Time Series Forecasting of Stock Prices using Neural Networks LSTM and GAN

Vijay Mane
Electronics and Telecommunication Engineering.
Vishwakarma Institute of Technology, Pune,India
vijay.mane@vit.edu

Ruturaj Patil
Electronics and Telecommunication Engineering.
Vishwakarma Institute of Technology, Pune,India
ruturaj.patil21@vit.edu

Shantanu Pawar
Electronics and Telecommunication Engineering.
Vishwakarma Institute of Technology, Pune,India
shantanu.pawar21@vit.edu

Nagesh Pujari
Electronics and Telecommunication Engineering
Vishwakarma Institute Of Technology, Pune,India
nagesh.pujari21@vit.edu

***Abstract* — This project focuses on forecasting Amazon stock prices using advanced machine learning techniques, specifically Long Short-Term Memory (LSTM) networks and Generative Adversarial Networks (GANs). The dataset comprises historical stock prices and includes parameters such as Low, High, Close, Open and trading Volume. LSTM is utilized to capture temporal dependencies within time-series data, providing robust forecasts, while GANs generate synthetic data to augment training, improving model accuracy and resilience. The system is deployed within a user-friendly environment on Colab, allowing for easy data input, model training, and real-time forecasting. Evaluation metrics, including RMSE and MAE, validate the predictive accuracy of the model, showing strong performance in anticipating stock price trends. By delivering near real-time predictions and visual analytics, this solution aids investors and analysts in making informed decisions, fostering proactive investment strategies. The project also highlights potential applications for similar forecasting in other financial and non-financial time-series data, offering scalability and adaptability for broader use cases. Through continuous refinement and model updates, this approach enhances forecasting accuracy and presents a valuable tool for long-term financial analysis and strategy.**

Keywords— Time Series Forecasting, LSTM, GAN, Stock Price Prediction, Machine Learning, Financial Analysis.

I. Introduction

In today's accurate stock price forecasting, financial markets are essential for traders., and financial analysts, investors seeking to make well-informed choices. Stock prices are inherently volatile and influenced by numerous factors, making prediction a complex yet valuable task. Traditional forecasting methods often rely on linear models that may fall short in capturing the dynamic nature of stock trends, especially with short-term fluctuations.

This project addresses these challenges by implementing an advanced forecasting model for time series that makes use of Long Short-Term Memory (LSTM) networks and Generative Adversarial Networks (GAN). By analyzing historical stock data,the model is designed to capture intricate temporal patterns and improve forecast accuracy over conventional approaches. The integration of LSTM's sequential processing capabilities with GAN's generative power provides an innovative approach to stock prediction. The aim is to offer a tool that supports investors in optimizing their strategies by providing reliable short-term forecasts, ultimately aiding in better decision-making in a fast-paced financial environment.

II. Literature Review

A new architecture of GAN to be used in predicting the stock price from combining both the financial news' sentiment analysis and the historical stock trends. The system thus uses a Naive Bayes Classifier in the analysis that feeds into a model with LSTMs in predicting the stock prices. The GAN's Generator uses the outputs from the LSTM; MLP, Utilizing a multi-layer perceptron as the discriminator to classify whether predictions are either real or fake. It's a hybrid model that combines market sentiment with traditional financial data to try to enhance accuracy in the stock price predictions[1].

A model called TK-GAN as a new approach for predicting the price of stock by GAN-based approach via financial data and financial text reports. It adopts domain-specific fine-tuning of BERT which utilizes Soft Attention for enhancing feature extraction, and uses AdamK optimizer for optimal learning rate adaptation. Experimental results using the Kweichow Moutai stock dataset indicate very good accuracy and low MAE (0.01949) and MSE (0.00091) values. The study indicates that emotional variables and textual variables could be applied but leaves room for further investigation into their combination and the addition of multiple data sources with financial statements and charts of volatility, since these could add richness to cross-modal learning and predictive performance.[2].

The idea of an unsupervised learning of normal market behaviors by the framework of GAN for detecting stock price manipulation using LSTM networks is proposed. The proposed model was trained only with data containing normal trading cases. The testing is done with manipulated ones. The anomaly behaviors of manipulation can be classified effectively, obtaining an accuracy of 68.1 % toward identifying pump-and dump schemes on the SET data set. Such an approach reveals the possibility of GANs in the anomaly detection domain, which doesn't require labeled manipulation data.[3].

The optimization of Long Short Term Memory (LSTM) models for prediction of stock price is examined in this study in the Indian market. This is a difficult endeavor because of noise, seasonality, and long-term trends. It compares stateful and stateless LSTM models and modifies the quantity of hidden layers. The paper provides important insights into designing LSTM for time series forecasting by addressing the dearth of documented criteria for LSTM hyperparameter selection[4].

The study uses OHLCV data and a combination of statistical and deep learning methods to investigate time series forecasting of Amazon stock price changes. It assesses LSTM and GAN models in addition to ARIMA and Fourier models, using sentiment analysis to take market sentiment into consideration. The study offers important insights into financial forecasting by demonstrating how well sophisticated deep learning methods—in particular, LSTM with sentiment integration—predict stock price fluctuations as opposed to absolute values.[5].

