3 The Architecture of GAN

We integrated these two models as our suggested GAN model using the two aforementioned structures for our generator and discriminator.

We already created the function in the Theoretical background section, and we utilise cross-entropy to calculate our loss for both the generator and discriminator in our GAN model structure. For the discriminator in particular, we coupled the generated stock price with the historical stock price of the input steps as our input. This step lengthens the data and improves the discriminator's accuracy as it learns the categorization.

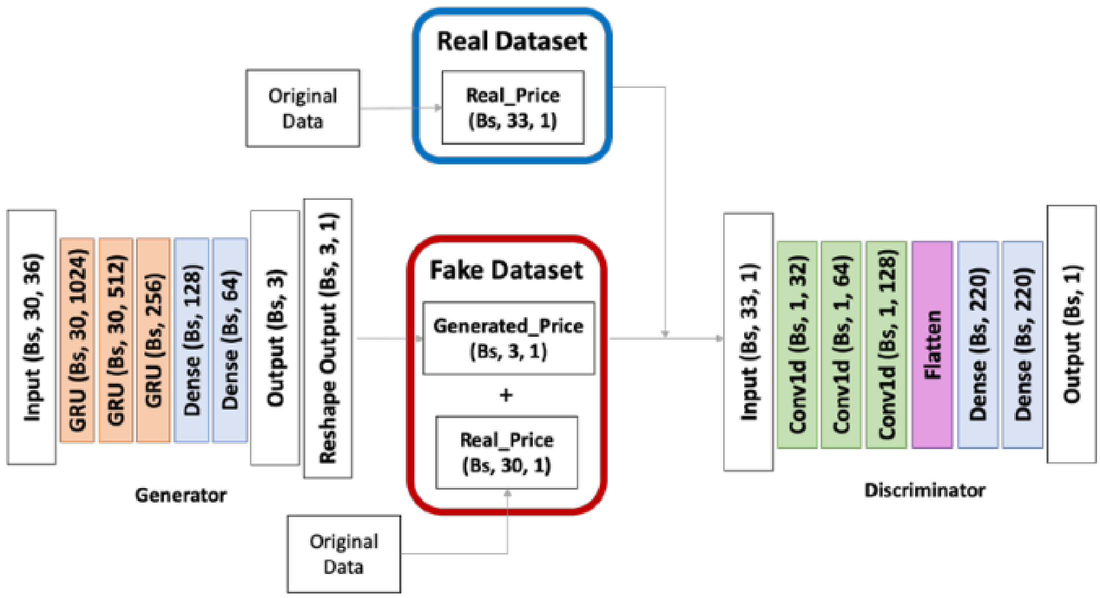

Fig. 3: Architecture of GAN

Chapter 5

Experimental Results

5.1 Training of our model

We predicted the stock closing price for the next three days using data from the last 30 days, along with 36 other features that may affect the price. The dataset will be split into a 70% training set and a 30% testing set. The testing set will be used to perform predictions with and without an unexpected event, which in this case is COVID-19 for the year 2020.

5.2 Experimental and results

Root Mean Square Error (RMSE), an indicator whose definition is as follows, was used to assess each model's performance.

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

We examined each model's RMSE using testing data (with and without 2020) in order to assess the models we constructed for this project. The N denotes the number of data points, x_i is the actual stock price, and $\hat{x}_l$ is the forecasted stock price.

5.2.1 LSTM

The LSTM model used Bidirectional LSTM in the first layer and the Adam algorithm with a learning rate of 0.001 as the optimizer. The batch size was 64, and the model was trained for 50 epochs on the stock price dataset. The GAN model used the entire dataset, which included 10 years of historical data and 36 correlated features. The testing dataset started on 07/21/2017, after the data was split into a training set and a testing set.

