



DEEP LEARNING TECHNIQUES GAN AND LSTM FOR STOCK MARKET CLOSING PRICE PREDICTION

Pappula Praveen^{*1}, Anjaneyulu Gudla²

^{1, 2, 3}Department of Computer Science and Engineering, School of Computer Science and Artificial Intelligence, SR University, Warangal, Telangana, India.

¹<http://orcid.org/0000-0002-2967-8435>, ²<http://orcid.org/0009-0008-8916-4031>

Email: *prawin1731@gmail.com, anjugudla5@gmail.com

ARTICLE INFO

Article History

Received: March 3, 2026

Reviewed: June 12, 2026

Accepted: July 12, 2026

Published: July 31, 2026

Keywords:

Stock Price

GAN

LSTM

Deep Learning

ARIMA.

ABSTRACT

Accurate stock price prediction is vital for investors and analysts to make informed decisions and mitigate risks. Traditional methods like ARIMA struggle with the complex, non-linear patterns of stock market data, leading to less precise forecasts. This study addresses the gap by leveraging advanced deep learning methods, specifically Generative Adversarial Networks (GANs) and Long Short-Term Memory (LSTM) networks, for stock price forecasting. Using historical stock data and 32 influencing features, we trained GANs and LSTMs, with Root Mean Square Error (RMSE) as the evaluation metric. The data was split into 80% training and 20% testing. Our results show that the GAN model achieved a lower RMSE of 5.36, outperforming the LSTM model (RMSE of 6.6) and traditional ARIMA models, demonstrating GAN's superior ability to capture complex patterns. While computational challenges exist, GAN's accuracy underscores its potential for stock price prediction. Future research could explore hybrid models and sentiment analysis to further improve prediction accuracy.



Copyright ©2026 by authors and Galileo Institute of Technology and Education of the Amazon (ITEGAM). This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

I. INTRODUCTION

Forecasting prices of stock market is a complex and demanding task because of the highly dynamic as well as non-linear characteristics of financial markets. Precise forecasts can provide significant benefits for investors as well as traders which enables them to make informed decisions, handle risks efficiently along maximize returns. Traditional methods of the prediction of stock price like ARIMA models often fail to capture accurately the intricate along with the unpredictable patterns inherent in market data. These limitations have promoted the investigation of advanced methods. Deep Learning approaches have gained popularity for their capacity to model complex patterns as well as the interconnections in time-series data. Methods like GANs as well as LSTM networks handle large volumes of data as well as uncover hidden connections within the data. They also provide more precise as well as reliable predictions comparing conventional data. GANs which can generate synthetic data and LSTMs are known for their proficiency in handling sequential data which provides promising results for the challenges presented by the stock market's volatility as well as complexity. Researchers want to improve the precision as well as robustness of stock price predictions by using these advanced DL techniques which leads to better investing strategies along with financial decision-making. Many researchers have explored deep learning (DL) models for stock price prediction. By [1] introduced the S_I_LSTM technique, combining traditional and non-traditional data like stock posts and financial news, enhancing accuracy in the Shanghai A-share market. In turn [2] evaluated DL techniques using Google's daily closing price, finding GANs superior to LSTM and GRU. According to [3] proposed a GAN with an economics-driven loss function for predicting financial time series. In turn [4] tested multiple DL models, highlighting GRU and addressing overfitting with GANs. By [5] improved accuracy by combining empirical mode decomposition with LSTM and GAN architectures. According to [6] developed the GAF-EWGAN method, integrating enhanced WGANs with Gramian Angular Fields. By [7] introduced GAN-TrellisNet, which combined TrellisNet and CNN for better stock prediction accuracy. In turn [8] created a GAN-based model with piecewise linear representation to forecast stock trading actions. By [9] used BERT and GAN for sentiment analysis to predict Apple Inc.'s stock, achieving high accuracy. According to [10] combined neural networks and GANs for S&P 500 predictions, achieving superior results with the G-LSTM+D-LSTM technique.

By [11] developed TSGAN, integrating ACNN, LSTM, ARIMA, and CNN for accurate stock predictions. In turn [12] combined Phase-space Reconstruction and GAN for effective forecasting. According to [13] introduced a BiLSTM and WGAN approach to predict stock prices for newly listed companies with limited data. By [14] reviewed various stock prediction techniques, noting the rise of DL methods from 2020 to 2023. In turn [15] highlighted the shift from traditional methods to advanced DL techniques, including Transformers, GANs, and GNNs. Financial news and social media have been shown to enhance prediction accuracy [16], while GANs and LSTMs often outperform traditional methods in capturing the non-linear dynamics of financial data [17]. Addressing overfitting and improving model robustness are common themes, with various studies exploring artificial data generation and hybrid approaches [18]. However, there is still a research gap in the comparative analysis of GANs as well as LSTMs, especially for the prediction of the stock market. Several research focuses on the utilization of a single DL technique with less exploration of the synergistic potential of combining GANs as well as LSTMs. Previous researches often suffer limitations like overfitting, lack of integration of diverse data sources as well as a limited focus on certain markets or stocks. The motivation of this study is to utilize advanced DL methods especially LSTM networks as well as GANs to enhance the predictions of the stock market. Traditional techniques often face difficulties when dealing with complex unpredictable characteristics of financial data. The main motivation of this research is the potential of utilizing advanced methods to combine various data sources like historical prices, technical indicators as well as sentiment analysis from news to create a more holistic as well as accurate prediction model. The objective is to demonstrate the practical applications as well as the benefits of these complex modeling methods by focusing on Apple Inc., a stock that is highly influential as well as volatile. The objective of this study is to enhance the accuracy of stock price prediction by utilizing advanced structures of LSTM as well as GAN. This study aims to contribute to expanding the literature on the utilization of DL methods in financial market analysis. This research focuses on the utilization of advanced DL techniques like LSTM networks and GANs to forecast stock prices which especially examines the stock price of Apple Inc.

