6501 Capstone Final Report – Group 1
Stock price prediction using GAN

Chen Chen, Hung-Chun Lin

Abstract
Deep learning is an exciting topic, it has been utilized in many areas owing to its strong potential. For example, it has been widely used in the financial area which is vital to the society, such as high-frequency trading, fraud detection and risk management. Stock market prediction is one of the most popular and valuable areas in finance. In this paper, we propose 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 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.

Introduction
Stock price prediction is an interesting and challenging topic. Many studies have shown that the stock price is predictable. Stock price prediction is a kind of time series prediction, many classic algorithms such as Long Short Term Memory (LSTM), ARIMA are used in time series predictions. GAN is one of the most powerful models, the generator and discriminator in the model are adversarial, which help to generate more accurate output. GAN is widely used in image generating, but not in time series prediction. Since there are few studies on time series prediction using GANs, and according to the result of their study, their conclusion is inconsistent. So, in this paper, we decide to use GANs to predict the stock price and to check whether the adversarial system can help improve the time series prediction. We will compare the traditional models LSTM, GRU with the basic GAN and Wasserstein GAN with gradient penalty (WGAN-GP) model.
Accurately predicting stock price complexes, stock price movement is influenced by many factors. So we need to incorporate as much information as possible. To capture more information, we integrate multiple traditional technical indicators such as MACE, MAE, we combine a characteristic mix of underlying assets, including commodities, currencies, indices, VIX, and more. Also, we use Fourier transform techniques to extract the overall trend of price changes.In addition, we use the latest Natural language processing (NLP) model FinBert to conduct sentiment analysis of financial-related news in the market. We try to predict the price movements of Apple.Inc. For the purpose, we will use daily closing prices from July 1st, 2010 to June 31st, 2020 (seven years for training purposes and two years for validation purposes).

Problem Statement
In this project, we will compare two algorithms applying to stock prediction. First, we will utilize the LSTM networks and GRU to do the prediction, both are known for being capable of learning order dependence in sequence prediction problems. Furthermore, we will utilize the GANs to improve the prediction, unlike the traditional GANs, in this project, the GAN model will be

implemented with one of the recurrent neural networks as a generator and a Convolution Neural Network as a discriminator.

The main goal of this paper is to improve the accuracy of basic LSTM and GRU models by utilizing the GAN model, which contains one discriminator to assist in training the generator.
The contribution in this paper can be summarized by the following:

- For the basic model, we compared two kinds of RNN based model, the GRU and the Long LSTM networks
- In this project, we proposed the GANs model, with GRU as a generator and CNN as a discriminator for predicting the multi-step ahead stock price.
- We try to improve the result by adjusting the loss function in the GAN model, which is the WGAN-GP
- In this project, we extract the daily news topic through Natural Language Process as one of the vital indexes in the features
- Compare model performance for normal times and COVID-19 period

Related Works

Stock prediction is widely used in traditional models such as LSTM, GRU and ARIMA. But there are few studies that make the prediction using GAN. And the result of using GAN to make the stock prediction is inconsistent. For example, Ricardo, A.,Carrillo, R.[1] compared the performance of the GAN model with traditional deep learning model LSTM, they used LSTM as the generator and CNN as the discriminator. Their specific goal was to predict whether the price would increase one day after our sample period, and the result showed there are no significant differences with GAN and traditional model LSTM, accuracy on GAN model is 72.68% compared with accuracy on shallow LSTM 74.16%, the performance of GAN is even a little bit worse. However, according to the study from K. Zhang,etc[2], they proposed a GAN model with the LSTM as the generator and Multi-Layer Perceptron(MLP) as the discriminator to forecasting the 1 day closing price of the stock, and also compared the result with baseline LSTM. The result showed that GAN performs better than their baseline traditional model, the accuracy for GAN model is about 75.54% while for baseline LSTM is 68.59%.

Theoretical background

1.1 LSTM

Long short-term memory (LSTM) is a specific recurrent neural network (RNN) architecture, it was proposed in 1997 by Hochreiter and Schmidhuber [3], that is unlike a traditional feed-forward neural network but includes the feedback connections. Furthermore, it can not only be utilized on single-point data but on the sequence of data as well. The basic components of LSTM are an input gate, an output gate and a forget gate, and the LSTM network was developed to resolve the vanishing gradient problem while training the traditional RNNs. LSTM is a cell memory unit which means that LSTM has the ability to remove or add information to the cell state.
LSTM has overcome the vanishing gradients and the exploding gradients problem that appeared in RNN through the specific internal structure of the units built in the model. Nowadays, LSTM has been known as a powerful method that is capable of processing, classifying, and making predictions based on time series data.

