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

A Project report submitted in partial fulfillment of the

requirements for the degree of

Bachelor of Technology

by

Ruturaj Patil	**Roll No.58**
Shantanu Pawar	**Roll No.64**
Nagesh Pujari	**Roll No.75**

Under the guidance of

Prof. Dr. Vijay Mane

DEPARTMENT

OF

ELECTRONICS & TELECOMMUNICATION ENGINEERING

VISHWAKARMA INSTITUTE OF TECHNOLOGY PUNE

2024 - 25

Bansilal Ramnath Agarwal Charitable Trust's

VISHWAKARMA INSTITUTE OF TECHNOLOGY, PUNE - 37

(An Autonomous Institute Affiliated to Savitribai Phule Pune University)

CERTIFICATE

This is to certify that the **Project Report** entitled **"Time Series Forecasting of Stock Prices using Neural Networks LSTM and GAN"** has been submitted in the academic year **2024-25** by

Ruturaj Patil	**Roll No.58**
Shantanu Pawar	**Roll No.64**
Nagesh Pujari	**Roll No. 75**

under the supervision of **Prof. Dr. Vijay Mane** in partial fulfillment of the requirements for the degree of Bachelor of Technology in **Electronics and Telecommunication Engineering** as prescribed by Savitribai Phule Pune University.

Guide	**Head of the Department**
Name: Prof. Dr. Vijay Mane	**Name:** Prof. Dr. Medha Wyawahare
Signature:	**Signature:**

External Examiner

Name:

Signature:

ACKNOWLEDGEMENT

We are very glad to introduce our project titled "Time Series Forecasting of Stock Prices using Neural Networks LSTM & GAN." It gave us a great pleasure to present our work before the esteemed audience, and we are very grateful to our project guide, Prof. Dr. Vijay Mane Sir, who guided and co-operated with us throughout this venture. Without his expertise and support, we would definitely have had a tough time in navigating through this project. We appreciate wonderful feedback and mentorship from Prof. Dr. Vijay Mane, which has been crucial in our methodology and its success. We wish to thank him for his dedication and precious inputs to our project. We also thank our Head of Department Prof. Dr. Medha Wyawahare ma'am and the distinguished Vishwakarma Institute of Technology, Pune, for continuous support and encouragement in creating an environment of research and innovation.

Ruturaj Patil
Shantanu Pawar
Nagesh Pujari

ABSTRACT

This project focuses on the forecasting of Stock prices using advanced machine learning techniques, including Long Short-Term Memory (LSTM) networks and Generative Adversarial Networks (GANs). For these foretellings, historical stock data parameters including Low, High, Close, Open, and Trading Volume are used to train the dataset. Synthetic stock price data will be generated using GANs, which will in turn augment the training set and increase the accuracy and robustness of the model by mitigating this problem of insufficient data. LSTM networks, especially known to capture temporal dependencies in time series data, will be used to predict the future stock prices by finding relevant patterns and trends within historical data.

Thus, the performance of the model is measured through metrics like Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE), showing that it is quite accurate in prediction for trends of stock prices. Though it is not operational currently, it provides desirable predictions with graphics analytics to investors and analysts as well for making decisions based on reality. This approach, while targeted for stock price forecasts, can thus be appropriately applied to other financial, even non financial time-series data streams, providing the scalability for broader applications. The continuous refinement of the model will aim toward better ability in terms of forecasting, making it a promising tool in financial analysis decision-making.

INDEX

List Of Figures

List Of Tables

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

In today's financial industry, clear and timely prediction of stock price is necessary to be done for investment decisions and risk mitigation. The stock price prediction schemes, primarily traditional ones, are based on techniques such as ARIMA or LSTM; however, they lack the sensitivity to intricate patterns or external factors that may influence the market. With the introduction of deep learning, much more complex models, such as GANs, have been developed and hold much promise in generating an even more accurate prediction.

A precise forecast of the stock price is essential for wise investment choices.. Traditional methods, such as ARIMA and LSTM, are mainly limited in their approach, whereas deep learning models such as GANs actually provide better accuracy, though manually integrating news sentiment or technical indicators into these models is quite time-consuming. This project aims to automate multiple data sources, including technical indicators, sentiment analysis, and most importantly, stock prices, into a deep learning model. This paper applies the GAN to the task of the goal of improving the accuracy of prediction and simplifying data processing to enhance better finance forecasts.

1.2 MOTIVATION

Complexities and volatility inherent in the financial markets, where stock price forecasting has become a critical tool for traders, analysts, and investors to make informed decisions. Traditional methods like linear regression and ARIMA often fail to capture the dynamic, non-linear nature of stock price movements, leaving a gap for more advanced techniques. Machine learning approaches, such as Long Short-Term Memory (LSTM) networks and Generative Adversarial Networks (GAN), offer a solution by effectively modeling sequential data and complex patterns. LSTM excels in capturing long-term dependencies in time-series data, while GANs introduce a generative approach to simulate realistic data distributions. Combining these methods creates a robust framework capable of providing accurate and reliable short-term predictions, addressing the need for sophisticated tools in financial decision-making. By leveraging historical stock data, this project aims to optimize trading strategies, empower stakeholders with actionable insights, and contribute to the field of stock price prediction by bridging the gap between traditional techniques and modern financial market complexities.

1.3 PROBLEM DEFINITION AND OBJECTIVES

Accurate and timely stock price forecasting is, therefore, very crucial in organizations and among individuals performing financial analysis because this has direct implications on informed investment decisions. However, high-accuracy forecasting of stock prices remains a daunting task since stock prices are inherently volatile and unpredictable. Most existing stock price forecasting methods have inherent limitations that limit the accuracy and reliability of such predictions:

- **Traditional Forecasting Models:** Classic methods include ARIMA, moving averages, and linear regression. These are bad models because they do not catch the fluctuations and possibly the complexities involved in natural non-linear patterns as observed in stock price time-series data.Assumptions in these models are not effective in handling increased variability in the market.
- **Lack of Data Utilization:** Most models use only historical stock data as input, leaving out other variables such as market sentiment, news, and external financial indicators, which might boost the eventual forecast.

The primary objective of this study is to develop a robust and automated stock price forecasting system leveraging advanced machine learning techniques, specifically Long Short-Term Memory (LSTM) networks and Generative Adversarial Networks (GAN). The system aims to:

1. **Forecast for Stock Prices**:

   The project focuses on automating stock price predictions using historical market data. By leveraging advanced deep learning models like Long Short-Term Memory (LSTM) and Generative Adversarial Networks (GAN), the system eliminates the need for manual analysis, which is often time-consuming and prone to human error. These models can capture intricate temporal patterns and dependencies within the data, providing a systematic approach to predict future stock prices effectively.

2. **Improve Accuracy of Predictions**:

   Traditional forecasting methods, such as linear regression or basic statistical models, often struggle with the complexity and non-linear nature of stock price movements. By employing neural networks, the system can identify subtle and complex patterns in the data, including market volatility and dependencies over time. LSTM's ability to process sequential data and GAN's capability to enhance training datasets lead to higher prediction accuracy, outperforming conventional approaches and offering a competitive edge in financial forecasting.

3. **Minimize Human Intervention**:

   This project automates critical stages of the forecasting process, including data collection, preprocessing, model training, and prediction generation. Automation reduces the dependency on manual efforts, enabling a streamlined workflow where human involvement is required only for oversight or adjustments. This minimizes the risk of errors and ensures consistent, reliable, and efficient model performance, even in highly dynamic financial environments.

4. **Real-time Forecasting and Response**:

   The system is designed to provide real-time predictions of stock prices, allowing stakeholders to make informed decisions promptly. With the ability to analyze the latest market trends and behaviors as they occur, the model supports dynamic decision-making. This feature is particularly valuable in volatile markets, where timely responses can significantly impact trading strategies and investment outcomes. By offering up-to-date insights, the system empowers investors and analysts to stay ahead in the financial landscape.

CHAPTER 2

LITERATURE SURVEY

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 FAN-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].

The effectiveness of homogeneous ensemble Artificial Neural Networks (ANNs) in forecasting closing prices of the CIMB stock market is examined in this paper. The goal of the project is to improve predicting accuracy by integrating numerous ANNs. According to empirical findings, the ensemble ANN performs better in terms of prediction accuracy than a single ANN[18].