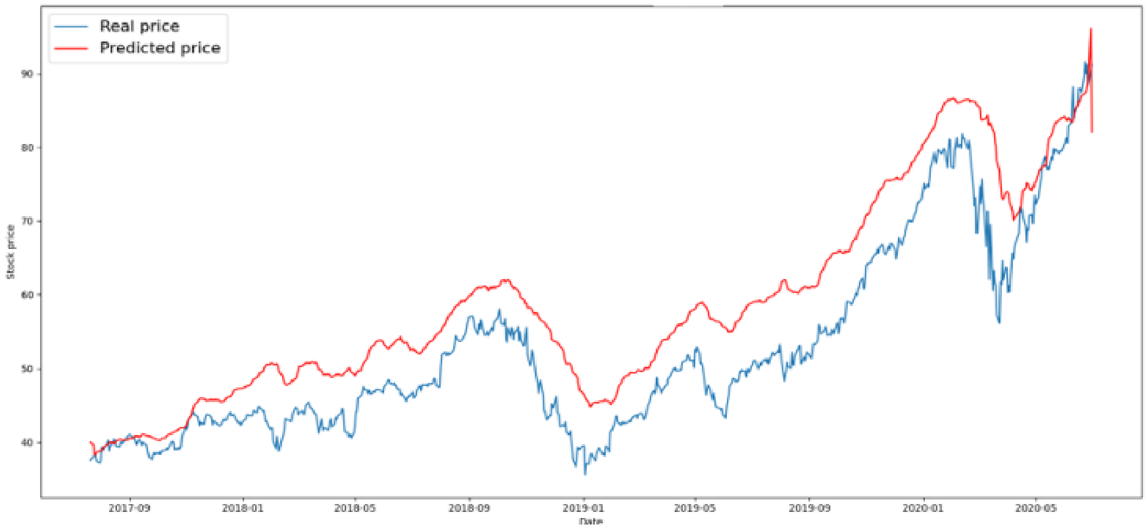

FIG. 4: LSTM including the forecasting of 2020

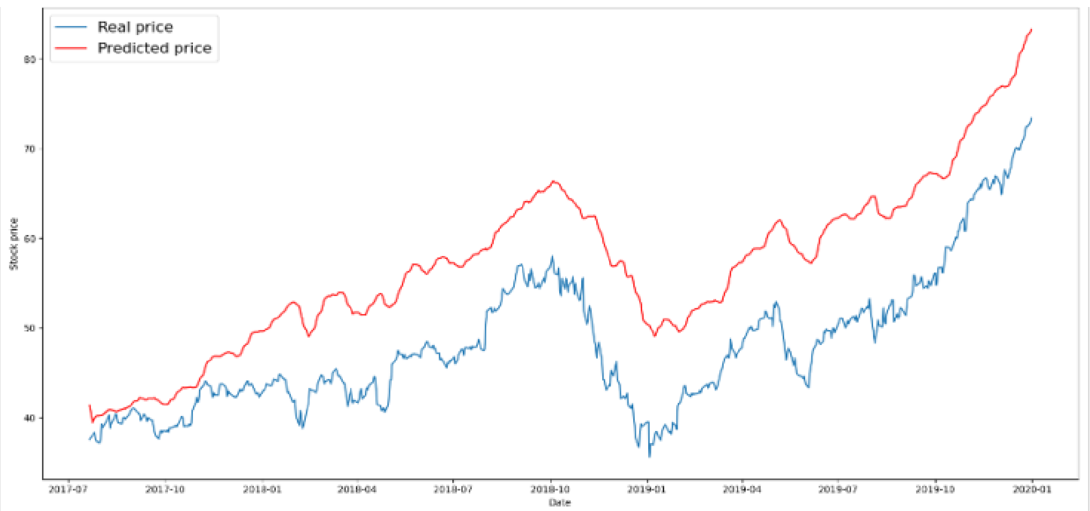

FIG. 5: LSTM excluding the forecasting of 2020

From Fig 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. Fig 5 is the result excluding 2020, then the RMSE is increased to 9.42, which is much higher than the result that includes 2020.

5.2.2 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.