The Least Squares Generative Adversarial Networks (LSGAN) uses a multi-step-ahead stock market prediction approach. To prepare the data for preprocessing, the model creates technical indicators, reduces noise using wavelet transformation, and eliminates outliers using z-score. By using a least-squares loss function rather than binary cross-entropy, the LSGAN increases the accuracy of its predictions for the S&P 500 index. The findings of the simulation show that LSGAN outperforms conventional models in stock price predicting.[6].

A multi-graph convolutional adversarial framework for stock price prediction VGC-GAN that does not rely on predefined graphs and captures intra-stock correlations. The model combines a GAN architecture with GRU and a Multi-Graph Convolutional Network (Multi-GCN) as the generator, supported by a CNN discriminator, to create numerous correlation graphs using historical data. Validated on real-world datasets, the model performs better in detecting hidden correlations and temporal connections by utilizing Variational Mode Decomposition (VMD) for noise reduction.[7].

The feature extraction and temporal processing to integrate present a GAN-TrellisNet multi-factor stock price prediction method. CNN serves as the discriminator and TrellisNet as the generator in this architecture to combine the temporal capabilities of RNN with the feature extraction skills of CNN. Input data richness is improved by a multi-factor approach that includes "alpha158+OCHLVC" factors. Its efficacy in quantitative finance is demonstrated by experiments conducted in a variety of markets, which show better accuracy than conventional GAN-based techniques in terms of RMSE, MAE, MSE and MAPE[8].

To predict the stock values of five companies listed on the Indian National Stock Exchange (NSE) using LSTM and GAN models. In the GAN framework, a dense neural network acts as the generator, and LSTM as the discriminator, which uses past stock price data to forecast closing prices for the following day. To replicate actual trading situations and assess how various intervals affect prediction accuracy, a rolling segmentation technique is used for dataset division.[9].

A CNN-based discriminator is combined with ARIMA, An AGAN-enhanced nonlinear fusion model for stock price prediction is produced by combining attention-based CNN (ACNN) and LSTM in the generator. The hybrid model successfully integrates temporal and frequency-domain data by utilizing the attention mechanism and GAN framework. The model's sophisticated feature extraction and prediction capabilities are demonstrated by experimental findings spanning historical datasets, which demonstrate notable gains in tracking stock price fluctuations when compared to baseline methods[10].

The effects of training dataset size and input feature on GAN-based stock prediction model performance investigated. It concludes that two to three sets of characteristics, such as core price, news score, and economic activity data, are frequently adequate for making accurate forecasts. The study shows that the influence of various features changes with the size of the training sample, with economic and mathematical data helping larger sets and news information helping smaller ones. For the best model performance, the study highlights how crucial it is to choose input features and dataset size wisely[11].

A stock price prediction technique that combines the GAN-TrellisNet model with sentiment analysis. It creates a sentiment index from stock-related comments using an LSTM-CNN-based sentiment analysis model, which is then included in the training data along with conventional stock data. In contrast to the ConvLSTM and GAN-LSTM models, the GAN-TrellisNet model, which employs CNN serves as the discriminator and TrellisNet as the generator, increases prediction accuracy while cutting down on training time. Data from three indices and ten equities are used to validate the strategy, which demonstrates improved performance[12].

A deep learning approach that combines technical indicators with Autoencoder Long Short-Term Memory (AE-LSTM) networks to predict stock prices. In order to remove noise and find anomalies using the z-score method, the model uses wavelet processing. The LSTM network forecasts the stock's closing price using the data that the autoencoder obtains. The results, which were assessed on the S&P 500 index, show that the AE-LSTM technique outperforms GAN-based models in daily adjusted closing price prediction[13].

The SF-GAN model for stock market prediction is presented in this research. It combines a Convolutional Neural Network (CNN) as the discriminator and a State Frequency Memory Neural Network (SFM) as the generator. The goal of the SF-GAN architecture is to decrease prediction errors while increasing the efficiency of stock closing price and trend forecasts. The efficiency of this method in predicting stock market trends is shown by experimental findings[14].

The Temporal Convolutional Networks-Generative Adversarial Nets (TGAN) model for stock market price prediction is presented. It combines Convolutional Neural Networks (CNN) as the discriminator and Temporal Convolutional Networks (TCN) as the generator. With reduced RMSE values for both single-step and multi-step predictions, experimental data demonstrate that TGAN performs better than conventional models like ARIMA, LSTM, and GRU in forecasting the closing prices of Apple's stock. The model's practical effectiveness was highlighted by its good performance across a variety of stock data[15].

Study optimizes a combined Generative Adversarial Network (GAN) and Long Short-Term Memory (LSTM) model for stock price prediction using a Genetic Algorithm (GA). To enhance model performance, GA is used to adjust hyperparameters such as the LSTM parameters and the size of the training data window. According to the experimental findings, the GA-optimized model performs better than the original GAN-LSTM model and forecasts stock prices with greater accuracy[16].

In order to forecast the price of stocks, this study investigates the optimization of combined Generative Adversarial Network (GAN) and Long Short-Term Memory (LSTM) model using a Genetic Algorithm (GA). The training data window size and LSTM parameters are two examples of hyperparameters that can be optimized using GA to enhance model performance. The GA-optimized model works better than the original GAN-LSTM model, according to the experimental data, and predicts stock prices with greater accuracy[17]

III. METHODOLOGY

The Fig.1 provided represents the workflow for time series forecasting using LSTM (Long Short-Term Memory) and GAN (Generative Adversarial Network) models. The procedure is divided into a number of crucial steps, all of which strive toward the ultimate goal of producing precise stock price forecasts using past data.. Here is an in-depth explanation of each step:

The forecasting process starts with the **Data Collection** phase, where a comprehensive dataset is acquired for analysis. The dataset comprises 2,497 individual data points and 36 variables, which may include stock prices, trading volumes, economic indicators, and other relevant features.This dataset acts as the starting point for all pipeline phases that follow.