Frequency decomposition and deep learning approaches uses to handle stock price prediction problems. In order to extract deep features and time sequences for one-step-ahead forecasting, it suggests hybrid models, CEEMD-CNN-LSTM and EMD-CNN-LSTM, which combine CNN and LSTM with empirical mode decomposition (EMD) or full ensemble EMD (CEEMD). The CEEMD-CNN-LSTM model outperforms EMD-CNN-LSTM and other conventional techniques in terms of prediction accuracy, according to the results[19].

A unique approach STING that uses generative adversarial networks (GANs) and self-attention to impute missing values in multivariate time series. To capture weighted correlations across sequences, it blends GANs with bidirectional recurrent neural networks and an attention mechanism. Tests conducted on real-world datasets demonstrate that STING performs better than the most advanced techniques in terms of downstream task performance and imputation accuracy[20].

The application of deep learning techniques for stock price prediction, including GANs, CNNs, LSTMs, and Deep Reinforcement Learning (DRL). It draws attention to issues including managing the properties of raw data, correlations between equities that are comparable, and inefficiencies in information processing. The article outlines current algorithms, their evolution, and upcoming developments in stock prediction methods[21].

Investigates the use of Recurrent Neural Networks (RNNs) with Long Short-Term Memory (LSTM) architecture for stock market prediction. It focuses on utilizing LSTM's capacity to identify temporal patterns and long-term connections in financial time series data. The study shows how successful LSTM is in predicting stock prices with accuracy, which helps financial markets make better decisions[22].

Literature Review Observations :

With the help of the Literature Review of several Research Papers, we found out some popular methodologies and common practices followed:

Table. 1 Popular Techniques identified by Literature Review

Training data quality and stock market trend prediction models will be enhanced by using GAN to create synthetic stock market data.	Zhang, Kang, Guoqiang Zhong, Junyu Dong, Shengke Wang, and Yong Wang. "Stock market prediction based on generative adversarial network." Procedia computer science 147 (2019): 400-406.
Employing Generative Adversarial Networks (GAN) to anticipate stock market prices and optimize training through adversarial learning.	Romero, Ricardo Alberto Carrillo. "Generative adversarial network for stock market price prediction." CD230: Deep Learning, Stanford University (2018): 5.
Utilizing adversarial learning to represent intricate market dynamics and improve prediction accuracy, Generative Adversarial Networks (GANs) are used to predict the stock market on high-frequency data.	Zhou, Xingyu, Zhisong Pan, Guyu Hu, Siqi Tang, and Cheng Zhao. "Stock market prediction on high‑frequency data using generative adversarial nets." Mathematical Problems in Engineering 2018, no. 1 (2018): 4907423.
By combining sentiment research with stock market data, Generative Adversarial Networks (GANs) are utilized to increase the model's forecast accuracy for stock prices.	Jadhav, Rahul, Shambhavi Sinha, Soham Wattamwar, and Pranali Kosamkar. "Leveraging Market Sentiment for Stock Price Prediction using GAN." In 2021 2nd Global Conference for Advancement in Technology (GCAT), pp. 1-6. IEEE, 2021.
Incorporating market sentiment and emotional influences on stock trends into Generative Adversarial Networks (GANs) for stock price prediction improves forecast accuracy.	Zhang, Rui, and Vladimir Y. Mariano. "Integration of Emotional Factors with GAN Algorithm in Stock Price Prediction Method Research." *IEEE Access* (2024).

CHAPTER 3
METHODOLOGY

This study describes the proposal of an advanced time-series forecasting solution, emphasizing the prediction of stock prices by leveraging sequential dependencies and complex data distributions. The proposed system integrates Long Short-Term Memory (LSTM) and Generative Adversarial Networks (GAN) to improve prediction accuracy, particularly in financial datasets.

Currently, prediction of a stock price is still based on some elementary models or manual analysis of historical data, which is time-consuming and inaccurate. Traditional methods often cannot interpret complex patterns in the data; thus, they often lead to highly suboptimal results from forecasting. Moreover, they suffer from significant problems in responding to market volatility and unexpected change, making them less reliable for investors and analysts. This challenge is addressed by The Time-Series Forecasting of Stock Prices using LSTM and GAN by using automation for forecasting. Advanced deep learning techniques are exploited by the system to analyze large datasets, find hidden patterns, and predict things in the financial sector that are even more accurate and scalable.

To solve the challenge of accurately forecasting Stock prices, this project uses an advanced LSTM and GAN.

More advanced powerful machine learning technologies improve the precision of the forecast and improve the capacity to adjust to shifting consumer preferences. Below is a presentation of the solution's main elements:

- **Data Collection and Preprocessing:** The system automatically collects the historical stock price data, including Open, High, Low, Volume, and other technical indicators. Data preprocessing includes handling missing values, normalization of features, and time-series feature generation to improve the capability of the model to forecast future values.

- **LSTM for Time-Series Prediction:** The project uses LSTM, which is a kind of RNN; it captures the time dependencies in the stock price data. From past movements in the stock prices, LSTM will model long-term trends and short-term fluctuations in the market, which are very important for carrying out accurate stock price predictions.
- **GAN for Data Augmentation:** Generative Adversarial Networks then will create synthetic stock price data as similar to the actual market data as possible. This supplemental data in the training will aid the LSTM model to be a better learner; therefore, improves generalization and robustness towards market volatility.
- **Model Training and Optimization:** The trained models incorporated the LSTM and GAN models on that processed data, hyperparameter optimized to achieve maximum performance. The system ensured model adaptability to changing market dynamics with continuous improvement over time.
- **Prediction and Forecasting:** The model will predict future stock prices after being trained using past data. The confidence intervals of the predictions-both in the short and long term-can give a good understanding of the possible movements of the market.

The process initiates with **Data Collection**, where a robust dataset of 2,497 data points and 36 variables, including stock prices and trading volumes, is compiled. This dataset serves as the foundation for subsequent phases.Next, **Data Preprocessing** ensures that the raw data is cleaned and standardized. Missing values are addressed using imputation techniques, and normalization is applied to bring all features to a similar scale. This step enhances model performance and stability.

Feature Engineering follows, generating time-related features such as moving averages, lagged variables, and domain-specific indicators like MACD and RSI. These enrichments enable models to better capture temporal dependencies and underlying patterns.

In the **Model Training** phase, the data is fed into two models:

- **LSTM** learns long-term dependencies in sequential data, providing baseline predictions.

- **GAN** consists of a generator, which creates synthetic data resembling real patterns, and a discriminator, which distinguishes between real and synthetic data. Their adversarial training improves the model's ability to capture complex data distributions.

The **Prediction Generation** phase combines the outputs of LSTM and GAN to generate accurate stock price forecasts. The generated predictions are evaluated during the **Error Measurement** phase using metrics such as RMSE, MAE, and MSE. These metrics guide optimization and model selection.

Finally, the results are presented through **Visualization**, offering intuitive insights via graphs, prediction plots, and error trends. This visual output supports stakeholders in making informed financial decisions.

The integration of sequential learning (LSTM), adversarial training (GAN), and robust error evaluation ensures reliable forecasting, making this system a valuable tool for financial analysis.

Block of Proposed System:

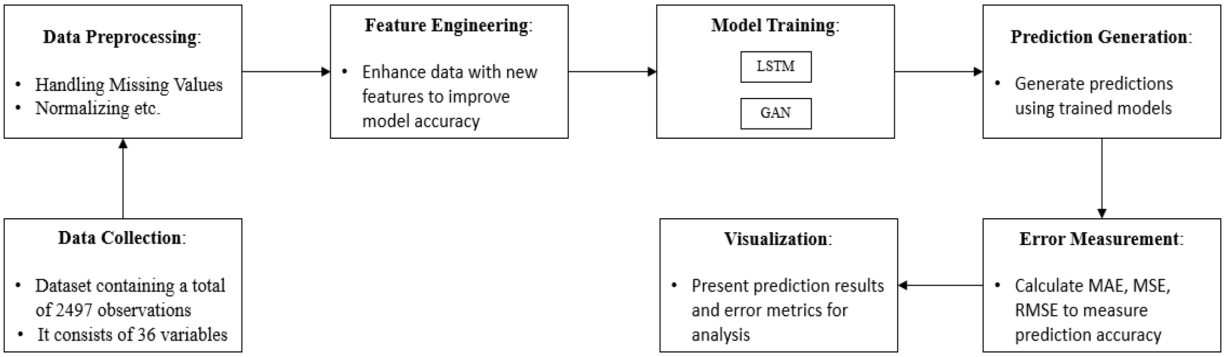
Data Preprocessing:
• Handling Missing Values
• Normalizing etc.
Feature Engineering:
• Enhance data with new features to improve model accuracy
Model Training:
LSTM
GAN
Prediction Generation:
• Generate predictions using trained models
Data Collection:
• Dataset containing a total of 2497 observations
• It consists of 36 variables
Visualization:
• Present prediction results and error metrics for analysis
Error Measurement:
• Calculate MAE, MSE, RMSE to measure prediction accuracy

Fig.1 Block Diagram of proposed system

The diagram provided represents the workflow for time series forecasting using LSTM (Long Short-Term Memory) and GAN (Generative Adversarial Network) models. The process is divided into several key stages, each contributing to the overall aim of generating accurate stock price predictions from historical 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 sensitive to the scale of input data. Normalization helps accelerate model convergence and improves training stability.