1.1 GRU

Gated recurrent units (GRUs) is a kind of RNN that using gating mechanisms to control the flow of information between cells in the neural network, which was derived from LSTM and is introduced in 2014 By Kyunghyun Cho et al [4]. GRU is composed by two gates, update gate and reset gate, these gates are used for filtering out what information should remain and what should be disposed of. Different from traditional RNN, GRUs solve the vanishing and exploding gradient problems. Unlike LSTM, GRU has fewer parameters than LSTM due to the lack of one gate. The other main difference is that GRUs also have a lack of the cell state from LSTM so that GRU can only store both long and short-term memory in the hidden state. Recently, GRUs have been shown to perform better than LSTM on certain smaller and less frequent datasets.

1.2 GAN

Generative adversarial networks (GANs) is a minimax problem, which is based on zero-sum non-cooperative games. In general, GAN is composed of two components, a generator and a discriminator. The generator is aimed to generate examples that can look as real as possible, and the goal for the discriminator is to distinguish the examples as real or fake (generated).

GAN is a technique that has been rapidly explored in the deep learning field. Based on the basic structure, people are developing different methods for improving the result by adjusting the structure and loss function. Then nowadays, there are various types of GANs have been proposed, such as Conditional GANs (cGANs) which using extra label information to improve the generator, Wasserstein GANs (WGANs) that includes Wasserstein distance to the loss function, WGAN with Gradient Penalty which add the regularization to their loss function, Cycle GAN, PGGAN and SAGAN which change the structure.

1.3 Basic GAN

In the original GAN, the loss function is based on KL-JS divergence, in the training process, the GAN model will use cross-entropy loss to minimize the difference between two distributions which is equivalent to minimizing the KL-JS divergence.

In this project, we train discriminator to maximize its objective function, the probability of assigning the correct label to the samples, the objective function for discriminator to maximizing is defined as:

$$\hat{V} = \frac{1}{m}\sum_{i=1}^{m} logD\left(y^i\right) + \sum_{i=1}^{m}(1 - log\ D(G\left(x^i\right))) \tag{1}$$

and then we train generator to minimize its objective function which is:

$$\hat{V} = \frac{1}{m}\sum_{i=1}^{m}(1 - log\ D(G\left(x^i\right))) \tag{2}$$

Where x is the input data for generator, y is the target from the real dataset, $G\left(x^i\right)$ 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:

$$\hat{V} = -\frac{1}{m}\sum_{i=1}^{m} log\ D\left(y^i\right) - \frac{1}{m}\sum_{i=1}^{m}(1 - log\ D(G\left(x^i\right))) \tag{3}$$

The loss function of generator is:

$$\hat{V} = -\frac{1}{m}\sum_{i=1}^{m} \quad (log\ D(G\left(x^i\right))) \tag{4}$$

Through the training process, it always needs to minimize the loss function to get the better result.

1.4 WGAN-GP

The discriminator in Basic GAN is not powerful enough. The training process is known to be slow and unstable. WGAN-GP is proposed to help stabilize and improve the training of GAN.
WGAN-GP proposed the Wasserstein distance to solve this problem. The Wasserstein distance (or Earth-Mover Distance (EMD)) is the minimum cost of transporting mass in converting the data distribution to the data distribution. The Wasserstein distance for the real data distribution *Pr* and the generated data distribution *Pg* is mathematically defined as the greatest lower bound (infimum) for any transport plan (i.e. the cost for the cheapest plan) [5]:

$$W(P_r, P_g) = inf_{\gamma \in \Pi(P_r, P_g)} E_{(x,y) \sim \gamma}[\|x - y\|] \quad (5)$$

Where Π(Pr, Pg) denotes the set of all joint distributions between Pr and Pg, Π contains all the possible transport plan γ. Using the Kantorovich-Rubinstein duality, we can simplify the calculation to:

$$W(P_r, P_g) = sup_{\|f\|_L \leq 1} E_{x \sim P_r}[f(x)] - E_{x \sim P_g}[f(x)] \quad (6)$$

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

$$|f(x_1) - f(x_2)| \leq |x_1 - x_2| \quad (7)$$

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. [5]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. [5] In addition, a gradient penalty is used in the discriminator.