Once the data is collected, it enters the **Data Preprocessing** phase, which is crucial for ensuring that the models receive clean and standardized data for training. This phase includes handling missing values and normalization. The dataset is analyzed to identify any missing or unusual values, and methods like forward/backward filling, mean imputation, or more sophisticated approaches are used to address data gaps.The data is normalized to bring all variables onto a similar scale, often between 0 and 1. This step is essential for models like LSTM and GANs, which are affected by the size of the supplied data. Normalization helps accelerate model convergence and improves training stability.

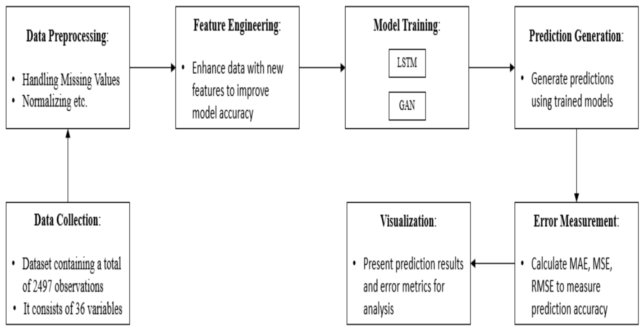

Fig. 1 Block Diagram

After the data has been pre-processed, feature engineering focuses on creating new features or altering pre-existing ones in order to improve the models' prediction ability. This may include making available time-related features such as moving averages, lagged variables, and rolling windows, which help most models capture temporal dependencies in the data. Domain-specific improvements, which could comprise financial metrics such as Relative Strength Index (RSI), Moving Average Convergence Divergence (MACD) among others might be useful in anticipating the stock price. Because it enhances the dataset and enables the model to discover intricate correlations hidden within it, feature engineering is a crucial step in the process.

The preprocessed and feature-engineered data is entered into two distinct models during the Model Training phase. Because it is trained first and uses its design to handle sequential data and identify long-term dependencies, the LSTM model is especially good for time series activities such as forecasting stock prices. The model learns from the input data and produces a baseline prediction by understanding sequential patterns and trends. In parallel, a GAN is trained, consisting of two components: the discriminator, which distinguishes between data produced by the generator and data that is real, and the generator, which creates artificial data that replicates actual stock price patterns. In order to test the generator's attempt to provide realistic predictions by distinguishing between synthetic and actual data, the generator and discriminator collaborate. This adversarial training helps the GAN learn complex and nuanced data distributions, improving the overall quality of predictions.

Once the LSTM and GAN models have been trained, they are used to generate stock price predictions. The **Prediction Generation** phase involves generating forward-looking stock price predictions using both models and comparing or combining the predictions to identify the most accurate forecast.

To ascertain how well the trained models work, the **Error Measurement** stage is essential. Metrics like Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE) are used to evaluate the predictions. These indicators aid in identifying the top-performing model and, if required, direct additional tuning and optimization.

The last stage is **Visualization**, there the results of the prediction and error metrics are displayed to ensure that users can easily interpret the model's performance and understand the forecast trends. The visual representation may include prediction plots showing actual versus predicted stock prices, error graphs for model comparison, and dashboard views providing an accessible and real-time overview of model predictions and performance metrics.

This comprehensive process enables accurate and insightful stock price forecasting. Starting from data collection and preprocessing, the workflow systematically improves data quality and model training, employing LSTM for sequential learning and GAN for capturing complex patterns. The integration of robust error measurement and visualization allows for effective model evaluation and interpretation, ensuring reliable forecasting outputs for financial decision-making.

Methodology Background:

LSTM (Long short-term memory):

Hochreiter and Schmidhuber were the first to propose this specific design within the RNN family in 1997 (Cho et al., 2014, 1724-1734), which, unlike feedforward neural networks that read the data one point at the time, has feedback loops that allow it to process sequences of data. The forget gates, output and input are some of the essential parts of an LSTM network. The vanishing gradient issue that typically arises during the training of traditional RNNs was specifically addressed by this approach. Another kind of memory cell that has the ability to add or remove information from the cell state selectively is called an LSTM. Thus, the intrinsic design of LSTM resolves the issues of disappearing gradients and exploding gradients that plague conventional RNNs. In applications like classification and prediction that use time-series data, LSTM has emerged as a very powerful tool in recent years.

GRU

Gating techniques are used by the GRU, a variant of the RNN, to control the information flow between cells. In fact, Kyunghyun Cho et al. created the GRU in 2014 as a variation of the LSTM. It has two gates: an update gate and a reset gate, which determine what should be retained and what should be thrown away. The GRUs also easily overcome the vanishing and exploding gradient problems which have traditionally affected the traditional RNNs. While LSTM has more parameters, GRUs have fewer: It maintains both short-term and long-term memory in the hidden state, bypasses one of the gates, and does not utilize the cell state. On some smaller, less frequent datasets, recent research has indicated that GRUs might even perform better than LSTMs.