II. MATERIAL AND METHODS

II.1 MATERIALS AND METHODS

Materials:

The first step in building a technique involves obtaining a dataset that includes the essential features required for training the model. In DL the dataset contains the target values that are used during the training process. In this particular scenario, the dataset comprises continuous feature extractions that are presented in a time series format. In this study, the daily closing price of the APPLE stock data is used. The series covers 497 trade days from 2nd January 2020 to 19th July 2024. Overall, 497 observations are present in the data set. While there are around 1187 trading days in this timeframe, the 497 trade days considered are based on specific criteria relevant to the study, such as days with significant market movements or availability of complete data. Figure 1 provides a comprehensive analysis of AAPL stock performance showing the closing prices, moving averages, daily returns, and their distribution. The detailed analysis helps in understanding the stock's behavior over the selected period.

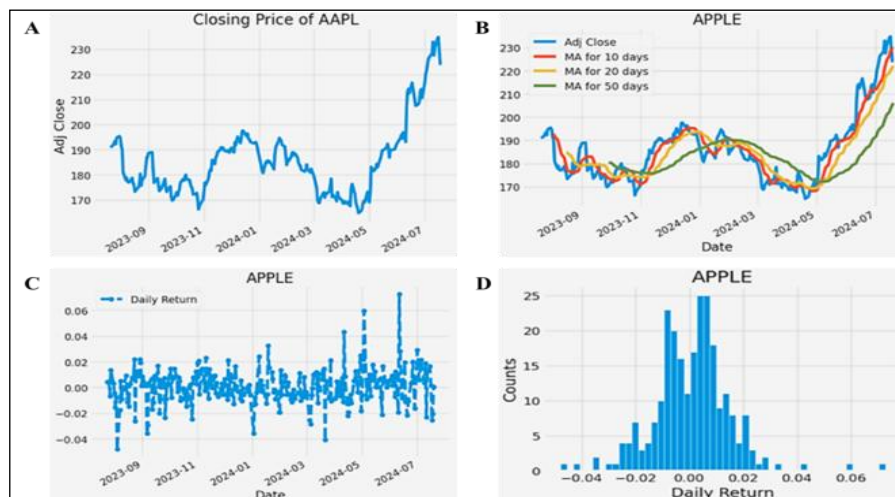


Figure 1: Analysis of AAPL Stock Performance. (A) Closing prices over the 497 trade days, illustrate the overall trend and key fluctuations. (B) Moving averages for 10 days, 20 days, and 50 days, highlighting long-term and short-term trends. (C) Daily returns, showing the day-to-day changes in stock price. (D) Distribution of daily returns, providing insights into the frequency and magnitude of daily changes. Graphs were generated using Python.

Source: Authors, (2026).

II.1.1 Long Short-Term Memory (LSTM):

LSTM is a type of recurrent neural network (RNN) designed by Hochreiter and Schmidhuber in 1997 to handle sequential data more effectively than standard feed-forward networks. To control information flow, it has an output gate, an input gate and a forget gate in addition to feedback loops. The main novelty of LSTM is that it can retain or discard data according to its current cell state, which helps to alleviate the vanishing gradient issue that standard RNNs frequently encounter while training. This unique design addresses issues related to vanishing and exploding gradients, making LSTM a robust tool for tasks involving time series data, such as sequence prediction and classification.

II.1.2 Generative Adversarial Network (GAN):

GANs are structured as a zero-sum game, where two components—a generator and a discriminator—compete against each other. The generator is crafted using GRU layers and is designed to produce data that appears convincingly real. Meanwhile, The discriminator's job is to differentiate between generated and genuine data. The loss function for the discriminator is:

$$V_D = \frac{1}{m} \sum_{i=1}^m \log D(y^i) + \frac{1}{m} \sum_{i=1}^m (1 - \log D(G(x^i)))$$

The loss function for the generator is:

$$V_G = \frac{1}{m} \sum_{i=1}^m (1 - \log D(G(x^i)))$$

GAN framework utilizes cross-entropy loss to align the generated data distribution with the real data distribution, effectively minimizing the Kullback-Leibler-Jensen-Shannon (KL-JS) divergence.

II.2 FEATURE EXTRACTION FROM DATA:

Technical Indicators:

In this research, we calculated several key technical indicators used by investors: exponential moving averages, 7-day and 21-day moving averages, Bollinger Bands, momentum, and the Moving Average Convergence Divergence (MACD). The formulas and their significance are:

7-day and 21-day Moving Average (MA):

$$MA_n = \frac{1}{n} \sum_{i=0}^{n-1} P_{t-i}$$

Moving Average smooth out price data to identify the trends by averaging past prices over a significant period. The 7-day MA captures short-term trends while the 21-day MA provides insights into medium-term trends.

Exponential Moving Average (EMA):

$$EMA_t = (P_t \times \frac{2}{n+1}) + (EMA_{t-1} \times (1 - \frac{2}{n+1}))$$

These indicators are particularly sensitive to recent price changes, allowing for a faster reaction to new market data. It helps in identifying the direction of the current trend along with potential reversals.

Momentum:

$$P_t - P_{t-n}$$

It measures the rate of price change. The rising trend is known as positive momentum and a declining trend is known as negative momentum. This helps traders gauge the strength of a price movement.

Bollinger Bands:

Bollinger Bands consist of two outside bands that are positioned ± 2 standard deviations apart from the center and a middle band that is made up of the 20-day moving average. This tool is used to gauge market volatility and outlines a range within which the price is expected to fluctuate. Prices may be overbought when they approach the top band, while they may be oversold when they approach the lower band.