Table 1. The comparison of Basic GAN and WGAN-GP loss function

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

Dataset and Features

1.1 Dataset Descriptions

The stock price data and stock index data are from Yahoo Finance, the dollar index is from Fred, and the news data are scrapped from SeekingAlpha. The target stock price in the model is Apple.Inc. stock close price. The statistical data are calculated using the stock closing price. There are a total 2497 observations and 36 variables in the dataset. The train data and test data are split into 7:3.

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

1.2 Feature Engineering

After downloading various asset historical data, we calculated some technical indicators and extracted some trend features. In addition, we use the NLP method to sentiment values of relevant news.

Technical indicators: we calculated most popular technical indicators for investors (7 and 21 days moving average, exponential moving average, momentum, Bollinger bands, MACD.)

News sentiment analysis: news can indicate potential stock price movements. We scrapped all the daily news of Apple.Inc, and used FinBert to analyze the news into positive, neutral or negative by giving a score between -1 to 1.

Fourier transforms: along with the daily closing price, we created Fourier transforms to extract long term and short term trends in the Apple stock. Fourier transforms take a function and create a series of sine waves, when combined, these sine waves approximate the original function, which can help the GRU network pick its prediction trends more accurately.[6]

1.3 Data Structure

The method we prepared our dataset for supervised learning is to divide the dataset with the rolling window equals 1, the illustration has been shown in figure 1. The original dataset is 2 dimensional, we need to reshape the data to 3 dimensions according to the timesteps.

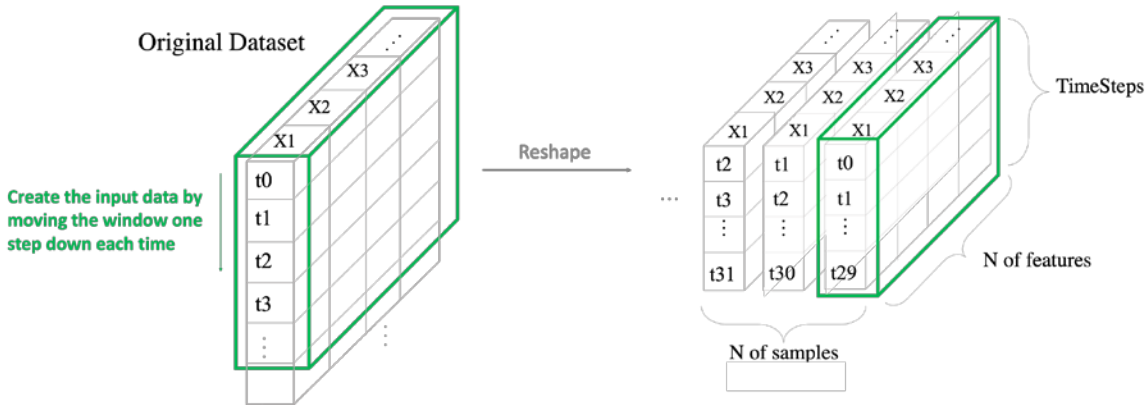

Figure (1)

Figure 2 is the illustration of the output we will get from the generator. Here, the number of output units equals 1. In our model, we can modify the timestep in figure 1 and output step in figure 2, in this paper, we built a many to many models with timestep 30 and output step 3(use 30 days historical price to predict 3 days stock price).

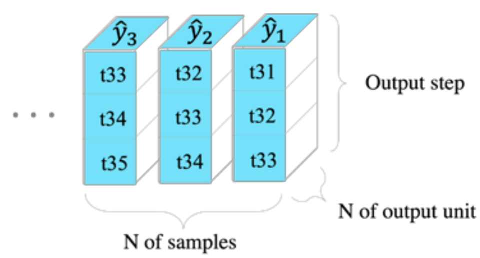

Figure (2)

Methodology

1.1 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 and also includes 36 features, includes Open, High, Low, Close, Volume, NASDAQ, NYSE, S&P 500, FTSE100, NIKKI225, BSE SENSEX, RUSSELL2000, HENG SENG, SSE, Crude Oil, Gold, VIX, USD index, Amazon, Google, Microsoft, MA7, MA21, MACD, 20SD, 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 do the multi-step ahead prediction, 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.