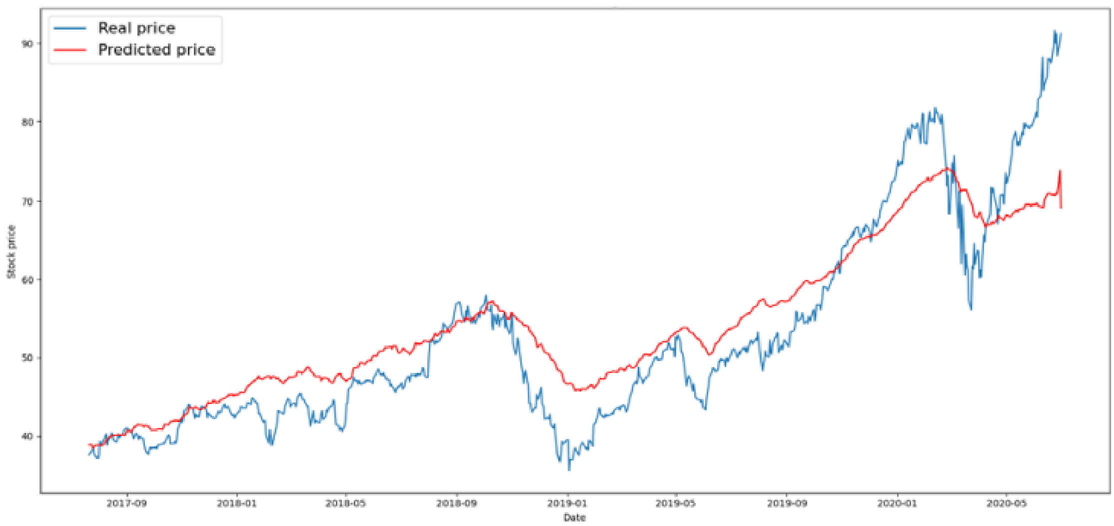

Fig. 6: GRU including the forecasting of 2020

Fig 6 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 7 is the result excluding 2020 for GRU, the RMSE is 4.08. The GRU model performs better when making predictions without predicting unexpected events.

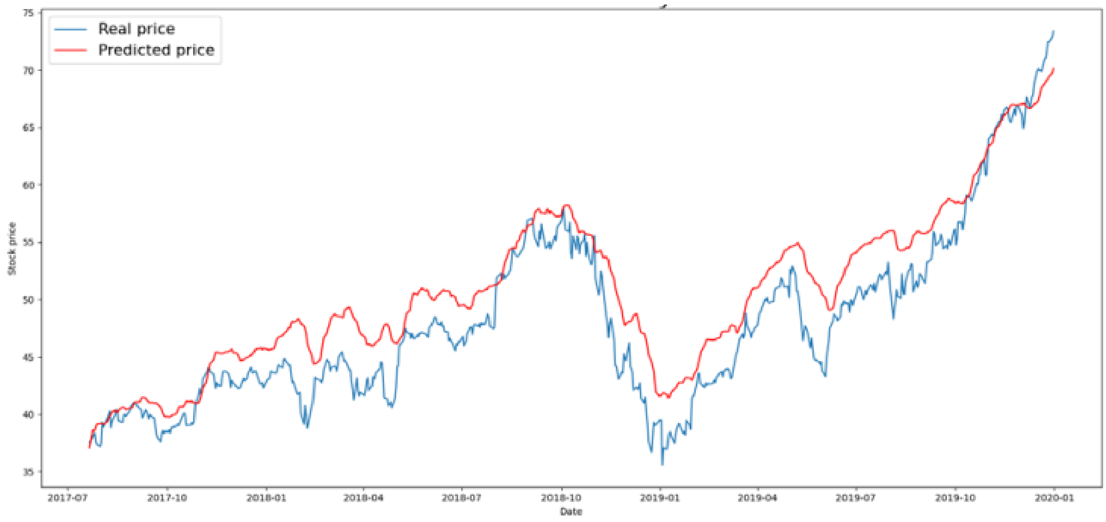

Fig. 7: GRU excluding the forecasting of 2020

5.2.3 Basic GAN

We propose a GAN model with an Adam optimizer, a learning rate of 0.00016, and a batch size of 128 trained on a dataset for 165 epochs. The loss plot of the model shows the discriminator's loss is initially higher than the generator's loss, but both become flat over time. The predicted results of the basic GAN model show an RMSE of 5.36, with a sudden surge in 2020 potentially due to COVID-19. Excluding 2020 from the forecast results in an RMSE of 3.09, indicating better performance without unexpected events.