Feature Engineering, following pre-processing of the data, is concerned with generating additional new features or modifying existing features that could enhance the predictive power of the models. 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 indicators such as Relative Strength Index (RSI), Moving Average Convergence Divergence (MACD) among others might be useful in anticipating the stock price. Feature engineering is an integral part of the process since it enriches the dataset and allows the model to learn complex relationships underlying the data.

The **Model Training** phase is where the preprocessed and feature-engineered data is fed into two different types of models. For time series tasks like stock price forecasting, the LSTM model is particularly effective since it is trained first and uses its design to handle sequential data and capture long-term dependencies.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 generator, which generates synthetic data that mimics real stock price patterns, and the discriminator, which differentiates between real data and data produced by the generator. The generator and discriminator work in tandem, with the generator attempting to produce realistic predictions while the discriminator challenges it by identifying synthetic versus real data. 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.

The **Error Measurement** phase is crucial to determine the performance of the trained models. The predictions are assessed using metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE). These metrics help in determining which model performs best, guiding further tuning and optimization if necessary.

The final stage is **Visualization**, where 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.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.

Flowchart:

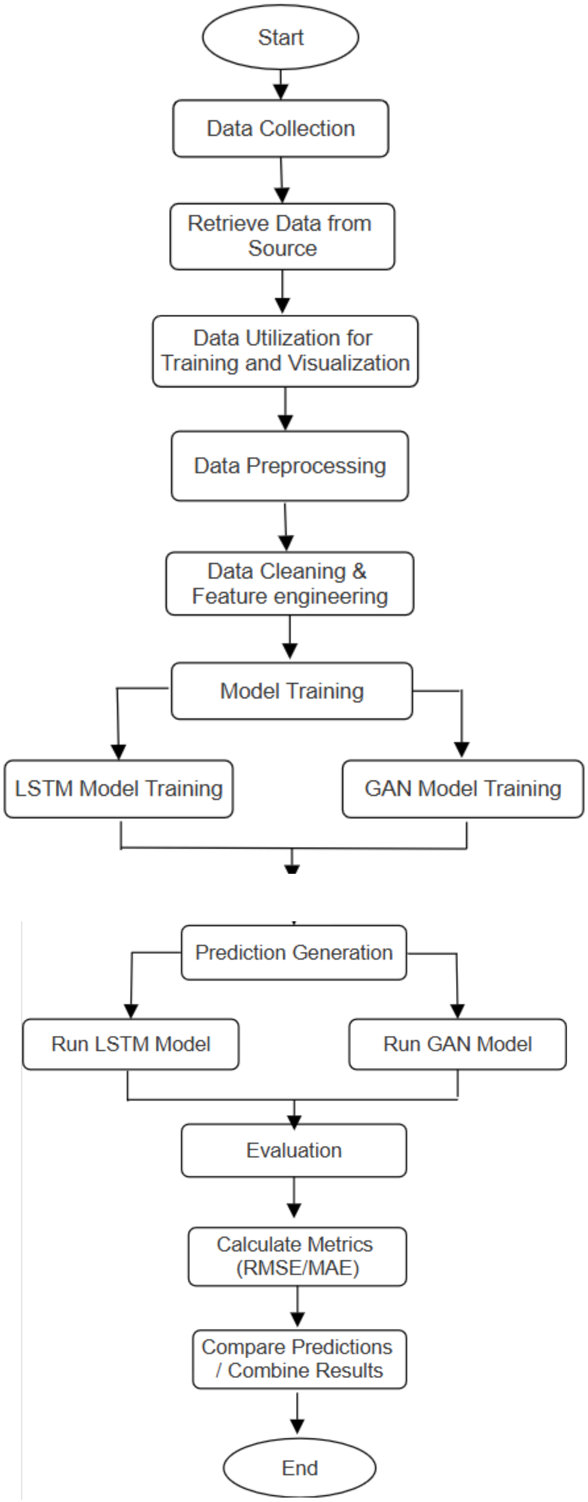
Start
Data Collection
Retrieve Data from Source
Data Utilization for Training and Visualization
Data Preprocessing
Data Cleaning & Feature engineering
Model Training
LSTM Model Training
GAN Model Training
Prediction Generation
Run LSTM Model
Run GAN Model
Evaluation
Calculate Metrics (RMSE/MAE)
Compare Predictions / Combine Results
End

Fig. 2 Workflow Diagram of Proposed System

Fig. 2 represents the detailed flow of the proposed project on data processing, training of the given machine learning models, making the predictions based on the inputs, and evaluating the performance. It is quite a pictorial way of illustrating the involved stages from data collection to visualization and even automated scheduling-structured and systematic processes. Every stage of the workflow is interconnected, with the data flowing smoothly from one step into the next, along with the activities that fine tune the models to produce accurate predictions. The steps followed in the flowchart are detailed in the proceeding sections.

❖ Data Collection:

Collecting data from raw sources for the project is the process that begins. A researcher or an application can retrieve data from online repositories, APIs, databases, or sensors in different forms, structures, and content. It is thus crucial to standardize and preprocess this varied format, structure, and content data before utilizing it in the remaining steps of the process. It is the building of infrastructures and methods necessary for the automation or facilitation of gathering data such that gathered data is valid and reliable. Data collection is an important step because it is the basis of all subsequent processes and analyses.

❖ Retrieve Data from Source

Once the data collection process is started, the next activity would be to retrieve the data from its storage place or repository, whether it may be cloud services, local servers, or databases at a remote location. Data retrieval can be done in various ways by querying databases using SQL or even accessing files through APIs. At this stage, a retrieval operation needs to be as efficient as possible, especially when it comes to large data, to prevent performance issues. The retrieved data is now ready for storage in a database or further processing at later stages.

❖ Data Utilization for Training and Visualization

Once the retrieved data is stored in the form of a raw CSV file for further processing, it ensures proper storage of data in a specific form of CSV format that is authentic, consistent, and easy to access for analysis. According to the nature of the data and the project's requirement, the CSV file is the central form of storage. This stage involves sorting the data into rows and columns so that they can be arranged in an orderly fashion for the querying and analysis to be effective. A CSV file enables efficient and safe storage of data to be fed into the other stages of analysis and modeling.

❖ Data Preprocessing

Data preprocessing is the transformation of raw data into clean and structured formats for modeling. Some of the sub processes involved in this include handling missing values, removal of outliers, and correction of inconsistencies present within the data set. It could further involve normalizing or standardizing numerical values to maintain consistency across features. Data preprocessing also ensures that the data is following the assumptions required by the algorithms used later on in the process. The performance of the predictive models is directly impacted by this improvement in the quality of the input data.

❖ Data Cleaning & Feature Engineering:

The act of locating and fixing any issues with the dataset is known as data cleaning so that it becomes accurate and usable. It includes missing values during handling by imputation or removal, elimination of duplicates, and correction of errors in data entry. In this step, inconsistent data types or formatting errors are also eradicated. Robust machine learning models' efficacy is contingent upon the absolute cleanliness of a dataset, because noisy data or incorrect information can lead to wrong results and even wrong forecasts. Data cleaning is often an iterative process where one has to inspect the dataset carefully and refine it continuously.

Feature engineering selects or even generates new features from raw data so that the performance of the machine learning model improves.Feature engineering might include the encoding categorical variables, scaling numerals, or many other related items like time features. Domain knowledge is important here to determine which characteristics are most crucial for that task. Well-engineered features can drastically improve the accuracy of models.