Moving Average Convergence Divergence (MACD):

$$\text{Signal Line: } 9\text{-day EMA of MACD}$$

Market momentum is reflected by the MACD, which shows a bullish trend when its value is positive and a bearish trend when its value is negative. The MACD line indicates possible buy or sell signals when it crosses above or below the signal line.

News Sentiment Analysis:

Stock price changes may be impacted by news. We gathered daily news stories on Apple Inc. and assessed each article's mood using the FinBert model. With a sentiment score ranging from -1 to 1, our analysis classified the news as good, neutral, or negative. These sentiment scores were then integrated into the model as additional features, allowing the model to incorporate the impact of market sentiment on stock price predictions.

Fourier Transforms:

The study employed Fourier transforms to identify long-term and short-term trends in Apple stock. By decomposing the data into sine wave components, Fourier transforms help in reconstructing the original signal, enabling the model to discern patterns and trends more accurately for predictive purposes.

II.3 DATA STRUCTURE:

This study prepared the dataset for supervised learning by dividing it with a rolling window equal to 1. The original dataset is two-dimensional and reshaped the data to three dimensions according to the timesteps as shown in Figure 2. The specific steps taken to prepare the data for the many-to-many model:

Rolling Window Approach: The rolling window technique involves creating overlapping subsets of the data where each window shifts by one-time step. For a rolling window of size 1, each subset contains the data point at time t and its previous data points.

Original Data: the dataset starts as a two-dimensional array with rows representing time steps and columns representing features.

Reshaping Process: For each rolling window, the data is split into input sequences (features at time t) and output sequences (features at time $t+1$ to $t+1$). This process transforms the data into three dimensions: [samples, timesteps, and features]. Each sample corresponds to a rolling window, each timestep within a sample corresponds to a specific time point in that window, and each feature corresponds to the various data attributes at that time point

Many-to-Many Model Preparation: Input Data: For the LSTM and GAN models, the input data is structured as [number of samples, number of timesteps, number of features].

Output Data: The output for each input sequence is also a sequence, allowing the model to predict multiple future time steps (many-to-many). Example: If the rolling window size is 5, each input sequence will have 5 timesteps, and the corresponding output sequence will also have 5 timesteps. Figure 2 illustrates how the original dataset is reshaped into a three-dimensional structure suitable for the many-to-many model.

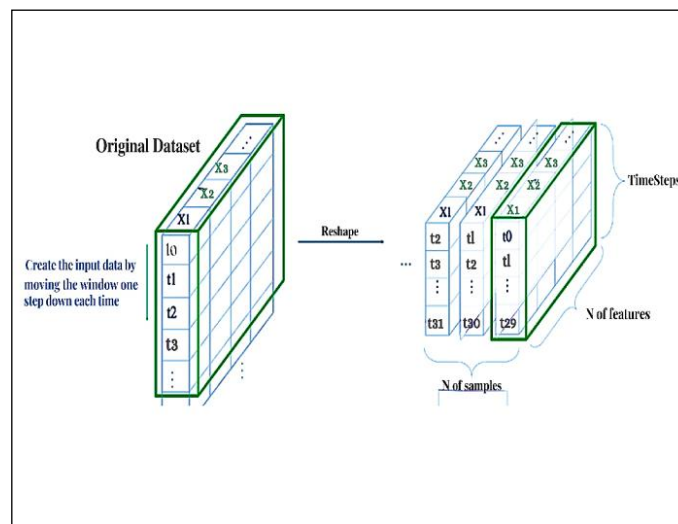


Figure 2: Reshaping the Original Dataset for Input Data Preparation.

Source: Authors, (2026).

This research developed a model that uses a many-to-many approach, utilizing a 30-day time step to forecast stock prices for an output step of 3, as depicted in Figure 3.

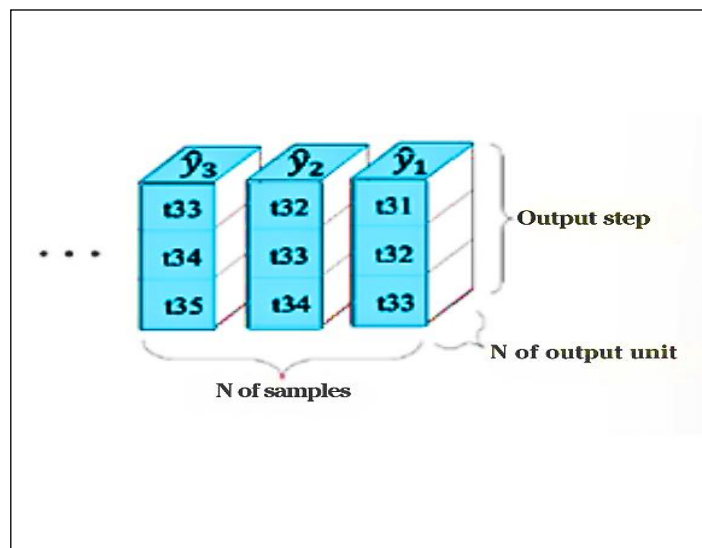


Figure 3: Many-to-Many Models with Timestep of 30 and an Output Step of 3.

Source: Authors, (2026).

II.4 METHODOLOGY

Generator:

The study used a GRU as the generator component in the GAN model, chosen for its stability and effectiveness in handling sequential data. The dataset spans a decade of historical stock prices and includes 32 distinct features such as Open, High, Low, Close, Volume, and various global indices like NASDAQ, NYSE, S&P 500, and others.

Additional metrics such as Crude Oil prices, Gold prices, VIX, USD index, and technical indicators (e.g., moving averages, MACD, Bollinger Bands) are also part of the dataset, alongside news sentiment data. We perform multi-step ahead prediction. The generator model receives input data in three dimensions, consisting of batch size, input steps, and features, and it produces output based on the batch size and output steps.