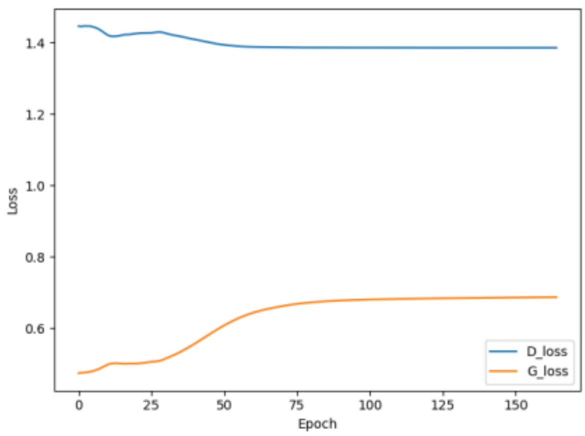

Fig. 8: loss plot of the basic GAN

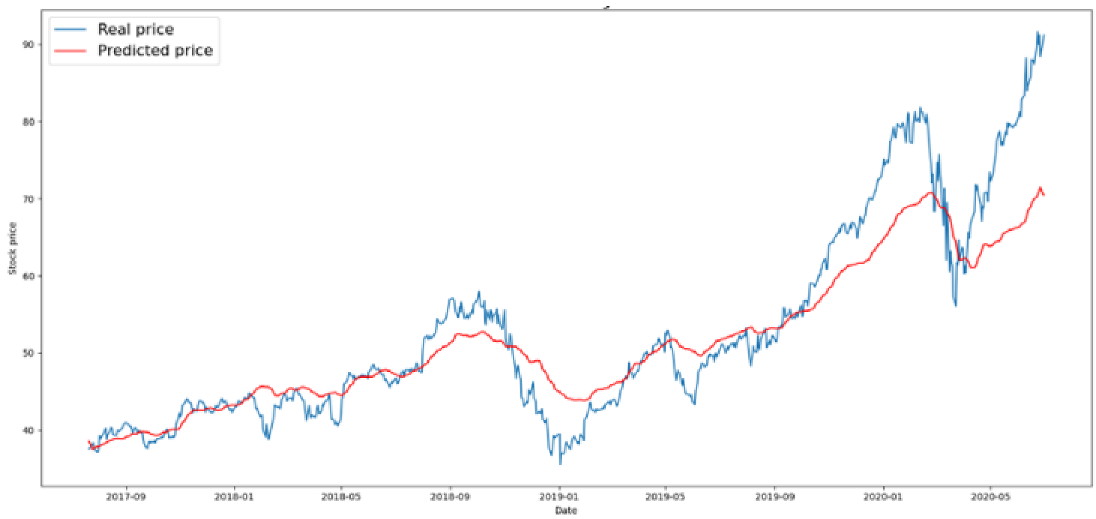

FIG. 9: GAN including the forecasting of 2020

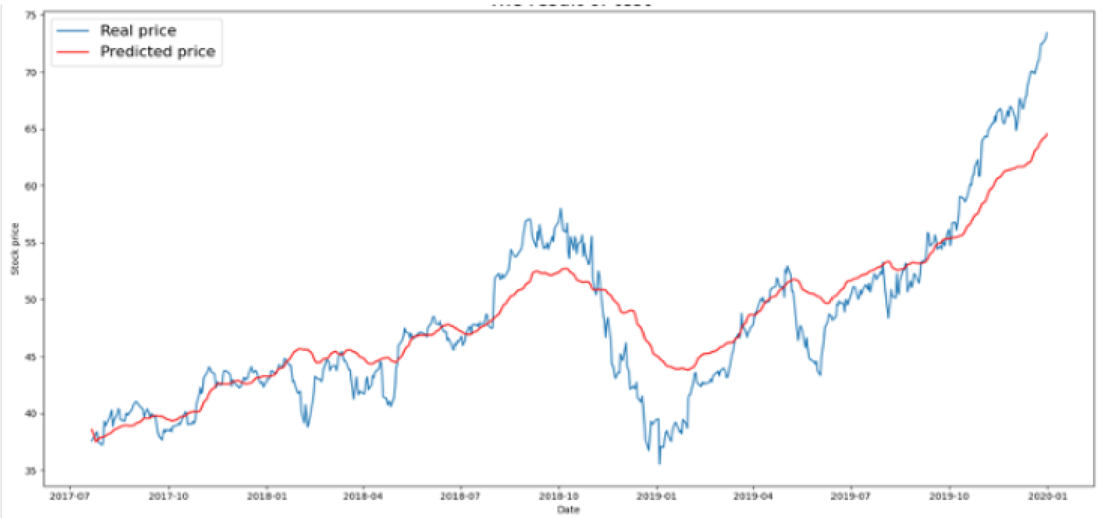

FIG. 10: GAN excluding the forecasting of 2020

5.2.4 WGAN-GP

We propose a WGAN-GP model with an Adam optimizer, a learning rate of 0.0001, and a batch size of 128 trained on a dataset for 100 epochs. The discriminator is trained once, and the generator is trained three times. The WGAN-GP model shows better performance than the Basic GAN model, with an RMSE of 4.77 in Fig 12. However, there is still a large gap between the real and predicted prices in 2020 due to COVID-19. Excluding the test data in 2020 in Fig 13 results in a decrease in RMSE to 3.88 but worse performance than the Basic GAN model.