1.2 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.

1.3 The Architecture of GAN

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

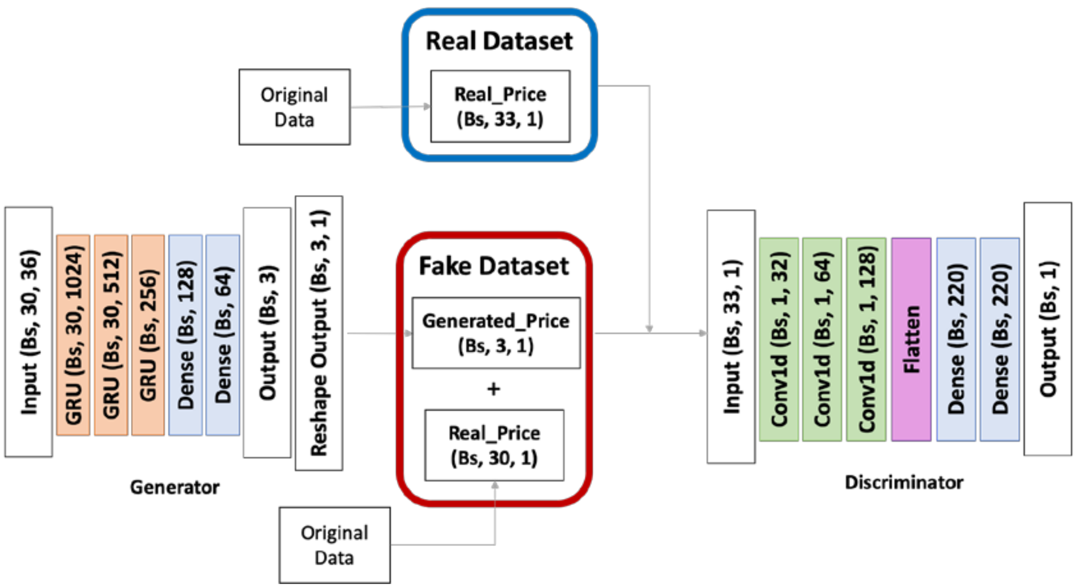

Figure (3)

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.

Experimental Results

1.1 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 36 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.

1.2 Experimental and results

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

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}(x_i - \widehat{x_i})^2}{N}} \quad (8)$$

The N is the number of the data points, x_i is actual stock price, and $\widehat{x_i}$ 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).

1.2.1 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, the testing dataset started on 07/21/2017.

Figure (4)

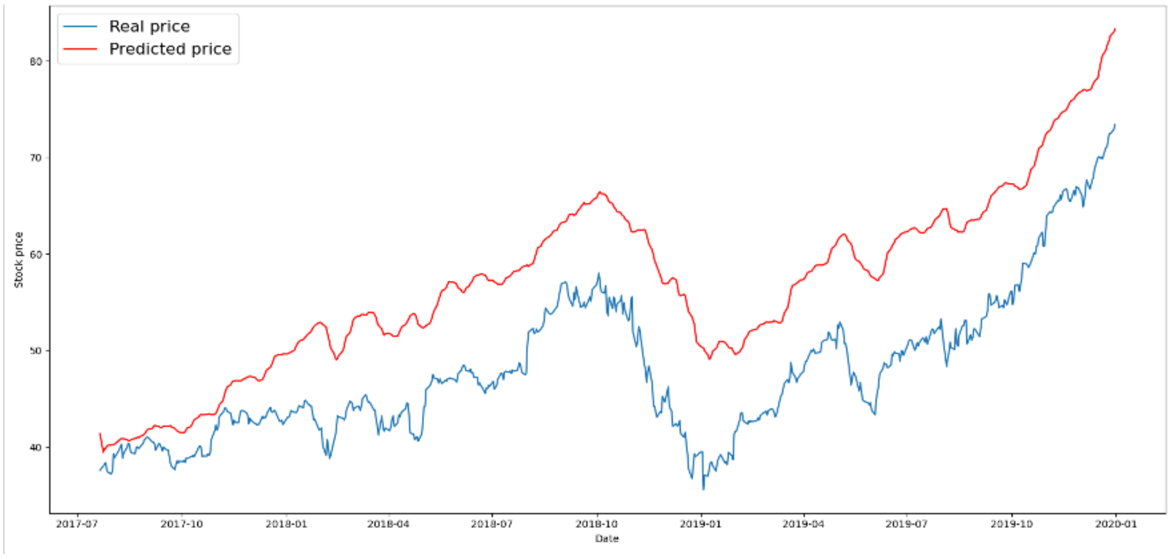

Figure (5)

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. And 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.