❖ Model Training

Model training is the step of applying machine learning algorithms to the processed data for learning the patterns or relationships that may be embodied in it. This is termed as feeding preprocessed data into a model so that it can predict or classify based on the data. Here, depending upon the nature of the problem, various models may be used; classification, regression, etc. The training phase is an iterative process wherein the hyperparameters are tuned, a best model is selected, and the model has generalized well to new, unseen data. Model training is one of the most resource-intensive stages and requires both computational power and domain expertise.

❖ LSTM Model Training

Long Short-Term Memory (LSTM) is a type of specialized recurrent neural network that is designed to capture sequential dependencies in time series data. In this step, the LSTM model uses historical data to predict future values or to classify sequences based on observed patterns from the past. Training occurs when the model adjusts its weights and biases using gradient descent algorithms in order to minimize its prediction errors. LSTM models are particularly apt for sequences which have long-range dependencies in data, making them a natural choice for time series forecasting or in sequence modeling. Training is a very computationally expensive step and must be fine-tuned to work the best.

❖ GAN Model Training

This step makes it a self-play for training two networks. The generator learns to produce seemingly real data and the discriminator knows how to classify real and generated data. In time, the networks improve since the generator generates better data that even the discriminator cannot distinguish from the real data. GANs are commonly used for tasks like image generation, data augmentation, and anomaly detection. The two models have to be properly fine-tuned and balanced for a successful GAN training process.

❖ Prediction Generation

After training the models, generating predictions is the next step. This includes running the trained LSTM and GAN models on the test data in order to generate future predictions or synthetic data based on learned patterns. For the LSTM, the predictions may be future values in a time series, whereas GANs might create new samples or augment the existing dataset. The model's capacity for generalization—that is, its ability to forecast unknown variables with a tolerable degree of accuracy—is crucial to prediction generation. The prediction phase is also an important criterion in deciding how good the models are in real-world situations.

❖ Run LSTM Model

At this point, the LSTM, which has been trained, is passed for generating predictions over the test data. The performance of the LSTM model is evaluated based on how accurately it makes future event or value predictions using the memory to grasp past dependencies. This step can be real-time prediction or batch prediction depending on the requirement of the project. Running the LSTM model allows one to apply the recognized patterns in forecasting outcomes and, hence, attains a better view into what the future behavior of the data will be.

❖ Run GAN Model

After the training process, the GAN model is run for synthetic data generation or for dataset augmentation. Here, the generator network will start creating some new data samples with the discriminator evaluating the genuineness of such samples. This helps create high-quality synthetic data, which can be applied in applications where real data may appear sparse or costly. Running the GAN model is critical in cases of building or simulating data closely related to the real world, such as producing realistic images, texts, or any complex form of data. Thus, having critical performance at the level of the GAN model determines whether this kind of model will be successful in producing convincing and useful data.

❖ Evaluation

Evaluation is the comparison of the performance of a model to criteria set using different metrics. In the evaluation step, the LSTM model and the GAN model are tested with the actual outcome or real data to check how correct they are and how reliable they are to get the perfect outcome in all situations. The metrics for evaluation may vary based on the task but may involve commonly used ones like the RMSE, MAE, accuracy, or F1 score. One evaluation period is to determine how well the model generalizes to new, unseen data and what improvements, if any, are therefore needed on either architecture of the model or in the training process.

❖ Calculate Metrics (RMSE/MAE)

Specific metrics for quantitative evaluation of model performance, including RMSE (Root Mean Square Error) and MAE (Mean Absolute Error), are calculated in order to determine the model's effectiveness. The most often used metric for regression tasks is RMSE, which measures the average size of the mistakes generated and concentrates on greater errors because it includes the squared term. On the other hand, In order to clearly determine the model's level of accuracy, MAE presents the average absolute difference between projected and actual values. These now become measures of importance to compare the performance of different models with the intent of ensuring the model meets threshold levels of performance.

❖ Compare Predictions / Combine Results

Now, comparisons of various models constructed using both LSTM and GAN will be made in order to predict which model does better. Moreover, if both the models are generating complementary information, the resultant improvements based on the overall prediction accuracy can be computed by combining the results. Further, comparisons can be done by analyzing errors in prediction and also visualizing predicted versus actual values in order to compute aggregated metrics. The results of several models combined are usually something better: stronger, more reliable predictions since they benefit from the strengths of each model.

❖ Visualization

Prediction and evaluation results are natively rendered as plots directly using graph libraries. With the help of graphs, charts, or even tables, users can easily understand the performance of models. Visualization is important in making tough insights accessible and actionable for stakeholders who can then take decisions with the end output of the model. These graphics enable the showing of real time pertinent information such as key metrics, prediction trends, and others that are most relevant in regards to observing model performance and results.

Methods:

➢ LSTM (Long short-term memory):

It is a particular architecture in the family of RNNs first proposed by Hochreiter and Schmidhuber 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. Some of the key components of an LSTM network are the input, output, and forget gates. This design was particularly tailored to deal with the vanishing gradient problem generally encountered in the training of conventional RNNs. LSTM is also a type of memory cell which can selectively add or subtract information from the cell state. The intrinsic design of LSTM thus puts to rest both the problems of exploding and vanishing gradients that afflict standard RNNs. Today, LSTM has proven to be a very potent tool in those applications which involve time-series data, like classification and prediction.

- **GRU**

The GRU is a flavor of RNN using gating schemes to steer the flow of information through the cells. Indeed, the GRU was invented by Kyunghyun Cho et al. in 2014 as a variant of LSTM, containing two gates: update gate and reset gate, responsible for defining what to be kept and what is to be discarded. 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 misses one of the gates and does not use the cell state and stores both long-term and short-term memory in the hidden state.

- **GAN**

A GAN is essentially a minimax optimization problem, in terms of zero-sum, non-cooperative game theory. Classically, a GAN would consist of two main parts-the generator and the discriminator. Its goal is to fool the discriminator by generating examples that look similar to real data, yet the discriminator needs to determine whether or not the input example is real or fake, or generated. Of late, this model has attracted great interest in the Deep Learning community. 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.

In this experiment, train the discriminator to maximize its objective function: the probability of correctly labeling samples. Let's define the objective function for the discriminator as:

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

Here, x is the input data for the generator, y is the target from the real data set, and G(x_i) is the generated or the fake data produced by the generator.

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 discriminator's loss function, 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 generator's loss function is:

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

In order to attain optimal model performance, the objective is to minimize these loss functions throughout the training phase.

- **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, or WGAN-GP.

WGAN-GP exploits the concept of Wasserstein distance, also referred to as Earth-Mover

Distance (EMD) for better GAN training. The Wasserstein distance can be regarded as the minimum cost to transform one data distribution to another. Mathematically, this is formally defined as the greatest lower bound (infimum) of any transportation plan to transform the real data distribution Pg to the generated one Pr - the cost of the cheapest plan:

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

Define the set of all possible joint distributions between P_r and P_g as $\Pi(P_r,P_g)$, and any imaginable transport plan γ. Using the Kantorovich-Rubinstein duality computation, the computation of Wasserstein distance can be reduced to

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

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

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

The WGAN-GP imposes the Lipschitz constraint using a gradient penalty. Define a differentiable function f to be 1-Lipschitz if its gradient norm everywhere is less than or equal to 1, that is, $\| \nabla f \|_2 \leq 1$. Here, if the gradient norm deviates from the target, 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, WGAN-GP uses a gradient penalty 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	$\frac{1}{m}\sum_{i=1}^{m}\left[D(y^i)-D\left(G(x^i)\right)+\lambda(\lVert\nabla D_{y^i \sim x^i}\rVert_2-1)^2\right]$	$-\frac{1}{m}\sum_{i=1}^{m} D(G(x^i))$

Table 2. The WGAN-GP loss function and Basic GAN comparison

- **The Generator**

We set GRU as a generator in the GAN model due to its stability. Our dataset has a history of the stock price over the past 10 years and comprises 36 features that include 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 make the multi-step ahead prediction, therefore in the generator we have to define the input step and output step, also the input of the generator will be three dimensional data, that is batch size, input-step and features, the output will be batch size and output step. Hence, to build the good-performance generator, three layers of GRU were applied with neuron numbers 1024, 512, and 256 followed by two layers of Dense, and the number of neurons in the latest layer will be equal to the number of steps in our output.