Generator Architecture:

GRU Layer 1:

- Neurons: 1024
- Activation Function: ReLU
- Dropout Rate: 0.2
- Regularization: L2 regularization with a coefficient of 0.01

GRU Layer 2:

- Neurons: 512
- Activation Function: ReLU
- Dropout Rate: 0.2
- Regularization: L2 regularization with a coefficient of 0.01

GRU Layer 3:

- Neurons: 256
- Activation Function: ReLU
- Dropout Rate: 0.2
- Regularization: L2 regularization with a coefficient of 0.01

Dense Layer 1:

- Neurons: 128
- Activation Function: ReLU

Dense Layer 2:

- Neurons: Number matching the output step to be predicted (e.g., if predicting 10 steps ahead, this layer has 10 neurons)
- Activation Function: Linear

Input and Output Dimensions:

- Input Dimensions: [batch size, input-step, number of features (32)], where batch size is the number of samples in a batch; input-step = number of timesteps in each input sequence.
- Output Dimensions: [batch size, output-step], where the batch size is the same as the input batch size, and the output-step is the number of future time steps to predict.

This detailed architecture ensures that the generator can effectively capture temporal dependencies and generate accurate multi-step ahead predictions.

Discriminator:

The discriminator in this GAN model is a Convolutional Neural Network (CNN) designed to differentiate between real and generated data, as depicted in Figure 4. It accepts both genuine and synthetic data as input. The architecture features three one-dimensional convolutional layers with 32, 64, and 128 neurons, followed by three dense layers with 220, 220, and 1 neuron, respectively. With the exception of the last layer, which employs a linear activation for WGAN-GP or a sigmoid activation for GANs, all of these layers utilize leaky ReLU as their activation function.

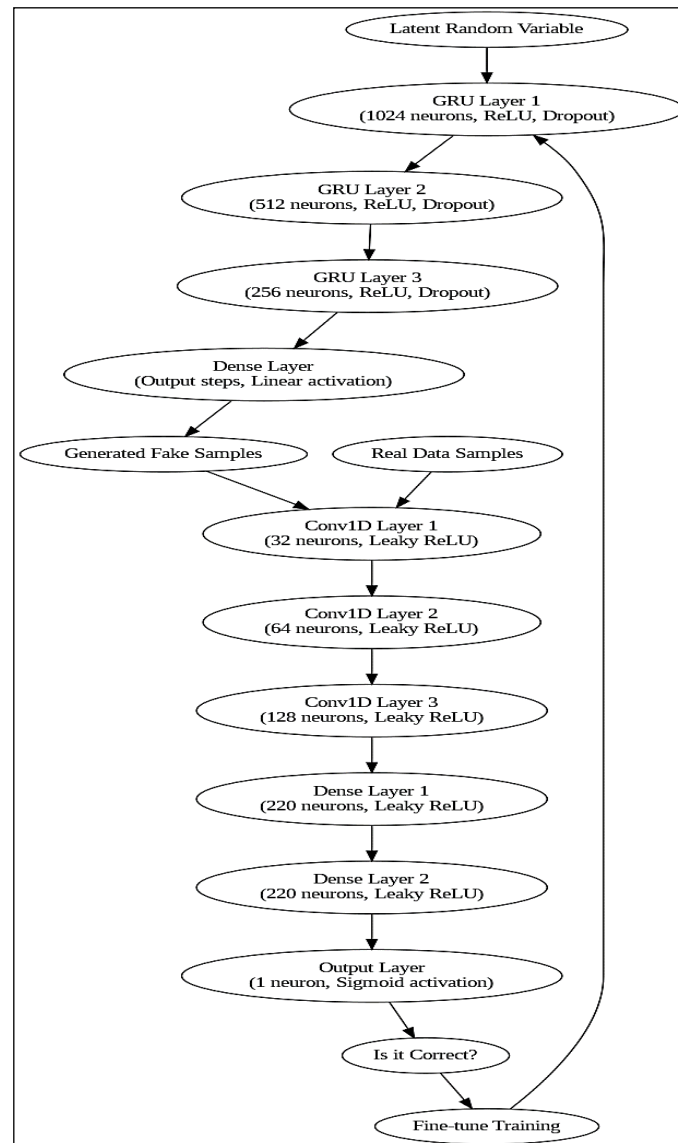


Figure 4: Flowchart of GAN Model.
Source: Authors, (2026).

Activation Functions:

Leaky ReLU:

This function successfully mitigates the dying ReLU issue, where neurons might stop learning, by allowing for a little positive slope (often set to 0.01) while the neuron is not active. By preventing neurons from staying dormant, this method encourages continuous learning. All of the discriminator's layers—aside from the output layer—are subject to Leaky ReLU.

Sigmoid:

The Sigmoid function squashes outputs to the range [0, 1], suitable for binary classification by representing probabilities. In GANs, the Sigmoid function in the discriminator's output layer indicates the likelihood that input data is real or fake. This combination of Leaky ReLU and Sigmoid activations enables the discriminator to effectively differentiate between genuine and fake data, enhancing the GAN model's performance.

Architecture of GAN:

Combining the generator and discriminator structures, we define our proposed GAN model. The model uses cross-entropy to calculate the loss for both the generator and discriminator. In the discriminator, as shown in Figure5 this study combines the generated stock price with the historical stock price of input steps, enhancing data length and increasing accuracy in classification.