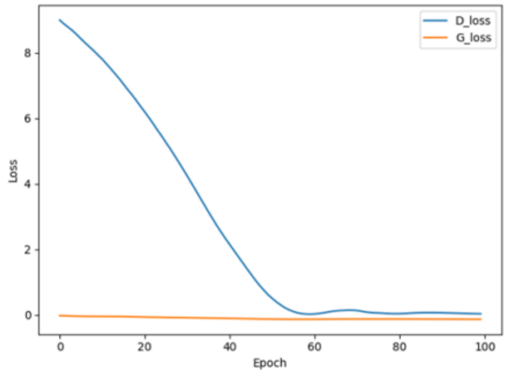

FIG. 11: loss plot of the basic WGAN-GP

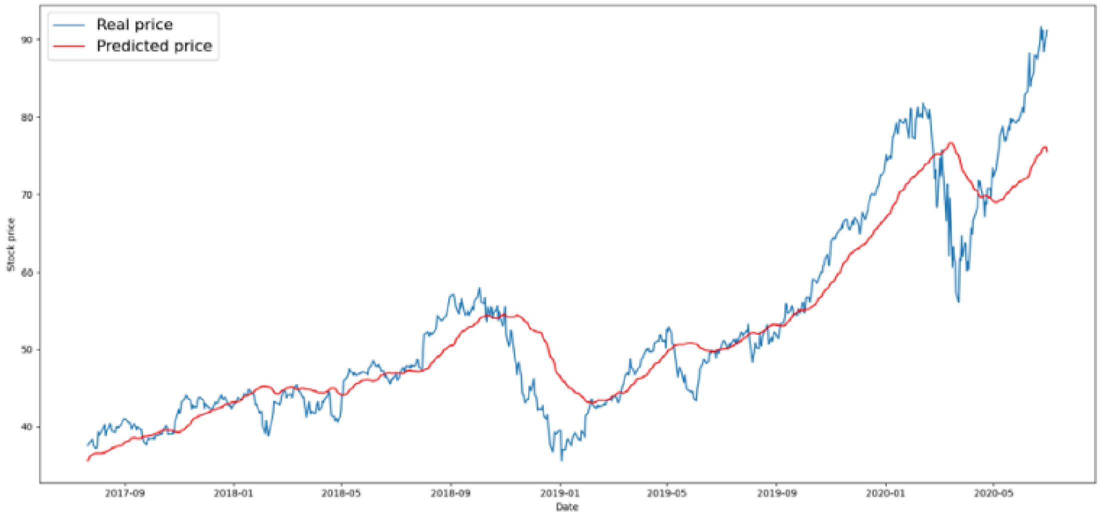

FIG. 12: WGAN-GP including the forecasting of 2020

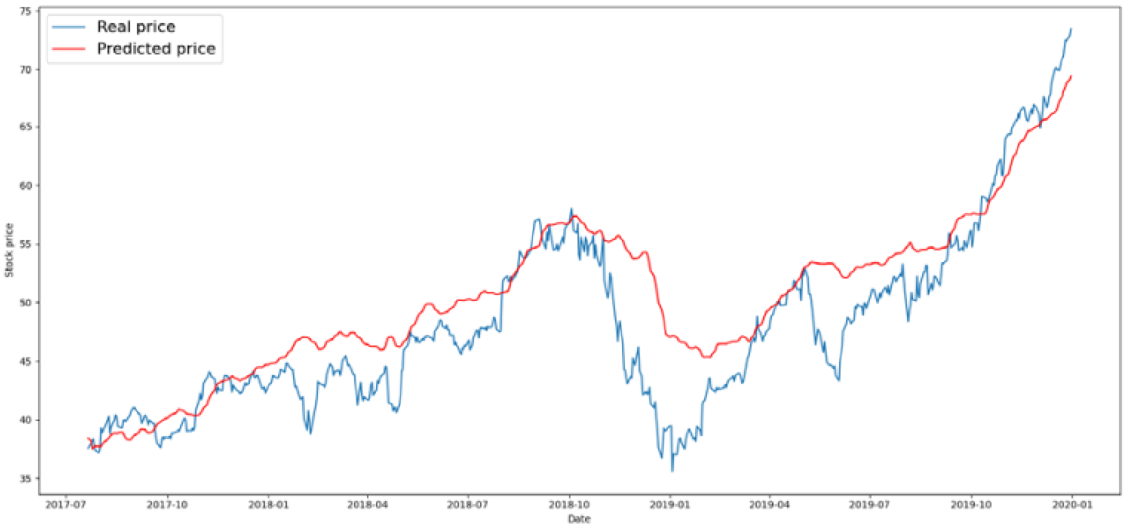

FIG. 13: WGAN-GP excluding the forecasting of 2020

5.3 Evaluation

According to the experimental results, the GRU model performs the best on the training dataset, while the WGAN-GP model performs better on the testing dataset when COVID-19 period data is included. On the other hand, when the COVID-19 period data is excluded from the testing dataset, the basic GAN model performs the best.

Despite these variations in performance across different datasets and models, the authors conclude that overall, GAN models outperform baseline traditional models based on their results. This suggests that GANs can be a promising approach for time-series data generation, especially in scenarios where unexpected events or anomalies may impact the data.

However, it is important to note that further research is needed to validate these findings and explore the potential limitations and challenges of applying GANs to time-series data generation in various real-world applications.

	LSTM	GRU	Basic GAN	WGAN-GP
RMSE of Training dataset	1.52	1.00	1.64	1.74
RMSE of Testing dataset (include 2020)	6.60	5.33	5.36	4.77
RMSE of Testing dataset (exclude 2020)	9.45	4.08	3.09	3.88

Table 5.1: RSME for different Models

Chapter 6

Conclusion and Future Work

We proposed a GAN which sets GRU as a generator and CNN as a discriminator. According to the experimental result, we have some conclusions. First, compared the GAN model with the traditional models, the GAN model can help to improve the GRU model and LSTM model, both basic GAN and WGAN-GP perform better than traditional models. One