- **The Discriminator**

The discriminator in our model of the GAN is supposed to be the Convolution Neural Network that would be trying to decide whether the input data of the discriminator is real or fake. This discriminator would always receive the input from either the original data or the generated data produced by the generator. It consists of three 1D Convolution layers, which are separately composed of 32, 64, and 128 neurons, while adding three other Dense layers at the end, containing 220, 220, and 1 neuron. The Leaky Rectified Linear Unit (ReLU) has been applied as an activation function amongst all layers except for the output layer set by Sigmoid activation function for GAN and linear activation for WGAN-GP. The Sigmoid function will result in one scalar output, 0 and 1, meaning real or fake.

➢ The Architecture of GAN

Our proposed GAN model is a combination of the two generator and discriminator architectures mentioned previously.

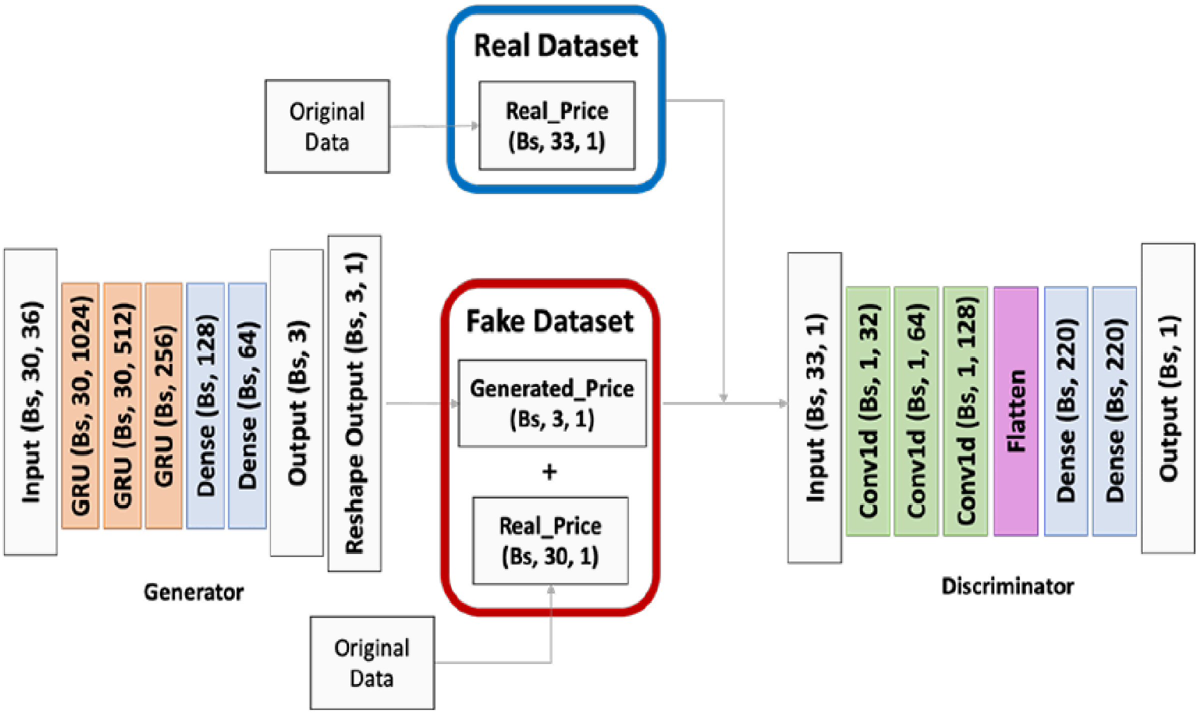

Fig. 3 GAN Structure

We already specified the function in the theoretical background, and we apply cross-entropy in our GAN model structure to compute our loss for both the discriminator and the generator. In particular, the discriminator increases the duration of the data and improves the discriminator's ability to learn the categorisation by combining the generated stock price with the previous stock price of the input stages

Use Case Analysis Document:

USE CASE	Time-Series Stock Price Forecasting using LSTM and GAN
Goal	To predict future trends by employing LSTM and GAN to forecast stock values.
Purpose	By using LSTM for sequence modeling and GAN for producing realistic projected data, the goal is to deliver precise stock price predictions.
Preconditions	1. A dataset containing historical Stock prices. 2. Preprocessing tools for data transformation. 3. Trained LSTM and GAN models.
Success Condition	1. Stock price predictions are accurate within a reasonable error margin. 2. The GAN-generated data closely matches the actual market behavior. 3. Predicted outcomes are trustworthy and useful for making decisions.
Failed Condition	1. Poor model performance or overfitting. 2. Inconsistent or inaccurate predictions. 3. Unforeseen market events not captured by the model.
Primary Actors	1. LSTM Model: Forecasts future stock values by using past data. 2. GAN: Generates synthetic stock data to augment the prediction model.

Secondary Actors	Data Preprocessing Pipeline: Prepares and cleans the data. Stock Market Analysts: Use the predictions to inform investment strategies.	
Trigger	New stock data is fed into the model, initiating the forecasting process.	
Description	**STEP**	**BASIC COURSE OF ACTION**
	1	Data Preprocessing: To prepare it for model training, historical stock data is cleaned, normalized, and formatted.
	2	Model Training (LSTM & GAN): LSTM is trained on the time-series data to learn temporal patterns, whereas GAN creates extra training data to boost model generalization.
	3	Model Evaluation: To guarantee reliable forecasting, the trained models are assessed using metrics such as Mean Absolute Error (MAE) and Root Mean Square Error (RMSE).
	4	Stock Price Prediction: Based on the trends in the current data, the trained LSTM model predicts future stock values.
	5	GAN Data Generation: In order to enhance predictions, the GAN creates synthetic stock price data that closely resembles actual stock price trends.
	6	Visualization and Reporting: Forecasted results are visualized on a dashboard and used for decision-making.

CHAPTER 4

RESULTS AND DISCUSSION

- **Training of our model**

  This paper's goal is to forecast the closing price of the stock for the upcoming three days using data from the previous thirty days. This project will input 36 features that could affect the price in addition to the historical closing price in order to train the forecasting model. The dataset is divided into 70% (1726 data) and 30% (739 data) training and testing sets during the training phase. As part of the testing process, we will then perform two distinct parts: prediction with and without an unexpected occurrence. COVID-19 for 2020 was the unforeseen occurrence for this project.

- **Experimental and results**

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

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

The *N* is the number of the data points, x_i is actual stock price, and $\hat{x}$ is predicted stock price, In order to assess the models we developed for this project, we assessed each model's RMSE on testing data (with and without 2020).

- **LSTM**

  We set the first layer of our LSTM model to be a bidirectional LSTM. The Adam algorithm, which has a learning rate of 0.001, is the optimiser that we are utilizing for our models in this work. We train 50 epochs on this stock price dataset with a batch size of 64. The entire dataset, which serves as the input for the GAN model in the baseline model, comprises 36 correlated features and ten years' worth of historical data. The testing dataset began on 07/21/2017 after the data was divided into train and test sets.