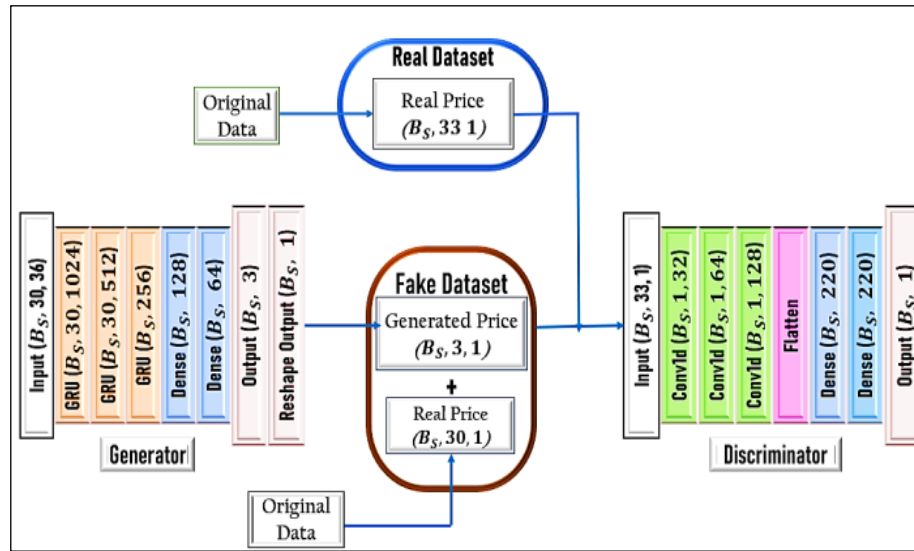


Figure 5: The architecture of the GAN model.

Source: Authors, (2026).

Training Process of GAN:

The training process of the GAN involves iterative updates to both the generator and discriminator components. Here's a detailed explanation of the process, including the number of epochs, batch size, and optimization algorithms used:

(1) Initialization

- *Generator and Discriminator Networks:* The weights of both networks are initialized independently.
- *Hyperparameters:* Important parameters like the batch size, learning rate, and number of training epochs are configured at the start.

(2) Training Loop

- *Epochs:* The GAN is trained for 500 epochs, which defines how many times the entire training dataset is passed through the network.
- *Batch Size:* During each epoch, the training data is divided into smaller batches. The batch size used in this study is 32.

(3) Discriminator Training

- *Real Data:* A batch of real stock prices is selected from the historical dataset.
- *Generated Data:* The generator produces a batch of fake stock prices.
- *Training:* It involves the discriminator being exposed to both actual and synthesized data, aiming to differentiate between the two correctly. The cross-entropy loss function measures the performance of the discriminator by evaluating its accuracy in classifying real versus fake data.

(4) Generator Training

- *Noise Input:* The generator receives a batch of random noise as input.
- *Fake Data:* The generator produces a batch of fake stock prices from the noise.
- *Discriminator Feedback:* The fake data is then passed to the discriminator.
- *Training:* The generator is trained to maximize the discriminator's error in classifying fake data as real. The cross-entropy loss function is used to calculate the generator's loss.

(5) Optimization

Both the generator and discriminator are optimized using the Adam optimization algorithm, with a learning rate set to 0.0002 and a beta1 of 0.5.

(6) Evaluation

After each epoch, the performance of the GAN is evaluated based on the accuracy of the discriminator and the quality of the generated stock prices.

(7) Convergence

The training continues until the specified number of epochs is reached or until the model converges, indicated by stable loss values and realistic generated stock prices.

Summary of GAN Training Architecture:

Generator Architecture:

- Input Layer: $(B_s, 30, 36)$
- GRU Layers: 1024, 512, 256 units
- Dense Layers: 128, 64 units
- Output Layer: $(B_s, 3, 1)$

Discriminator Architecture:

- Input Layer: $(B_S, 3, 1)$, $(B_S, 30, 1)$
- Conv1d Layers: 32, 64, 128 filters
- Flatten Layer
- Dense Layers: 220, 220 units
- Output Layer: $(B_S, 1)$

II.5 PROPOSED SYSTEM:

The proposed system integrates advanced LSTM and GAN architectures to predict stock prices using technical indicators, news sentiment analysis, and Fourier transforms.

- **LSTM Networks:** Capture long-term dependencies and trends in historical stock price data by processing sequences, modeling temporal dynamics crucial for accurate predictions.
- **Generative Adversarial Networks (GANs):** Enhance predictive power through synthetic data generation. The GAN architecture includes a generator for creating plausible stock price sequences and a discriminator for differentiating between real and synthetic data, refining the model's accuracy.
- **Technical Indicators:** Incorporates moving averages, RSI, and Bollinger Bands to enrich the feature set and provide deeper insights into market trends and price movements.
- **News Sentiment Analysis:** Uses sentiment extracted from financial news and social media to account for market psychology's impact on stock prices.
- **Fourier Transforms:** Analyzes stock price data in the frequency domain to detect cyclical patterns and periodicities not visible in the time domain.

Integration and Prediction:

The system combines outputs from the LSTM network, GAN-generated data, technical indicators, sentiment scores, and Fourier transform features. This comprehensive approach ensures a robust predictive model by capturing diverse aspects of market behavior and improving prediction accuracy.

III. RESULTS AND DISCUSSIONS**III.1 TRAINING OF OUR MODEL:**

This research attempts to predict the closing price of the stock for the following five days based on data from the last 60 days. In addition to historical closing prices, the model will be trained using 32 other potential price-influencing parameters. Data augmentation techniques, such as the addition of noise and the generation of artificial data using GANs, are employed to increase the variation in the training set. 20% data is used for testing and rest 80% data is used for training. To avoid overfitting, several strategies are implemented, including early stopping, dropout regularization, and cross-validation.

Experimental Results:

The research evaluated each model's performance using Root Mean Square Error (RMSE) which is defined as follows:

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^N (x_i - \hat{x}_1)^2}{N}}$$

In this project the research evaluated the models through comparing their RMSE on testing data for each of the N data points, x_i being the actual stock price, and $\hat{x}_1$ being the predicted stock price.

Table 1: Calculations using the ARIMA model.