of the key findings of this work is that, when there is an unexpected event like COVID-19, WGAN-GP performs better than basic GAN, but in normal periods, basic GAN performs better. However, to our knowledge, a GAN model including RNN is unstable, it is very difficult for these models to tune hyperparameters, without good parameters you may have bad results.

Future research should be devoted to the development of hyperparameter tuning. In the GAN model, if each of the parameters, in each layer and for the whole model, can be tuned more accurately, we believe the result would have significantly improved. There are many research teams proposed methods of reinforcement learning for hyperparameter optimization, such as Rainbow based on Q-learning and Proximal Policy Optimization (PPO). Based on the basic structure in this paper and exploring further reinforce learning, we hope the GAN model with RNN can produce much more reliable forecasting of the stock price.

Chapter 7

References

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

Zhang K., Zhong G., Dong J., Wang S., Wang Y. (2019). *Stock market prediction based on generative adversarial network.*

Cho Kyunghyun, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. *Learning phrase representations using RNN encoderdecoder for statistical machine translation.*

Zhou X., Pan Z., Hu G., Tang S., Zhao C. (2018). *Stock Market Prediction on High-Frequency Data Using Generative Adversarial Nets*

Banushev, B., (2020). *Using the latest advancements in AI to predict stock market movements.*

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.*

Schulman, John, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. *Proximal policy optimization algorithms.*

Jonathan, H., (2020). *GAN — Wasserstein GAN WGAN-GP.*