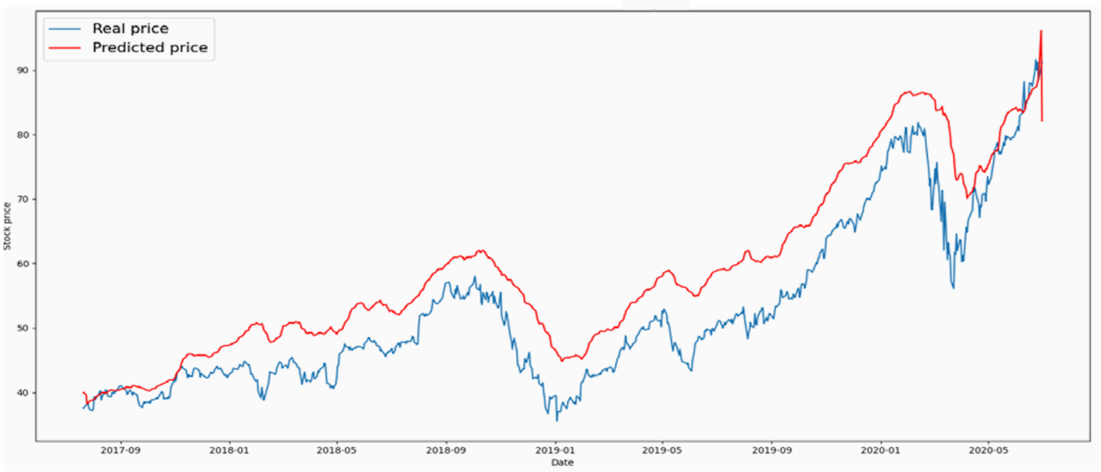

Fig 4 RMSE Comparison with and without 2020 Data

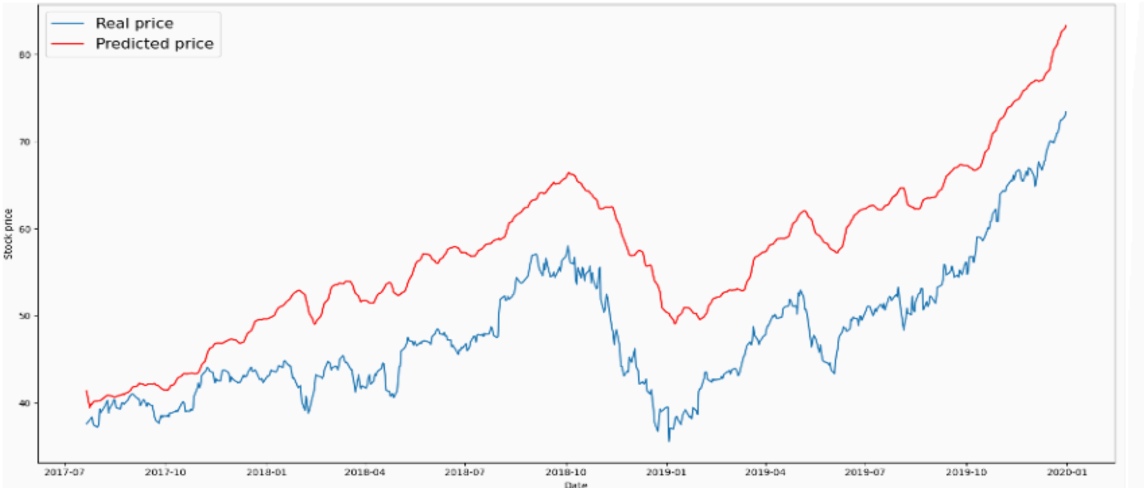

Fig 5. GRU vs LSTM Performance Comparison for 2020

The results of LSTM, which includes forecasting 2020, is shown in Fig 4. The true stock price is shown by the blue line, and the forecasted stock price is shown by the red line. The RMSE is 6.60. In the period up to the last day of May 2020, the projected stock price appears to be slightly higher than the actual stock price. The projection then approaches the actual stock price, and Fig. 5 shows the result of the scenario in which 2020 is not included. This raises the RMSE to 9.42, which is significantly higher than the outcome that includes 2020.

- **GRU**

The second fundamental model in this study is the GRU model. constructing this model. We employed two GRU layers in this model, and we trained it for 50 epochs using the Adam algorithm as the optimiser. The batch size is 128 and the learning rate is 0.0001

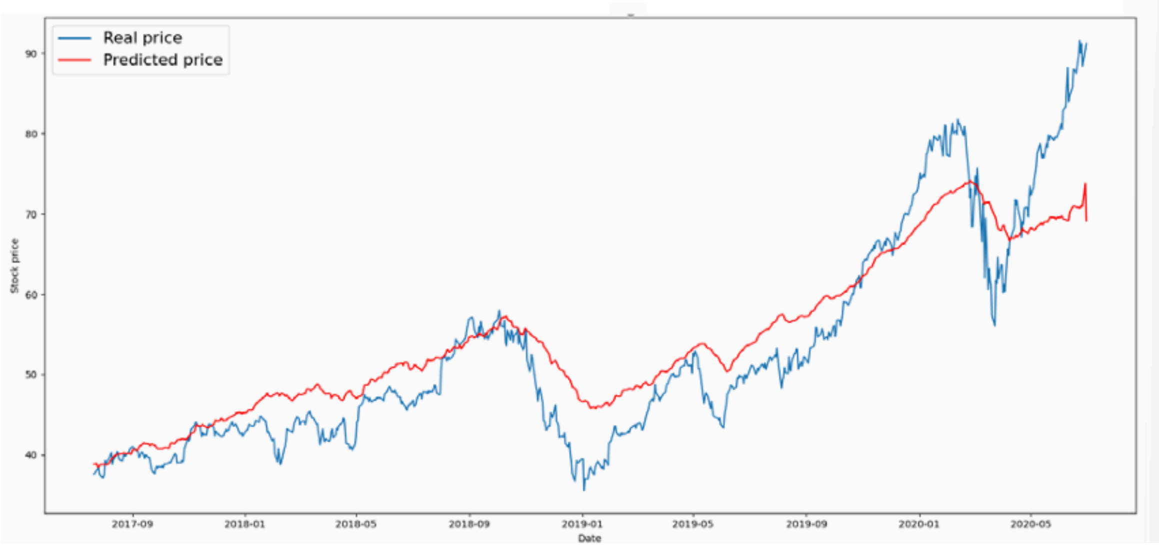

Fig 6. GRU Performance with and without 2020 Data

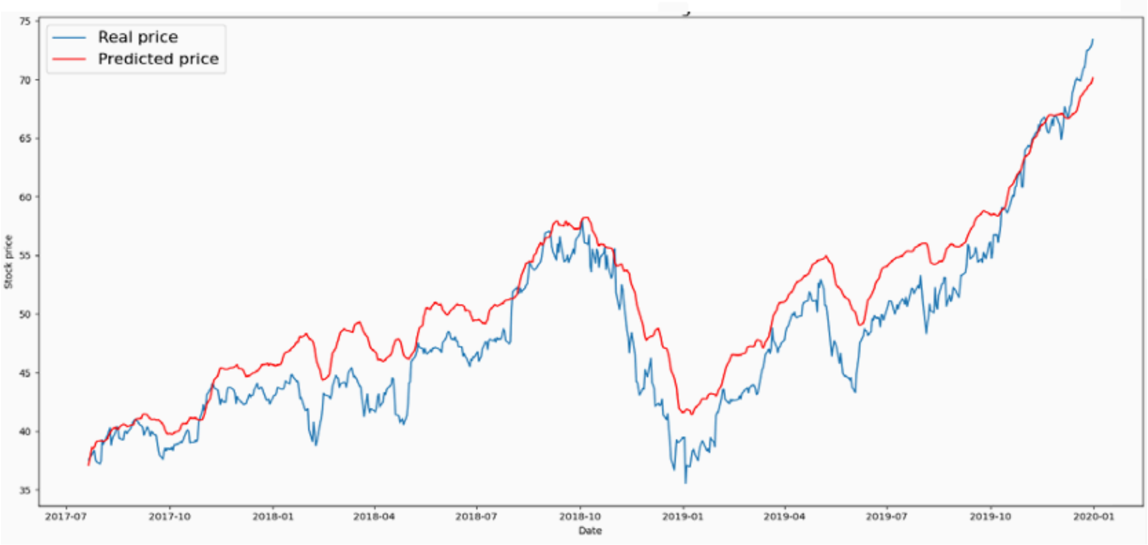

Fig.7 GRU Performance Excluding 2020 Data

Fig 6. GRU with inclusion of 2020. RMSE is 5.33. As well, As can be seen from the figure, the GRU model performs better than the LSTM mode prior to May 2020. This figure

shows that the projection collapsed after May 2020. Figure 7. GRU excludes 2020; RMSE is 4.08. When the GRU model makes predictions without anticipating anomalous occurrences, it performs significantly better.

- **Basic GAN**

The methodology section of this paper proposes the structure of the GAN model. The Adam algorithm, which has a learning rate of 0.00016, is the optimiser utilized in this model for our models in this paper. We train the model on this dataset for 165 epochs with a batch size of 128.