GAN

From the perspective of zero-sum, non-cooperative game theory, a GAN is basically a minimax optimization problem. A GAN would traditionally be composed of two primary components: the discriminator and the generator.The purpose is to deceive the discriminator by producing instances that resemble actual data; however, the discriminator must ascertain whether the input example is produced or real. The Deep Learning community has recently shown a significant deal of interest in this paradigm. From the basic GAN framework, researchers continue to improve performance by changing the network structure and loss functions. Among the others are Conditional GAN (CGAN), which refines the generator by including additional label information. Next is the Wasserstein GAN (WGAN), which enhances the loss function with the Wasserstein distance. Another is WGAN with Gradient Penalty, introducing regularization. Lastly, there's Cycle GAN, PGGAN, and SAGAN, all of which were known to make structural alterations.

Basic GAN

The basic idea in the original GAN framework is that the loss function must be derived from the KL or Jensen-Shannon divergence. In essence, cross-entropy minimizes KL-JS divergence when training a GAN model by using this to lessen the gap between the two distributions.

Train the discriminator in this experiment to maximize the likelihood that samples will be correctly labeled, which is its objective function. Let's define the discriminator's goal function as follows:

$$V_{\mathrm{D}} = \frac{1}{m}\sum_{i=1}^{m}\log D(y_i) + \frac{1}{m}\sum_{i=1}^{m}\log(1 - D(G(x_i)))$$

Here, G(xi) is the generated or fake data that the generator produces, x is the input data, and y is the target from the dataset.

It is trained to minimize its own objective function:

$$V_{\mathrm{G}} = \frac{1}{m}\sum_{i=1}^{m}\log(1 - D(G(x_i)))$$

The loss function of discriminator's, which is used to calculate the loss during GAN training, is:

$$L_{\mathrm{D}} = -\frac{1}{m}\sum_{i=1}^{m}\log D(y_i) - \frac{1}{m}\sum_{i=1}^{m}\log(1 - D(G(x_i)))$$

Additionally, the loss function of generator's is:

$$L_{\mathrm{G}} = -\frac{1}{m}\sum_{i=1}^{m}\log D(G(x_i))$$

During the training phase, the goal is to reduce these loss functions in order to achieve optimal model performance.

1.5 WGAN-GP

The discriminator in a standard GAN fails to be strong enough and thus has a slow and unstable training procedure. Somewhat variation which addressed the stability improvement together with the performance is a Wasserstein GAN with Gradient Penalty.

For improved GAN training, WGAN-GP takes advantage of the Wasserstein distance idea, commonly referred to as EMD (Earth-Mover Distance). The Wasserstein distance is used to transform one data distribution to another. can be thought of as the lowest possible cost. The largest lower bound (infimum) of any transportation plan to convert the actual data distribution Pg to the generated one Pr—the cost of the least expensive plan—is the formal definition of this concept in mathematics:

$$W(P_r, P_g) = \inf_{\gamma \in \Pi(P_r, P_g)} \mathbb{E}_{(x,y)\sim\gamma}[||x - y||]$$

Define γ as every transport plan that can be thought of, and γ as the collection of all conceivable distributions of Pr and Pg together. Applying the calculation of Kantorovich-Rubinstein duality, the computation of Wasserstein distance can be reduced to

$$W(P_r, P_g) = \sup_{f:\|f\|_L \le 1} \mathbb{E}_{x\sim P_r}[f(x)] - \mathbb{E}_{x\sim P_g}[f(x)]$$

The function f is one that meets the 1-Lipschitz constraint, and sup stands for the least upper bound:

$$|f(x_1) - f(x_2)| \le \|x_1 - x_2\|$$

The WGAN-GP uses a gradient penalty to enforce the Lipschitz constraint. Define a differentiable function f to be 1-Lipschitz if its gradient norm everywhere is less than or equal to 1, that is, $\| \nabla f \|_2 \le 1$. Here, If the target is not met by the gradient norm, the model is penalized.

Unlike the Basic GAN, WGAN-GP does not employ sigmoid function in the discriminator, but it outputs a scalar score for the discriminator rather than indicating probability about how "real" the input data is. Moreover, A gradient penalty is used by WGAN-GP in the discriminator to keep the training stable.

	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. The WGAN-GP loss function and Basic GAN comparison

1.1 The Generator

Due to its stability, the GRU is used as the generator in the GAN model. Close,Open, Low,High, 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, the absolute and angle of 3, 6, and 9 components, and News are among the 36 features of the dataset, which covers a ten-year history of stock prices.

Since it is a multi-step ahead prediction task, the generator is defined by the input and output stages. The generator will receive three-dimensional data, such as features, input steps, and batch size. Nevertheless, the output will include output stages and batch size. For a well-functioning generator, three GRU layers have been used consisting of 1024, 512, and 256 neurons in each. These are followed by two Dense layers, where the number of neurons in the final layer correlates to the number of output steps.

1.2 The Discriminator

To ascertain if the input data is genuine or fake, the GAN model employs a Convolution Neural Network as the discriminator. The input might be either the original data or the data produced by the generator.

Three 1D convolution layers make up its construction: 32 neurons make up the first convolution layer, 64 neurons make up the second, and 128 neurons make up the third. After that, there are three dense layers of 220, 220, and 1 neuron each. With the exception of the output layer, which use Sigmoid during GAN training, and WGAN-GP, which employs linear activation, all layers use the leaky ReLU as their activation function. The Sigmoid function indicates if the input data is authentic or fraudulent by returning an output of either 0 or 1.