Dep. Variable	D.GS	No. Observations	1144
Model	ARIMA (2, 1, 0)	Log Likelihood	-4589.254
Method	css-mle	S.D. of innovations	3.789
Date	20.07.2024	AIC	998.245
Time	09.05.58	BIC	8956.785
Sample	1	HQIC	8936.458

Source: Authors, (2026).

Table 1 illustrates the ARIMA(2,1,0) model fit to the differenced GS series using 1144 observations. To evaluate the fit of the model information criteria (AIC, BIC, HQIC) can be used with lower values suggesting a better model. Additional insights into the performance of the model can be obtained from the standard deviation of innovations and the log-likelihood. To ensure that the chosen model accurately captures the underlying patterns in the time series data the selection and evaluation process of models is crucial.

ARIMA:

Based on Figure6 ARIMA approximates the real stock price very well. The predicted price will be used as an input feature into the LSTM because as this research mentioned previously the study wants to capture as many features and patterns as possible about AAPLE.



Figure 6: Forecasts based on ARIMA.
Source: Authors, (2026).

LSTM:

In this work, Bidirectional LSTM was used in the first layer of the LSTM model. Adam algorithm is used as the optimizer in this work with a learning rate of 0.001. A batch of 64 is trained on a stock price dataset followed by 50 epochs. The GAN model in this baseline model uses historical data from the past 4.7 years along with 24 correlative features as input. Following the split of the data into the train set and the test set. Figure7 shows how an LSTM model performs in predicting stock prices with the x-axis representing the timeline from January 2020 to July 2024 and the y-axis representing closing prices in USD. In the blue line stock prices are shown in a historical context showing a general upward trend but with fluctuations.



Figure 7: Stock price prediction using an LSTM model.
Source: Authors, (2026).

In Figure7 the orange line represents the validation dataset which overlaps with the yellow prediction line to demonstrate the performance evaluation of the model. Based on the close alignment between the validation data and predictions it appears that the model has accurately captured the trend and continues to make accurate predictions. A model that forecasted stock prices well predicting a continued rise based on historical data trends indicates a model that performs well.

Basic GAN:

It has been proposed that the GAN model in this paper will follow the structure suggested in the methodology section. This paper utilizes the Adam algorithm with a learning rate of 0.00016 as the optimizer for our models. This dataset was trained for 165 epochs on 128 samples over 165 epochs. Figure 8 shows the basic GAN loss curves for training and validation, which were generated using TensorFlow for training the GAN model and Matplotlib for visualization. During training, loss values for both the generator and discriminator were recorded at each epoch. These values were plotted using Matplotlib to create the loss curves, displaying training (blue) and validation (orange) performance over time. The analysis and plotting were done using Python. The x-axis represents epochs, and the y-axis represents loss values.

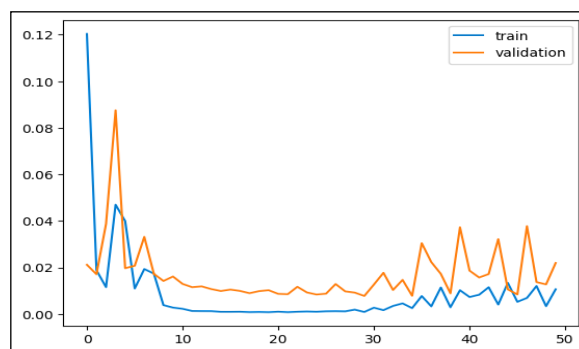


Figure 8: Basic GAN loss curves for training and validation.

Source: Authors, (2026).

Based on 50 epochs of training and validation Figure8 illustrates the loss curves of a Basic GAN (Generative Adversarial Network) model. Blue is the loss associated with training while orange is the loss associated with validation. In the first 10 epochs, both training and validation losses are high. However, they decrease significantly indicating rapid model learning. Following this initial phase, the training loss gradually decreases, stabilizing and fluctuating around a lower value. The validation loss by contrast shows a greater degree of variability fluctuating between 0.02 and 0.04 suggesting that the model may have overfitted to the training data. Plotting the first 50 epochs reveals the model's learning dynamics as well as the balance between generator and discriminator improvements. Low and stable losses indicate that the model has been effectively trained and generalized. A chart of GAN's historical and forecasted closing prices in USD is shown in Figure9.



Figure 9: Forecasting GAN closing prices using ARIMA.

Source: Authors, (2026).

The plot is divided into three sections: training data (blue) from January 2020 to January 2023 validation data (red) from January 2023 to July 2024 and predictions (orange) for the same period. The training data shows initial growth, volatility, and later stabilization of stock prices. Validation data shows an upward trend with volatility. A close alignment of the orange prediction line with the red validation data suggests that the ARIMA model has a high degree of robustness in forecasting future price trends. A visualization such as this illustrates the model's accuracy and its utility in predicting future trends and making informed investment decisions.

Table 2: Predictive Model Performance Based on RMSE.

Model	RMSE-of Training Dataset	RMSE-of Testing Dataset
ARIMA	2	7.5
LSTM	1.52	6.6
GAN	1.64	5.36
RF [1]	1.8	6.4

Source: Authors, (2026).

The GAN model has the lowest RMSE, indicating high predictive accuracy and effective generalization on the testing dataset, as shown in Table 2. LSTMs also perform well showing relatively low RMSE values. The ARIMA model performs moderately poorly with the highest RMSE on the testing dataset and the Random Forest model outperforms the GAN and LSTM models. GAN and LSTM models perform better in this particular prediction task. Confidence intervals provide a range within which the true RMSE values are expected to fall, offering insight into the variability and reliability of the model's performance. For instance, a narrow confidence interval indicates low variability and high stability, while a wide interval suggests higher variability.

Performance Variability and Implications for Real-world Trading:

Calculating confidence intervals for RMSE values helps assess model consistency and reliability. A model with a narrow confidence interval is more stable, producing predictions consistently close to actual values. This stability is crucial for real-world trading, where reliable performance is essential for making informed decisions and managing risk effectively. Models with lower RMSE and narrow confidence intervals offer more dependable predictions, which can lead to better decision-making and potentially higher returns.