1.2.2 GRU

GUR 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.

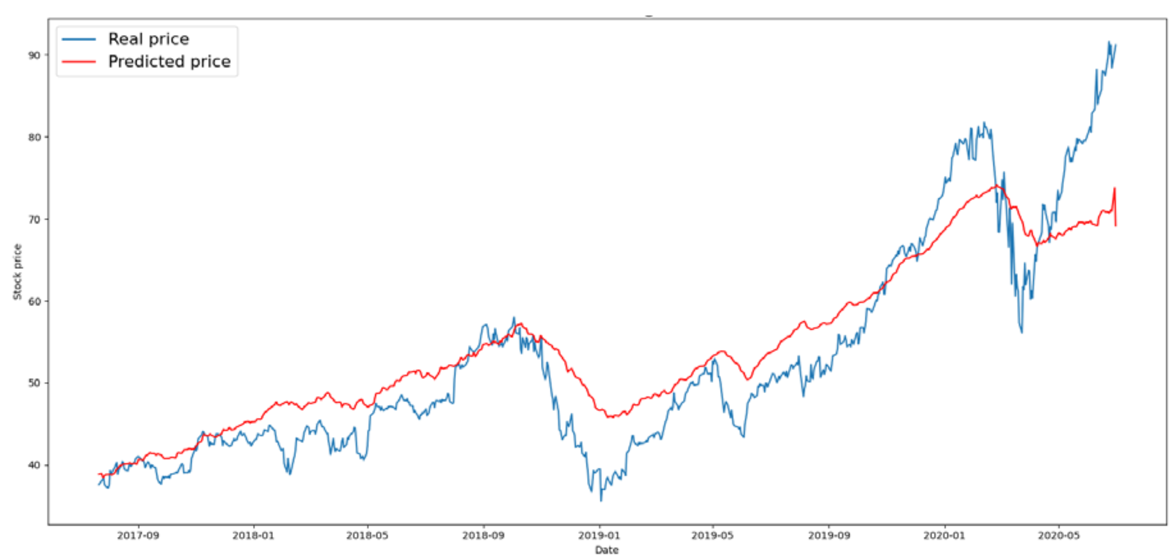

Figure (6)

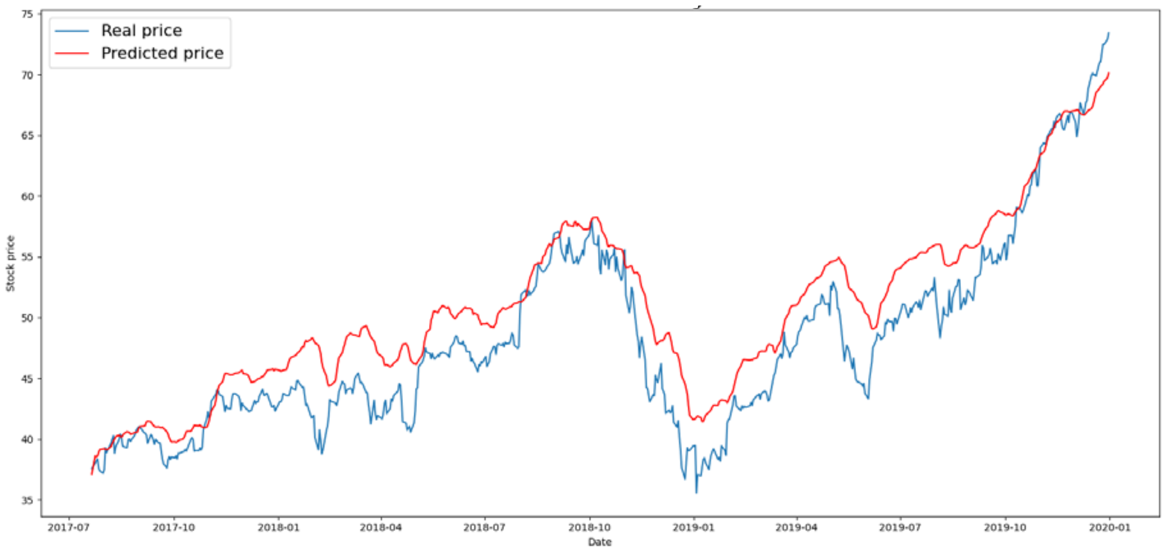

Figure (7)

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.