1.3 The Architecture of GAN

The proposed GAN has a structure consisting of the above-defined generator and discriminator architectures in combination.

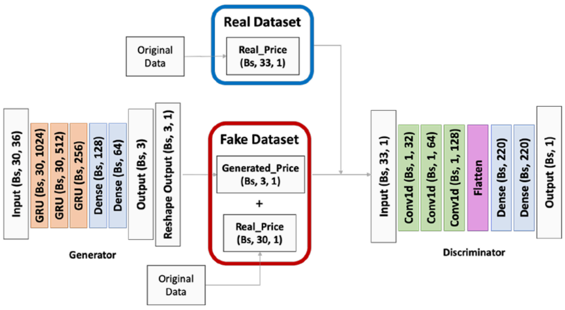

Fig. 2 GAN Structure

The role, as discussed in the background theory, is utilized, and cross-entropy is utilized in the GAN model in order to calculate the loss for both the discriminator and the generator. It particularly enhances the ability of learning categorization by the discriminator through raising the duration of data and mergingthe stock price that was produced using the previous stock price from the input stages.

IV. Results and discussion

1.1 Training of our model

Using past data of thirty days, it aimed to predict the closing stock price for the next three days. In the present project, the forecasting model is trained by using 36 features that might influence the prices, along with the historical closing price.

Training uses seventy percent of the dataset, while thirty percent is used for testing. Predictive scenarios with and without an unexpected occurrence are the two halves of the testing phase. The unforeseen event for this project is COVID-19 in 2020.

1.2 Experimental and results

Root Mean Square Error (RMSE), the indicator used in this study to assess each model's performance, is known as follows:

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

The N is the number of the data points, x_i is actual stock price, and $\hat{x}$ is predicted stock price,The RMSE of each model is assessed on the testing data (with and without 2020) in order to evaluate the models created for this research.

1.2.1 LSTM

A bidirectional LSTM is the initial layer of the provided LSTM. In this work, a learning rate of 0.001 is applied to the models using the Adam optimizer. The baseline model uses this collection of stock prices as input for the GAN model, which is trained across 50 epochs with a batch size of 64. Ten years of historical data and 36 connected features make up the entire dataset under review. After the total data was separated into training and testing sets, the testing dataset began on July 21, 2017.

Fig.3

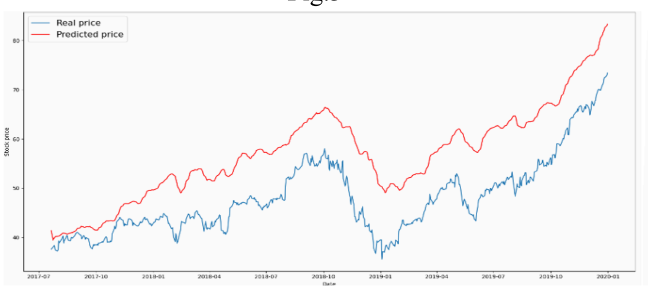

Fig.4

The results of LSTM along with the forecasted value for the year 2020 are shown in Fig. 3. The actual stock price is marked by the blue line and the forecasted stock price is marked by the red line. The RMSE is found to be 6.60. Until the last day of May 2020, the projected stock price appears to be a bit higher than the actual stock price. Then, the projection aligns much closer with the real stock price. Result for the case, in which 2020 is excluded is presented in Fig. 4. Then, in this case, the RMSE increases to 9.42, that is several folds greater than the result, including 2020.

1.2.2 GRU

The GRU model is the second basic model used in this investigation. This model is built using two GRU layers and trained for 50 epochs with the Adam algorithm acting as the optimiser. The learning rate is set at 0.0001 and the batch size is set at 128.

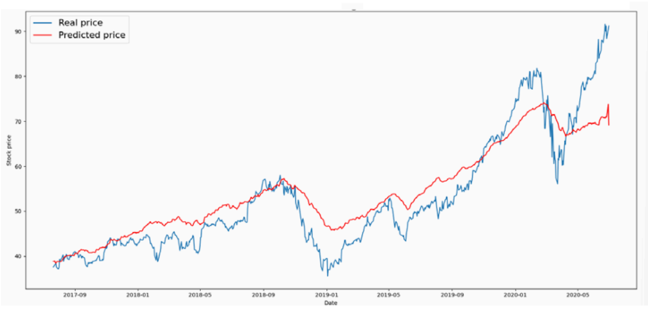

Fig.5

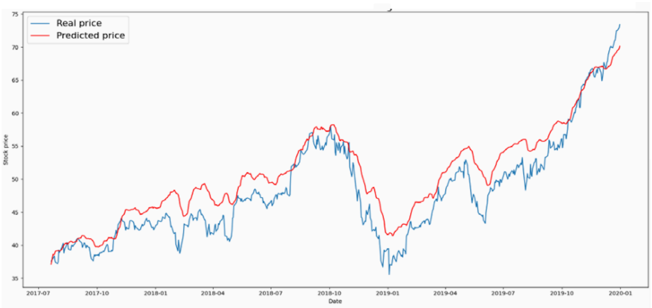

Fig.6

Fig. 5 shows the GRU model without incorporating 2020, with an RMSE of 5.33. From the above figure, it has been seen that before the month of May 2020, the LSTM model is inferior to the GRU model. From the above figure, it has been seen that the projection collapses after May 2020. Fig. 6 depicts the GRU model without incorporating 2020, with an RMSE of 4.08. If it does not assume any anomalous events at the time of making predictions, then the GRU model largely outperforms.