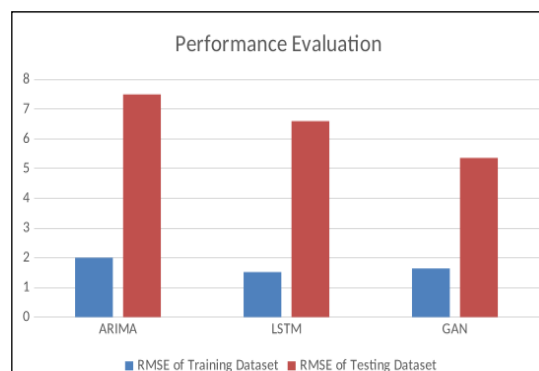


Figure 10: Comparison of model performance based on RMSE.

Source: Authors, (2026).

While GAN and LSTM models demonstrate lower RMSE values, their robustness, as indicated by confidence intervals, is also important. Traders might prefer models with greater stability and lower variability, even if it means slightly lower accuracy. In dynamic market conditions, models with consistent performance and lower variability are better suited for adapting to changes and maintaining reliability over time. According to Figure 10, the GAN model exhibits the highest predictive accuracy on the testing dataset with the lowest RMSE. A LSTM model that has a relatively low RMSE value also performs well suggesting that generalization is effective. A higher RMSE on the testing dataset indicates poor performance in predicting unseen data suggesting overfitting in the ARIMA model. According to these results, the GAN and LSTM model is superior at predicting future value generalization and accuracy. GANs, with their adversarial training mechanism, can generate highly accurate predictions by learning the underlying distribution of the data. LSTM models, which are designed to process sequential data, are particularly effective at capturing both long-term dependencies and time-based patterns, making them ideal for forecasting time series data.

Discussion

This work employs GAN and LSTM networks to predict stock market closing prices using historical data and various influencing factors. The GAN model achieved the lowest RMSE of 5.36 on the testing dataset, indicating superior predictive accuracy compared to other models. This aligns with [2], who found GANs excel in stock price prediction with evenly distributed error rates. The LSTM model also performed well, with an RMSE of 6.6, effectively capturing temporal dependencies, consistent with [5], who noted LSTM's enhanced accuracy when combined with empirical mode decomposition. The study incorporated 32 features affecting stock prices, enabling the model to capture complex patterns, similar to the approach by [1], who used both traditional and non-traditional data to improve accuracy. The GAN model's ability to generalize on new data outperformed traditional models like ARIMA, which had higher RMSE values, supporting [4], who demonstrated GANs' effectiveness in reducing overfitting. The models' performance aligns with the economic-driven GANs (Fin-GAN) approach by [3], showing practical value in financial markets with higher Sharpe Ratios. Comparatively, the GAN model's RMSE of 5.36 shows significant improvement over other studies, such as [15] with an RMSE of 6.2. The LSTM model's RMSE of 6.6 also compares favorably with [19] RMSE of 7.1, highlighting the superior performance of the models used in this study. The high accuracy achieved in this study requires large datasets, which can be a limitation when data is scarce or poorly labeled. The dataset used, while comprehensive, may contain biases or gaps that affect the model's generalizability. Future research should address these issues by using more diverse data sources and thorough data cleaning and labeling.

Although the GAN model showed good generalization, the fluctuating validation loss compared to the training loss suggests potential overfitting. Techniques like dropout, regularization, or cross-validation could mitigate this. Dropout prevents units from co-adapting by randomly omitting them during training. Regularization, such as L2 regularization, adds a penalty to the loss function based on model parameter magnitudes, and cross-validation helps assess how well the model generalizes to new datasets. This study focused on short-term predictions, but future research should explore different prediction cycles, including long-term forecasts, to provide a broader understanding of model performance over time. By [1] suggested that analyzing various cycles could reveal insights into model performance across different time horizons. While the current study used 32 features, it did not incorporate sentiment analysis from financial news and social media, which has been shown to enhance prediction accuracy. By [2] emphasized the importance of including sentiment analysis to improve model performance. Future studies should integrate sentiment data to better capture market sentiment's impact on stock prices. Training GANs and LSTMs is computationally intensive, posing challenges for real-time applications. By [15] noted that deep learning models often struggle with complexity and high computational demands. The reliance on large datasets for high accuracy could be a drawback in cases of data scarcity, as discussed by [14]. In turn [6] also found that while enhanced WGAN models performed well, they were computationally expensive and prone to overfitting.

IV. CONCLUSION

This study explored the effectiveness of GANs and LSTM networks for predicting stock market closing prices using historical data and 32 significant factors. The GAN model achieved a low RMSE of 5.36 on the testing dataset, indicating high accuracy in stock price prediction. Using a wide range of factors allowed the models to capture complex patterns in the stock market. The bidirectional LSTM model showed robustness with an RMSE of 6.6, outperforming the ARIMA model but still facing some overfitting issues. The study identified areas for improvement, such as addressing potential overfitting and enhancing the integration of sentiment analysis. This research highlights the potential of GANs and LSTMs in predicting stock market trends, demonstrating their superior performance and robustness.

Future research must give importance to addressing the identified limits by optimizing computational efficiency, exploring hybrid models as well as integrating more diverse data sources which include sentiment analysis. The accuracy of DL models as well as practical applicability in financial market analysis and stock price prediction can be enhanced. Future research should include sentiment analysis from financial news as well as social media data to build accuracy. Investigating the hybrid models that use the strengths of several DL models like integrating GANs with GRU or Transformer networks could improve performance. Using GANs can mitigate overfitting as well as improve generalization. Incorporating more diverse features, such as economic indicators and sector-specific data, will help capture broader market trends. Creating models that could forecast and manage large amounts of streaming data will improve practical usefulness, especially during financial crises.