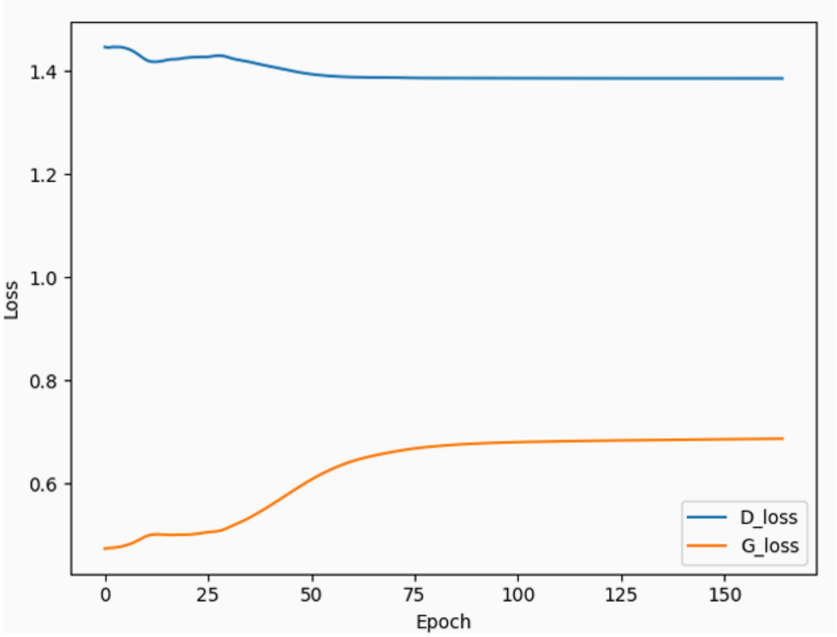

Fig.8 Loss Plot of Basic GAN Model (Discriminator vs. Generator)

Fig 8: Loss plot of the basic GAN model, where the loss curve of the discriminator is represented by the blue line and the orange line is representing the loss curve of the generator.

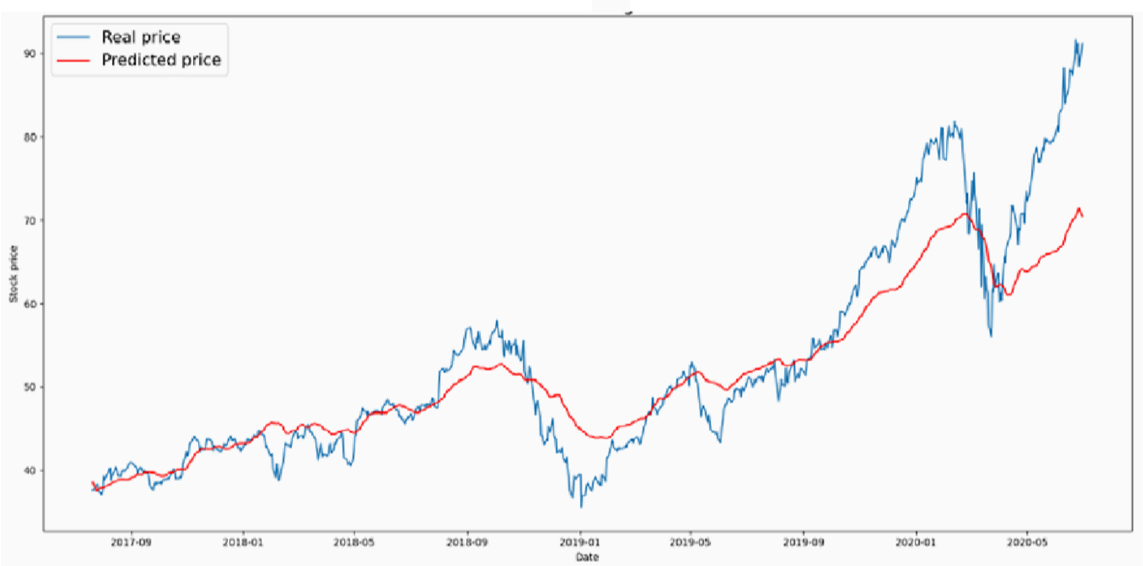

Fig. 9 Predicted Results from Basic GAN Model with RMSE of 5.36

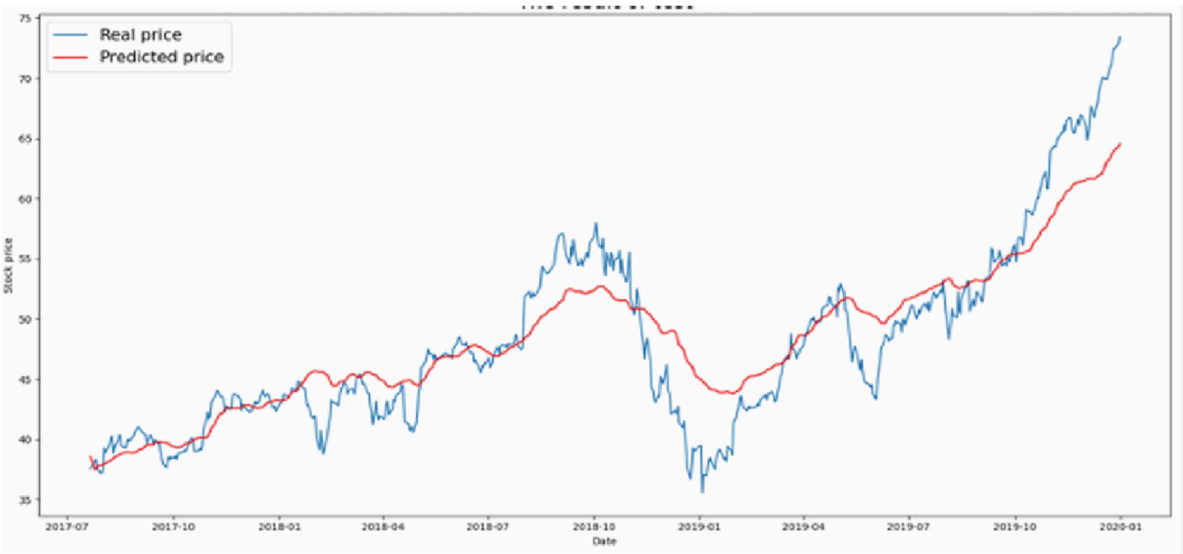

Fig.10 Baseline GAN Model Results Excluding 2020 Forecast with RMSE of 3.09

Figure 9 shows the outcome of the basic GAN model, with an RMSE of 5.36. This indicates that there was a significant discrepancy between the actual price and the price prediction for 2020, as well as a sudden spike that may have been brought on by the unforeseen COVID-19 event. The RMSE drops to 3.09 when the basic GAN model excludes 2020 predictions, as shown in Fig. 10. This indicates that the basic GAN for forecasting outperforms both basic models when the unforeseen event is removed.

- **WGAN-GP**

In the technique section, we suggested the WGAN-GP model's structure. With a learning rate of 0.0001, the optimiser in this model is likewise an Adam algorithm. We use this dataset to train the model for 100 epochs, using a batch size of 128. Additionally, we train the generator three times and the discriminator once.

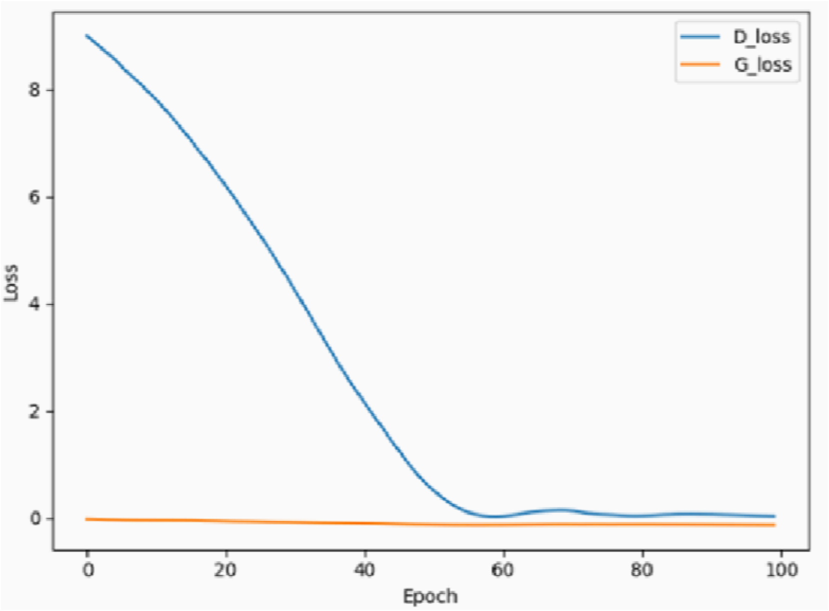

Fig. 11 WGAN-GP Model Loss Plot with Zero Discriminator Loss

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.