1.2.3 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 epochs.

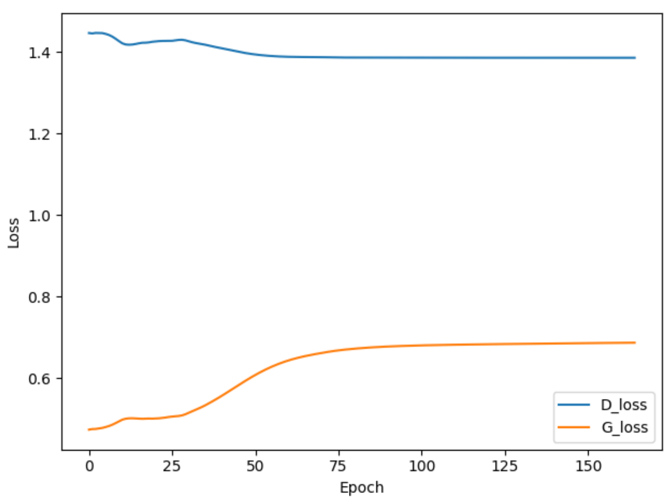

Figure (8)

Fig 8 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 flatten.

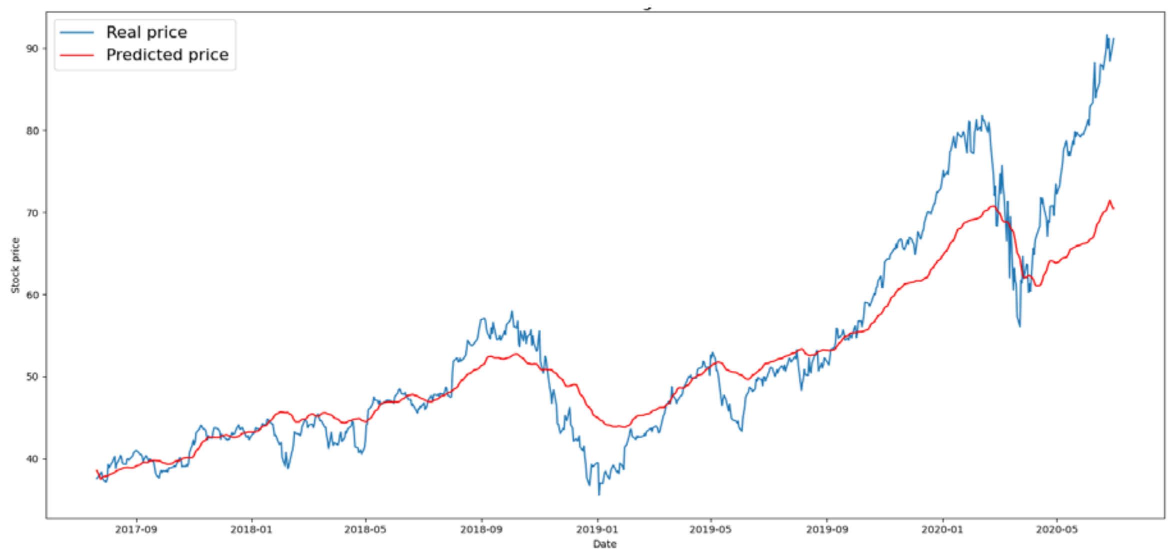

Figure (9)