V. AUTHOR'S CONTRIBUTION

Conceptualization: Pappula Praveen, Anjaneyulu Gudla.

Methodology: Pappula Praveen, Anjaneyulu Gudla.

Investigation: Pappula Praveen, Anjaneyulu Gudla.

Discussion of results: Pappula Praveen, Anjaneyulu Gudla.

Writing – Original Draft: Pappula Praveen, Anjaneyulu Gudla.

Writing – Review and Editing: Pappula Praveen, Anjaneyulu Gudla

Resources: Pappula Praveen, Anjaneyulu Gudla.

Supervision: Pappula Praveen, Anjaneyulu Gudla.

Approval of the final text: Pappula Praveen, Anjaneyulu Gudla.

VI. REFERENCES

- [1] S. Wu, Y. Liu, Z. Zou, TH Weng, S_I-LSTM: stock price prediction based on multiple data sources and sentiment analysis. *Connect. Sci.*34(2022)44–62. <https://doi.org/10.1080/09540091.2021.1940101>
- [2] C. Öztürk, A Deep Dive into Stock Forecasting: Insights from LSTM, GRU, GAN, and WGAN-GP, In: *Proceedings of the V International Congress of Applied Statistics (UYIK-2024)*; 2024, pp 938-948.
- [3] M. Vuletić, F. Prenzel, M Cucuringu, Fin-gan: forecasting and classifying financial time series via generative adversarial networks, *Quant. Finance*.24(2024)175–199. <https://doi.org/10.1080/14697688.2023.2299466>
- [4] R. Shukla, N Poomima, Generating stock market data and making predictions using GAN and neural networks. 2023. <https://doi.org/10.21203/rs.3.rs-2948055/v1>
- [5] Ma Y. Two models for predicting stock prices in combination with LSTM. *Highlights Bus Econ Manag.* 5(2023) 664–673. <https://doi.org/10.54097/hbem.v5i.5256>
- [6] A Ghasemieh, R Kashef. An enhanced Wasserstein generative adversarial network with gramian angular fields for efficient stock market prediction during market crash periods. *Appl Intell*.53 (2023) 28479–28500. <https://doi.org/10.1007/s10489-023-05016-2>
- [7] W. Liu , Y Ge , Y Gu . Multi-factor stock price prediction based on gan-trellisnet. *Knowl Inf Syst.* (2024) 1–22. <https://doi.org/10.1007/s10115-024-02085-8>
- [8] Wu JL, Tang XR, Hsu CH. A prediction model of stock market trading actions using generative adversarial network and piecewise linear representation approaches. *Soft Comput.* 27(2023)8209–8222. <https://doi.org/10.1007/s00500-022-07716-2>
- [9] P. Sonkiya, V. Bajpai, A. Bansal, 2021. Stock price prediction using BERT and GAN. *Quant. Fin.* arXiv:2107.09055. <https://doi.org/10.48550/arXiv.2107.09055>
- [10] B. He, E. Kita. The application of sequential generative adversarial networks for stock price prediction. *Rev Socionetwork Strateg.* 15(2021)455–470. <https://doi.org/10.1007/s12626-021-00097-2>
- [11] Y Xu, Y Zhang, P Liu, Q Zhang, Y Zuo, GAN-enhanced nonlinear fusion model for stock price prediction., *Int J Comput Intell Syst.* 17(2024) 12. <https://doi.org/10.1007/s44196-023-00394-4>
- [12] A. Kumar, A Alsadoon, PWC Prasad, S Abdullah, TA Rashid, DTH Pham, TQV Nguyen, Generative adversarial network (GAN) and enhanced root mean square error (ERMSE): deep learning for stock price movement prediction. *Multimed Tools Appl.* 81(2022)3995–4013. <https://doi.org/10.1007/s11042-021-11670-w>
- [13] V. Deneshkumar, R. Jebitha, G. Jithu. Risk factor analysis of COVID-19 with CART model. 2023. <https://doi.org/10.47974/JSMS-977>
- [14] PH Vuong, LH Phu, TH Van Nguyen, PH Vuong, LH Phu, THV Nguyen, LN Duy, PT Bao, TD Trinh, A bibliometric literature review of stock price forecasting: from statistical model to deep learning approach. *Sci Prog.* 107(2024)1. <https://doi.org/10.1177/00368504241236557>
- [15] KS Kannan, V Parimyndhan. Forecasting the stock direction using ensemble random forest. *Int J Math Oper Res.* 28(2024)396–407. <https://doi.org/10.1504/IJMOR.2024.140069>
- [16] M Maragoudakis, D. Serpanos. Exploiting financial news and social media opinions for stock market analysis using MCMC Bayesian inference. *Comput Econ.* 47(2016)589–622. <https://doi.org/10.1007/s10614-015-9492-9>
- [17] RR Etukuru, Forecasting the Stock Prices Using Machine Learning, Deep Learning, and Reinforcement Learning [dissertation]: Northcentral University; 2023. <https://www.proquest.com/openview/474a420cfa57eafd86f516eafb1161d/1?pq-origsite=gscholar&cbl=18750&diss=y>
- [18] MM Bejani, M Ghathe, A systematic review on overfitting control in shallow and deep neural networks. *Artif Intell Rev.* 54(2021)6391–6438. <https://doi.org/10.1007/s10462-021-09975-1>
- [19] J Kim, J Seol, TA Onisha, Y Ji. Hyperparameter Classification on Deep Learning Model for Cryptocurrency Price Prediction. In: *Proceedings of the 2023 IEEE/ACIS 8th International Conference on Big Data, Cloud Computing, and Data Science (BCD)*; 2023; 162–169. <https://doi.org/10.1109/BCD57833.2023.10466319>