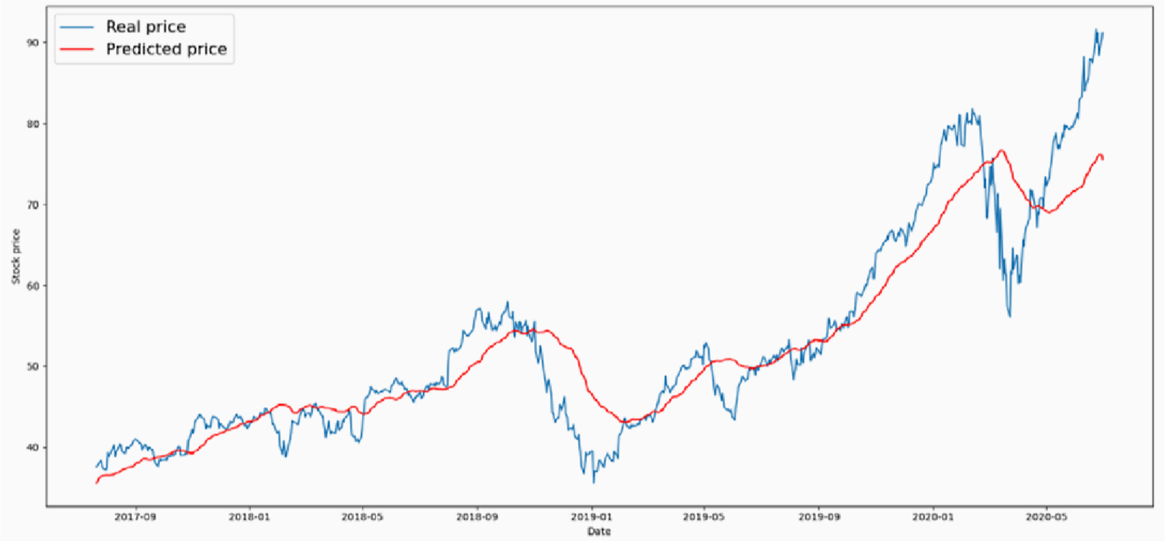

Fig.12 WGAN-GP Model RMSE with Impact of COVID-19 on Predictions

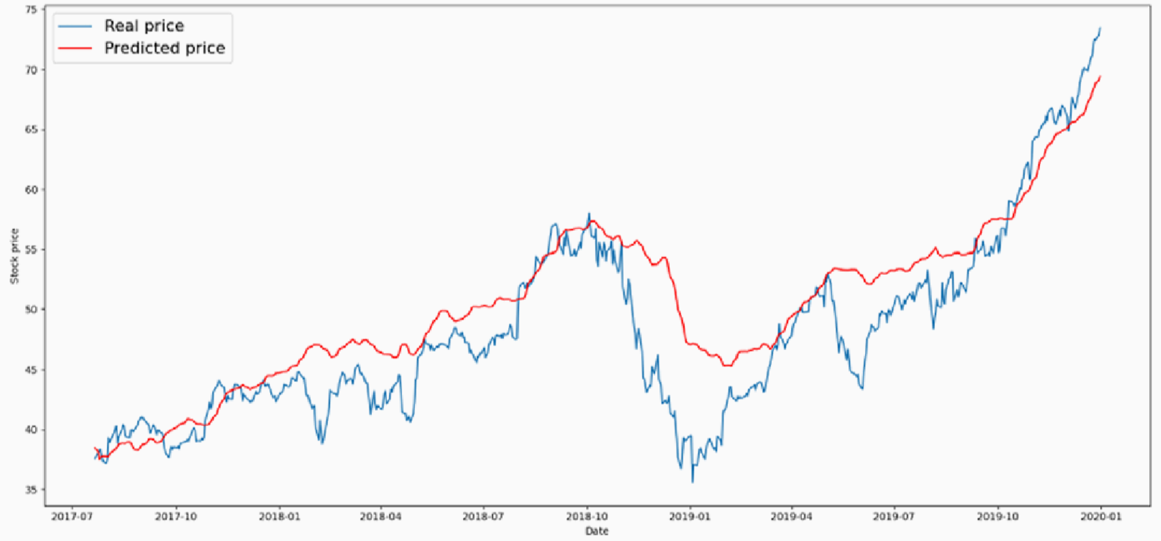

Fig. 13 RMSE Comparison: WGAN-GP vs Basic GAN

Fig 12 is the prediction that comes out from the WGAN-GP model, and the RMSE is 4.77 which is the best one across all models. As can be observed at the basic GAN in 2020, the prediction starts to fall very wide apart from the real price due to the unexpected event COVID-19 occurs. Finally, we remove the test data in 2020, as shown in Fig 13. The RMSE of the forecasting goes to 3.88; it performs worse than the basic GAN model.

➢ 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

Table 3. The evaluation on different models

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.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. 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.

- ➢ **Challenges**

The implementation of the proposed technique poses several challenges, few most important ones are as follows:

- **Model Complexity and Hyperparameter Tuning:** The GAN models, especially involving RNNs like GRUs, are not exactly easy to tune. With instability, hyperparameter selection was a bit challenging with the forcing implications on accurate predictions using untuned parameters.
- **Surprising events and deviation from the predictions:** In the period of COVID-19. Large price deviations from predictions were caused by unprecedented volatility levels, making it harder for a traditional model to be able to deal with unless there were special adjustments for such events.
- **Training Stability and Gradient Issues:** GAN model training was unstable, and lots of work needed to be done using gradient penalties for enforcing the Lipschitz constraint, especially in the WGAN-GP variant. Slow convergence, especially balancing loss for the generator with loss for the discriminator, requires more complexity
- **Limitations in Time Series Prediction with GANs:** GANs are not as widely adopted for time series data, therefore, it was really challenging to adapt them for predicting stock prices especially since the generated output must be realistic at multiple time steps.
- **Complex Data Preprocessing and Feature Engineering:** Inputs are multi-dimensional; in most ingenuity, there is a way technical indicators, financial indices, and even sentiment analysis structure and preprocess into feeding appropriately into the GAN model

CHAPTER 5
CONCLUSION AND FUTURE SCOPE

In the proposed system, we have designed a GAN-based model for the stock price forecasting problem using GRU as the generator and CNN as the discriminator. GRU is an architecture which supports sequential learning well, while CNN is robust in the pattern recognition and enhances the precision of the prediction. From the experiment, we can observe that the GAN-based models like a simple GAN and WGAN-GP are well outperformed the traditional approaches such as LSTM and GRU. It is demonstrated that, in fact, WGAN-GP excels greatly in volatile market conditions such as in the case of the COVID-19 pandemic since it depicts enhanced robustness and stability. In contrast, basic GAN is performing exceptionally well on stable market conditions and attains a superior accuracy over others.

Although these are encouraging results, hyperparameter tuning is a very challenging task associated with the stability and performance of the model. Most models that fail to fine-tune properly may result in unstable models, improper convergence, or suboptimal performance. GAN architecture is complex with interactive parameters, and the manual optimization involved in this process is time-consuming and computationally demanding. Here, advanced hyperparameter optimization techniques come into play. Reinforcement learning methods, such as Rainbow, the Q-learning method, and PPO, would be explored in this context to automatically and optimize the process of tuning so these models run more stably and effectively.

This research would emphasize how GANs, a special type of deep learning models, transform the time-series forecasting process and handle complex dynamic financial data that is prone to errors. A well-designed GAN would help financial forecasting capture intricate patterns and respond to crisis situations; hence, development should be further done on adaptive strategies of learning and self-tuning mechanisms for being improved. Reliability and accuracy in GAN models are expected to be even much better due to automation of optimization combined with gains in reinforcement learning, which promises much better performers in highly dynamic, unpredictable financial environments.

Future work regarding this project would be the sophisticated hyperparameter tuning techniques that could eventually improve stability and predictability of a GAN-based model in the task of stock price forecasting. This would include the Rainbow or Proximal Policy Optimization algorithms which could eventually automate and enhance the adjustment process itself. Adaptive architectures for GAN would increase the robustness of the model when volatility arises from changing parameters according to market behavior.

Improvement in the underlying feature set by incorporating real-time data sources like social media sentiment and macroeconomic indicators could add predictive power. Handling unexpected market-related noise reduction and anomaly detection events could be further improved. Better scalability for practical deployment would be ensured through enhanced computational strategies and distributed training approaches. Novel hybrid models involving GANs and other deep learning approaches could unlock new avenues toward very reliable and more efficient predictions of stock market indicators

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.