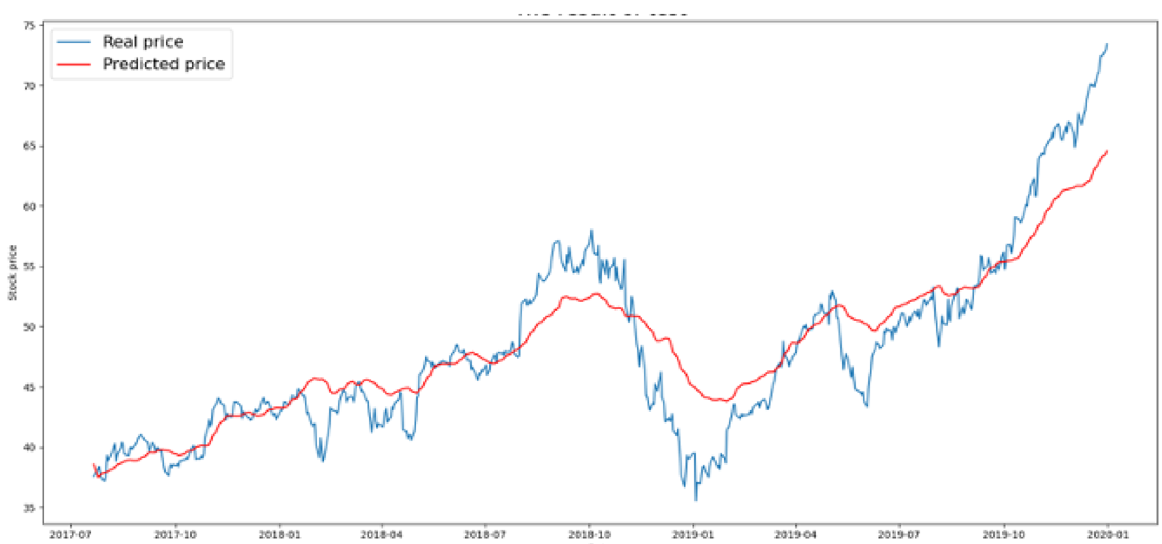

Figure (10)

Fig 9 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 due to the unexpected event, COVID-19. Fig 10 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.

1.2.4 WGAN-GP

The structure of the WGAN-GP model in this paper has been proposed in the methodology section. In this model, the optimizer is also Adam algorithm with a learning rate 0.0001. The size of batch is 128 and we train the model on this dataset for 100 epochs. And we train the discriminator once, train the generator 3 times.

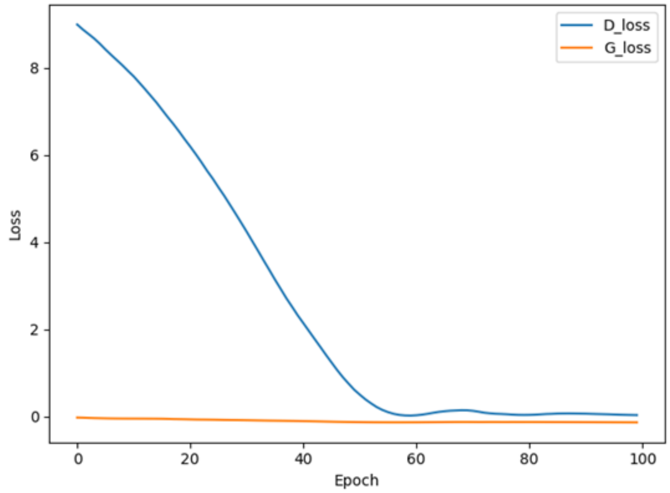

Figure (11)

Fig 10 is the loss plot of the WGAN-GP model, the blue line is the loss path of the discriminator and the orange line is the loss path of the generator. The discriminator loss decreases towards 0. Compared with the Basic GAN, the discriminator in WGAN-GP learns better.

