Deerwalk Journal of Computer Science and Information Technology

Prediction of Stock Trading Action using Generative Adversarial Network in Nepali Stock Market

Pranav Dahal *

dahalpranav01@gmail.com

Abstract

Can deep learning approaches achieve similar accuracy as statistical models in predicting the stock market? This research introduces an approach integrating Generative Adversarial Networks (GANs) with Long Short-Term Memory (LSTM) generators and Convolutional Neural Network (CNN) discriminators to predict stock trading actions. By analyzing and computing on historical Open, High, Low, Close and Volume (OHLCV) data of the stock market, we generate a wide range of input features using Fourier transforms, auto-encoders, and several technical indicators. Extreme Gradient Boosting (XGBoost) is employed to filter and extract important features. The extracted features are fed into the LSTM generator, whose output is then fed to the CNN discriminator to discriminate the buy, sell, or hold signals. Results demonstrate the model's effectiveness in capturing temporal and spatial patterns, with implications for refining stock trading strategies and enhancing financial decision-making for maximizing profits.

Keywords: *Stock Prediction; Technical Analysis; GAN Model; Stock Trading*

1. Introduction

The financial landscape, marked by its dynamic and complex nature, has prompted the need for advanced predictive tools in stock trading. Traditional models face challenges in capturing intricate patterns and adapting to evolving market dynamics. In response, this research focuses on developing a robust predictive model by harnessing the potential of Generative Adversarial Networks (GANs) alongside advanced machine learning techniques. The motivation stems from the realization that current methodologies fall short in providing accurate predictions. The primary objective is to design a model capable of generating precise buy, sell, or hold signals for stocks, utilizing historical OHLCV data and extracting meaningful features and to provide investors with an adaptive and accurate tool that enhances decision-making in the dynamic and complex landscape of stock markets. This study will include prediction on historical OHLCV data of Nepali stock market from March 23,2018 to August 23,2024.

1.1. Background and Motivation

Traditional approaches to stock trading predictions often fall short in capturing the intricate patterns and dynamic market behavior, leaving investors in need of more sophisticated tools. The increasing availability of historical OHLCV data provides an opportunity to leverage advanced machine learning techniques for improved forecasting. This research is motivated by the growing complexity of financial markets, where unforeseen events and dynamic trends make traditional models less reliable. For instance, recent global events have underscored the need for adaptable and precise predictive tools to navigate the uncertainties in stock markets. Moreover, a comprehensive review of existing literature reveals a gap in methodologies that effectively combine Generative Adversarial Networks (GANs) with LSTM generators, CNN discriminators, and advanced feature extraction techniques for stock trading predictions. By addressing these challenges, this research aims to contribute a relevant and timely solution to the growing demand for accurate and adaptive predictive models in the realm of stock trading.

1.2. Problem Definition

The problem at hand revolves around the prevalent issue of investors with insufficient stock market knowledge making suboptimal decisions in buying and selling stocks, resulting in significant financial losses. There is a clear and pressing need to address this problem as it hampers the financial well-being of individual investors and undermines the efficiency of the broader financial market. Investors lacking comprehensive understanding of stock market dynamics often struggle to interpret complex patterns and to adapt with changing market conditions. This deficiency in knowledge and decision-making capability underscores the significance of developing a predictive model that can guide investors by generating accurate buy, sell, or hold signals for stocks. The proposed solution involves the integration of machine learning techniques, including Generative Adversarial Networks (GANs), built with LSTM generators, and CNN discriminators, along with several feature extraction methods. By combining these elements, the objective is to empower investors with a robust and adaptive tool that enhances decision-making in the unpredictable realm of stock trading, ultimately mitigating financial losses and fostering more informed investment strategies.

1.3. Scope and Limitation

The scope of this research project is to provide traders with basic knowledge of the stock market a user-friendly predictor that enables them to make informed decisions and maximize their profits. The project will encompass the development and implementation of a predictive model, integrating Generative Adversarial Networks (GANs) with LSTM generators, CNN discriminators, and several feature extraction techniques. The focus will be on creating a tool that simplifies complex market dynamics, allowing traders with limited expertise to effectively interpret and act upon generated signals for buy, sell, or hold actions. The scope also extends to evaluating the model's performance and usability in real-world trading scenarios.

While the project aims to empower traders with basic knowledge, it is essential to acknowledge certain limitations. The predictor's effectiveness may be influenced by external factors such as market emotions, unexpected market events, economic shifts, or sudden geopolitical developments, which are inherently challenging to predict. Additionally, the success of the predictor depends on the accuracy and reliability of historical data, and any anomalies or inaccuracies in the data may impact the model's performance. These limitations are crucial to consider in understanding the boundaries and applicability of the proposed predictive model.

2. Background Study and Literature Review

2.1. Background Study

The background study for this project delves into the fundamental theories, general concepts, and terminologies integral to the development of a predictive model for stock trading actions. In the dynamic landscape of financial markets, understanding key principles is crucial for devising effective strategies and tools. Fundamental theories such as market efficiency, random walk hypothesis, and behavioral finance provide the groundwork for comprehending the underlying dynamics of stock price movements.