1.2.3 Basic GAN

This paper's methodology section suggests the GAN model's structure. This model uses the Adam algorithm as its optimiser, which has a learning rate of 0.00016. This dataset is used to train the model across 165 epochs with a batch size of 128.

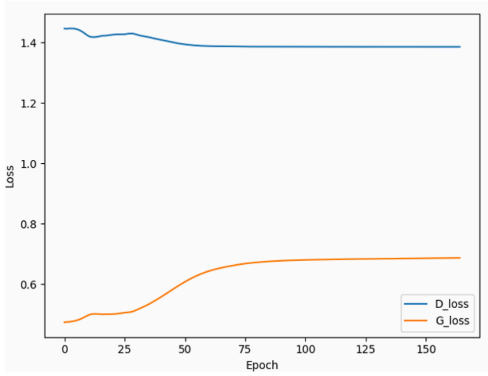

Fig.7

The loss plot for the basic GAN model is displayed in Fig. 7, where the orange line represents the generator's loss curve and the blue line represents the discriminator's loss curve. The discriminator's loss was higher than the generator's in the beginning, and as training progresses, both curves of loss gradually flatten out.

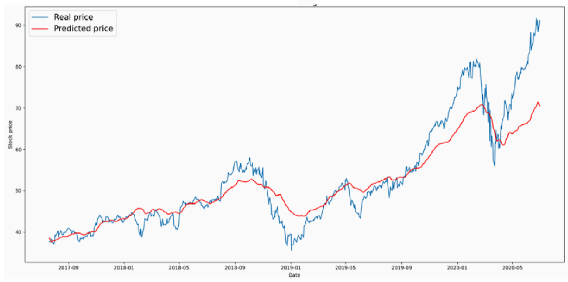

Fig. 8

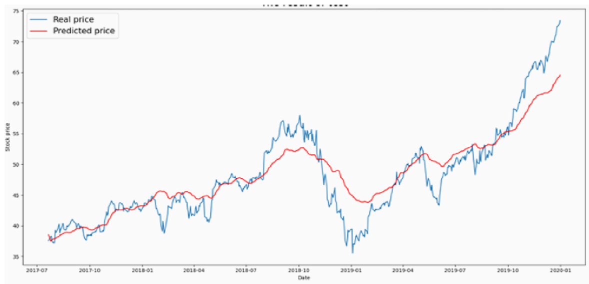

Fig.9

As we can see in Fig. 8, this results from the basic GAN model; the RMSE of 5.36 depicted a huge gap between the actual price and price prediction for 2020 along with a sharp spike, which possibly resulted from an event that was not foreseen, such as COVID-19. In Fig. 9, if 2020 predictions are excluded from the basic GAN model, the RMSE goes down to 3.09. This indicates that the underlying GAN model for the predictive task is superior to the basic models in the absence of the unknown event.

1.2.4 WGAN-GP

In the approach section, the author then specifies the WGAN-GP model's structure. Once more, the Adam algorithm serves as the optimizer in this case, but it is applied with a learning rate of 0.0001. This dataset was used to train the model for 100 epochs with a batch size of

128. Additionally, the discriminator is trained once and the generator is trained three times.

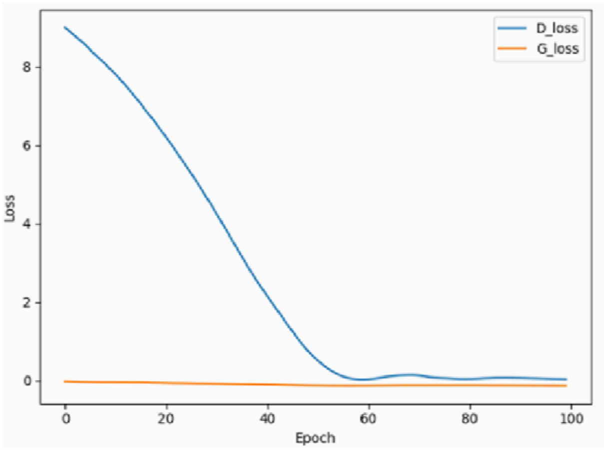

Fig. 10

In Fig 10, the loss plot of the WGAN-GP model. In this figure, the path of loss for the discriminator is given in blue, and the path of loss for the generator is shown in orange. The loss of discriminator is decreased toward 0. Comparing the loss path of Basic GAN, the learning of the discriminator was much better in WGAN-GP.