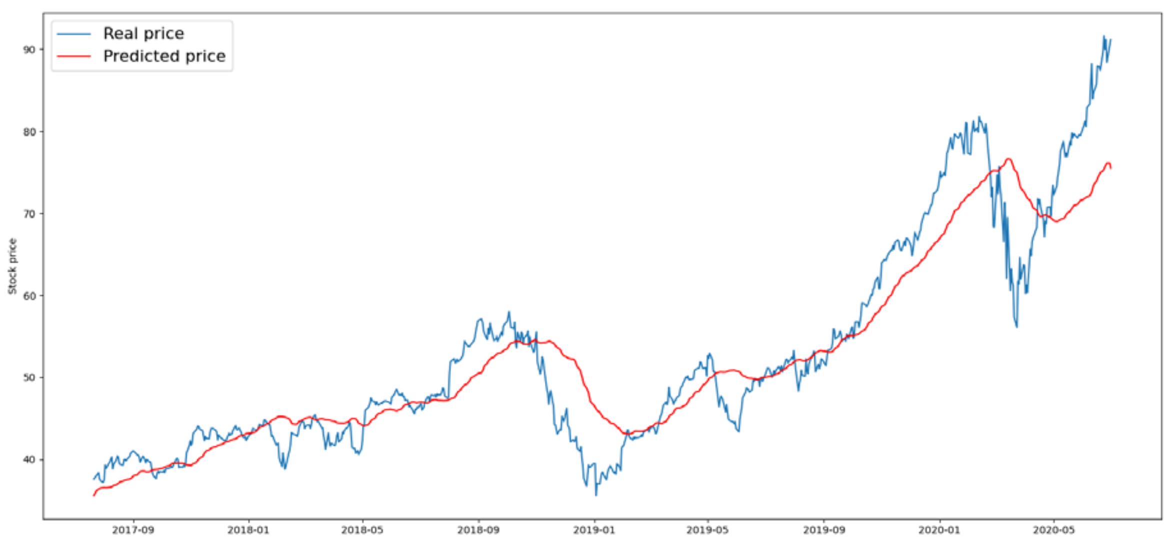

Figure (12)

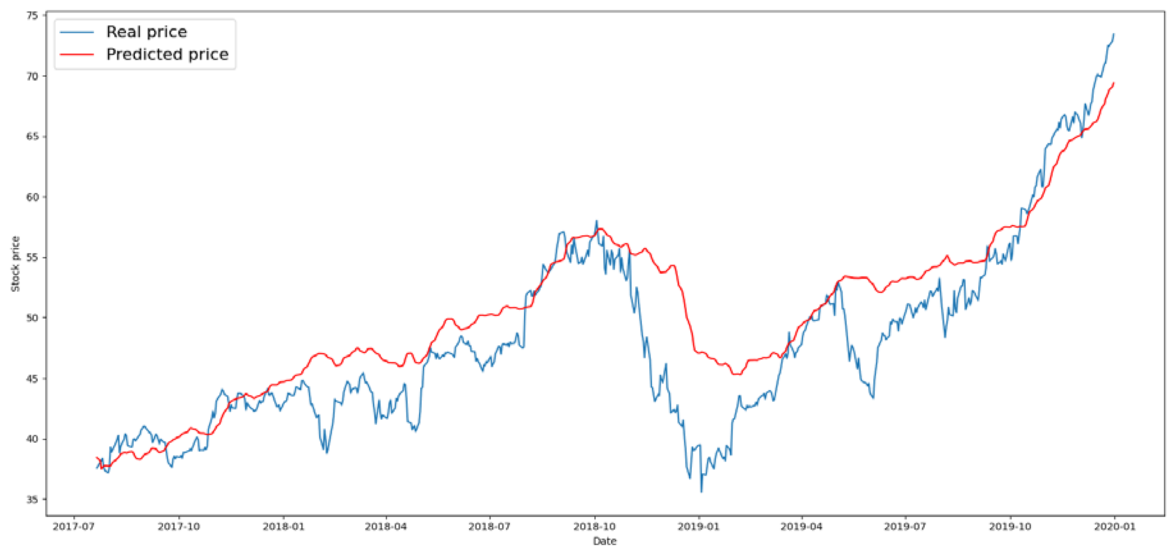

Figure (13)

Fig 12 is the predicted result of the WGAN-GP model, and the RMSE is 4.77, from this figure, we can see the prediction is similar with basic GAN starts to have a large gap between the real price and the predicted price in 2020 and the sudden surge might due to the unexpected event, COVID-19. Fig 13 is the result of the excluding 2020 forecasting, and the RMSE decreases to 3.88.

Evaluation

The table compares the training RSME and testing RMSE for different models.

Table 2. The evaluation on different models

	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

For the training dataset, GRU performs the best, but for the testing data, when we include COVID-19 period data, WGAN-GP performs the best, but when we exclude that period, Basic GAN performs the best. But overall, GANs models perform better than the baseline traditional models according to our result.

Conclusion

In this paper, 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 [7] based on Q-learning and Proximal Policy Optimization (PPO) [8]. 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.

Reference

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

[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-6a1a2aa1b490>