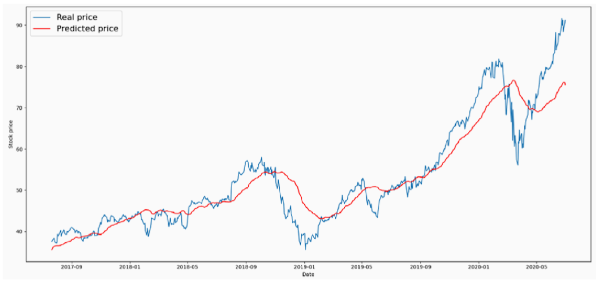

Fig.11

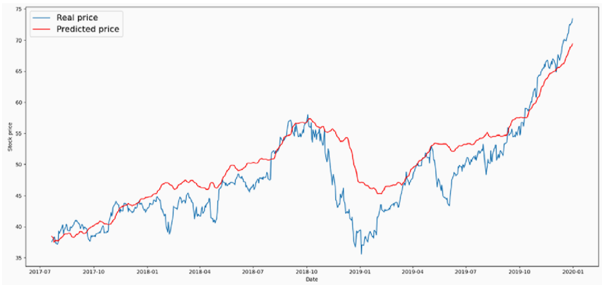

Fig. 12

Fig. 11. The WGAN-GP prediction, with an RMSE of 4.77-best over all the models. Again, as we have seen with the base GAN model, the prediction starts deviating strongly from the real price at 2020 because of the sudden COVID-19 incident. Lastly, as shown in Fig. 12, the test data from 2020 is eliminated. It performs worse than the base GAN model since the forecast's RMSE is raised to 3.88.

Evaluation

The testing RMSE and training RSME for various models are contrasted in the table.

	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

On the training set, GRU does the best. Yet on the test set, if I include the data from the period of COVID 19, WGAN-GP performed the best. But if I do not include the period, then Basic GAN performs the best. Still, on average, the GANs models have outperformed the baseline traditional models with regard to our result.

Evaluating the model's performance, the table brings out the comparative RMSE analysis for training and testing various architectures- LSTM, GRU, Basic GAN, and WGAN-GP. It appears that as far as the training dataset is concerned, GRU has generated the minimum RMSE at 1.00, and therefore this model shows that it performed extremely well in learning the proper training patterns of data. Other models are LSTM, Basic GAN, and WGAN-GP with higher values of training RMSE than ideal. This means GRU may learn the central data patterns better in training and subsequently minimize errors well.

When taking 2020 into account in the evaluation of testing performance, WGAN-GP has the lowest testing RMSE at 4.77, resistant to turbulent periods and subsequent patterns, which include the pandemic period. Without the data from 2020, relative model performance varies, where Basic GAN holds the best value among the others with RMSE at 3.09, though with a stronger predictive capacity within stable circumstances. Overall, compared with the LSTM and GRU architectures, GAN-based models show superior capacity in dealing with complicated data and robust forecasting power when faced with uncertain or fluctuating scenarios. This makes it clear that GAN models can tolerate changes in data patterns; hence, there is a potential advantage over traditional sequence models for certain applications.

V. Conclusion

The offered system proposes a GAN-based model for creating the stock price forecasting problem, where CNN is the discriminator and GRU is the generator. GRU is the architecture that supports sequential learning well, and CNN is robust in pattern recognition, which improves the precision of the prediction. From the experiment, the GAN-based models like simple GAN and WGAN-GP have also outperformed the classic models, such as LSTM and GRU. WGAN-GP seems to function amazingly in the highly volatile markets like the COVID-19 pandemic, showcasing some superior robustness and stability. In contrast to the above statement, the basic GAN performs amazingly well in the case of stable market conditions and has better accuracy than all others. Although this produces promising results, hyperparameter tuning really is quite a task altogether regarding stability and performance in the model. When any model fails to fine-tune accordingly, this can lead to instability, improper convergence, or suboptimal performance. The GAN architecture is quite complicated and interacts with parameters, making it a time-consuming and computationally demanding process related to manual optimization involved. I use these for more refined hyperparameter optimization, such as the reinforcement learning methods Rainbow, Q-learning, and PPO. These can actually automate and optimize the tuning procedure and ensure that these models can run much more steadily and effectively.

This paper will focus on the unique characterization of a transformation in GANs with respect to time-series forecasting and process complex dynamic financial data with some scope for error. I strongly believe that a well-designed GAN can capture underlying intricate patterns of a financial forecast and is rapidly responsive to the crisis situation, and therefore, further development should focus on adaptive learning strategies and self-tuning mechanisms for improvement. Combining automation of optimization with the benefits derived from reinforcement learning promises much better performance compared to the actual results delivered by GAN models so far in highly dynamic and unpredictable financial environments and is therefore expected to significantly improve reliability and accuracy in GAN models.

VI. References

1. R. Jadhav, S. Sinha, S. Wattamwar and P. Kosamkar, "Leveraging Market Sentiment for Stock Price Prediction using GAN," 2021 2nd Global Conference for Advancement in Technology (GCAT), Bangalore, India, 2021, pp. 1-6, doi: 10.1109/GCAT52182.2021.9587497

2. Zhang, Rui, and Vladimir Y. Mariano. "Integration of Emotional Factors with GAN Algorithm in Stock Price Prediction Method Research." IEEE Access (2024).

3. T. Leangarun, P. Tangamchit and S. Thajchayapong, "Stock Price Manipulation Detection using Generative Adversarial Networks," 2018 IEEE Symposium Series on Computational Intelligence (SSCI), Bangalore, India, 2018, pp. 2104-2111, doi: 10.1109/SSCI.2018.8628777.

4. Bhardwaj, Muskaan, Aditya Roy, and Saurabh Bilgaiyan. "StockGAN: Enhancing Stock Price Prediction with GAN and Sentiment Analysis." In 2024 Second International Conference on Networks, Multimedia and Information Technology (NMITCON), pp. 1-6. IEEE, 2024.

5. Bhardwaj, Anjul, and Uday Pratap Singh. "An Empirical Study on: Time Series Forecasting of Amazon Stock Prices using Neural Networks LSTM and GAN models." In Proceedings of the 5th International Conference on Information Management & Machine Intelligence, pp. 1-10. 2023.

6. Faraz, Mehrnaz, and Hamid Khaloozadeh. "Multi-step-ahead stock market prediction based on least squares generative adversarial network." In 2020 28th Iranian Conference on Electrical Engineering (ICEE), pp. 1-6. IEEE, 2020.

7. Ma, Dongbo, Da Yuan, Maojun Huang, and Ling Dong. "VGC-GAN: A multi-graph convolution adversarial network for stock price prediction." Expert Systems with Applications 236 (2024): 121204.

8. Liu, Wenjie, Yebo Ge, and Yuchen Gu. "Multi-factor stock price prediction based on gan-trellisnet." Knowledge and Information Systems (2024): 1-22.

9. Maiti, Ayan. "Indian stock market prediction using deep learning." In 2020 IEEE REGION 10 CONFERENCE (TENCON), pp. 1215-1220. IEEE, 2020.

10. Xu, Yingcheng, Yunfeng Zhang, Peide Liu, Qiuyue Zhang, and Yuqi Zuo. "GAN-Enhanced Nonlinear Fusion Model for Stock Price Prediction." International Journal of Computational Intelligence Systems 17, no. 1 (2024): 12

11. Gao, Yufeng, Jingfan Feng, and Zheng Tao. "Exploring the Impact of Input Features on GAN-Based Stock Prediction." In 2023 IEEE 5th International Conference on Power, Intelligent Computing and Systems (ICPICS), pp. 219-223. IEEE, 2023.

12. Yebo, G. E., L. I. U. Wenjie, and G. U. Yuchen. "Stock Price Prediction Method by Fusing Sentiment Analysis and GAN-TrellisNet." Journal of Computer Engineering & Applications 60, no. 12 (2024).

13. Faraz, Mehrnaz, Hamid Khaloozadeh, and Milad Abbasi. "Stock market prediction-by-prediction based on autoencoder long short-term memory networks." In 2020 28th Iranian conference on electrical engineering (ICEE), pp. 1-5. IEEE, 2020.

14. Jiang, Jie. "Stock market prediction based on SF-GAN network." In 2021 6th International Symposium on Computer and Information Processing Technology (ISCIPT), pp. 97-101. IEEE, 2021.

15. Xu, Mengfei, Jiejie Chen, Ping Jiang, Baiyan Hu, and Honggang Yang. "Stock Market Price Prediction Based on TGAN." In 2023 9th International Conference on Control Science and Systems Engineering (ICCSSE), pp. 246-251. IEEE, 2023.

16. He, Bate, and Eisuke Kita. "GA-based optimization of generative adversarial networks on stock price prediction." In 2021 International Conference on Computational Science and Computational Intelligence (CSCI), pp. 199-202. IEEE, 2021.

17. Wang, Lu, and Zhensheng Huang. "Research on Stock Price Volatility Prediction Based on Generative Adversarial Network." In 2021

International Conference on Networking, Communications and Information Technology (NetCIT), pp. 332-335. IEEE, 2021.

18. Gan, Kim Soon, Kim On Chin, Patricia Anthony, and Sim Vui Chang. "Homogeneous ensemble feedforward neural network in CIMB stock price forecasting." In 2018 IEEE International Conference on Artificial Intelligence in Engineering and Technology (IICAIET), pp. 1-6. IEEE, 2018.

19. Rezaei, Hadi, Hamidreza Faaljou, and Gholamreza Mansourfar. "Stock price prediction using deep learning and frequency decomposition." Expert Systems with Applications 169 (2021): 114332.

20. Oh, Eunkyu, Taehun Kim, Yunhu Ji, and Sushil Khyalia. "STING: Self-attention based Time-series Imputation Networks using GAN." In 2021 IEEE international conference on data mining (ICDM), pp. 1264-1269. IEEE, 2021.

21. Duan, Guoao, Mengyao Lin, Hui Wang, and Zhuofan Xu. "Deep neural networks for stock price prediction." In 2022 14th International Conference on Computer Research and Development (ICCRD), pp. 65-68. IEEE, 2022.

22. Sutradhar, Koushik, Sourav Sutradhar, Iqbal Ahmed Jhimel, Suneet Kumar Gupta, and Mohammad Monirujjaman Khan. "Stock market prediction using recurrent neural network's LSTM architecture." In 2021 IEEE 12th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON), pp. 0541-0547. IEEE, 2021.

23. Wei, Shizhao, Shungang Wang, Siyi Sun, and Yan Xu. "Stock ranking prediction based on an adversarial game neural network." IEEE Access 10 (2022): 65028-65036.

24. Xu, Huize, Yuhang Zhang, and Yaoqun Xu. "Promoting Financial Market Development-Financial Stock Classification Using Graph Convolutional Neural Networks." IEEE Access 11 (2023): 49289-49299.

25. Thangamayan, S., Bhupendra Kumar, K. Umamaheswari, M. Arun Kumar, Dharmesh Dhabliya, S. Prabu, and N. Rajesh. "Stock price prediction using hybrid deep learning technique for accurate performance." In 2022 International Conference on Knowledge Engineering and Communication Systems (ICKES), pp. 1-6. IEEE, 2022.