Key concepts, such as technical indicators, Fourier transform, and auto encoders, form essential components of feature extraction techniques. These techniques aim to capture relevant information from historical OHLCV data, enriching the model with meaningful insights. The integration of Generative Adversarial Networks (GANs), LSTM generators, and CNN discriminators aligns with advancements in machine learning, offering a holistic approach to predictive modeling [1].

Terminologies related to stock trading actions, including Buy, Sell, and Hold, serve as the core output of the model. Understanding the implications of these actions within the context of trading strategies is essential for users, particularly those with basic stock market knowledge.

The primary aim of this research project is to bridge the gap between theoretical concepts and practical application by developing a user-friendly predictor. By synthesizing these elements, the predictive model seeks to empower traders with limited expertise, enabling them to make informed decisions and maximize profits in the ever-evolving landscape of stock markets. The subsequent literature review will elaborate on these concepts, providing a more in-depth exploration of existing studies and frameworks that have paved the way for the proposed aims of this research.

2.2. Literature Review

The paper “Stock Market Prediction Based on Generative Adversarial Network" by Guoqiang Zhong proposes a novel architecture based on Generative Adversarial Networks (GANs) for predicting stock market closing prices. It utilizes a Long Short-Term Memory (LSTM) network as the generator and a Multi-Layer Perceptron (MLP) as the discriminator [2].

Employing GANs for stock market prediction is a relatively new and potentially beneficial direction. The adversarial training between generator and discriminator encourages learning complex, realistic data distributions. Utilizing LSTM, adept at handling sequential data, aligns well with the time-series nature of stock prices. Also, the paper compares the proposed GAN model with traditional methods like ARIMA, demonstrating some improvement in prediction accuracy [2].

There are some limitations mentioned in the paper like, the study uses data from a single stock over a relatively short period. Incorporating broader market data and diverse features could enhance generalizability and capture more complex dynamics. While effective, GANs often lack interpretability, making it difficult to understand why predictions are made and limiting trust in their results. Also, GANs are susceptible to overfitting, especially with limited data. The paper does not explicitly address this concern [2].

Overall, this paper presents an interesting exploration of GANs for stock market prediction. While promising, it highlights the need for further research to address data limitations, interpretability, and robustness before practical application [2].

The paper “A prediction model of stock market trading actions using generative adversarial network and piecewise linear representation approaches” has proposed the GAN-based frameworks to improve the prediction performances of trading strategy. The experimental results indicate that the GAN-based frameworks yield excellent performance; in particular, the WGAN and WGAN-S predictors yield high average CRs and counts of positive CRs. However, the GAN-based predictors also outperform the traditional supervised learning models such as the H-LSTM predictor. The generator has predicted the more effective trading actions, and the discriminator has distinguished whether these trading actions originate from the PLR or the generator [3].

In addition, the PLR approach provides a reliable training target for GAN-based frameworks and improves stock market trading performance. Therefore, the GAN-based framework has generated more diverse trading strategies and has decreased the overfitting problem. There are few limitations in the paper, buying and selling actions are all-in or all-out on the stock market. In practice, there are fixed units for stock trading. Moreover, herein, trading fees and transaction levies are not considered when ROI is computed, and traders may be required to pay larger amounts when executing trades frequently. In the future, seed trading action sequences for GAN models can apply multiple methods, such as the buy and hold strategy and the stock ticker strategy. Therefore, the discriminator of the GAN-based framework will be able to obtain more real trading action sequences from historical trading information to detect whether a sequence is real or fake [3].

The generator of GAN is used to generate/predict daily trading actions, and the discriminator is used to detect the real/fake trading actions from the PLR/generator of GAN. Experimental results indicate that the proposed GAN-based frameworks outperform the LSTM network [3].

2.3. Existing System

In the context of stock market prediction, existing applications such as MeroLagani's AI Chart have paved the way by employing artificial intelligence to provide traders with strategies for buy, sell, and hold decisions based on pivot points, support, and resistance levels. This application serves as an interesting point of reference, and our research draws inspiration from certain aspects while aiming to introduce novel elements to enhance predictive capabilities [4].

MeroLagani's AI Chart's focus on pivot points, support, and resistance aligns with traditional technical analysis methods. Our research acknowledges the importance of these elements in understanding market trends and incorporates them as foundational features. However, our approach extends beyond these conventional indicators by integrating advanced machine learning techniques like Generative Adversarial Networks (GANs), LSTM generators, and CNN discriminators.

3. Algorithm Details

3.1 Fourier Transform

Fourier Transformation is a mathematical technique that can be used in stock price prediction to analyze and identify underlying patterns in the data. Fourier Transformation is based on the principle that any periodic function can be represented as a sum of simple sine and cosine waves with different frequencies and amplitudes.

Fourier transforms take a function and create a series of sine waves (with different amplitudes and frames). When combined, these sine waves approximate the original function. Mathematically speaking, the transforms look like this:

$$G(f) = \int_{-\infty}^{\infty} g(t) e^{-i2\pi f t} dt \quad \text{.....(i)}$$

We will use Fourier transforms to extract global and local trends in the stock prices, and to also denoise it a little.

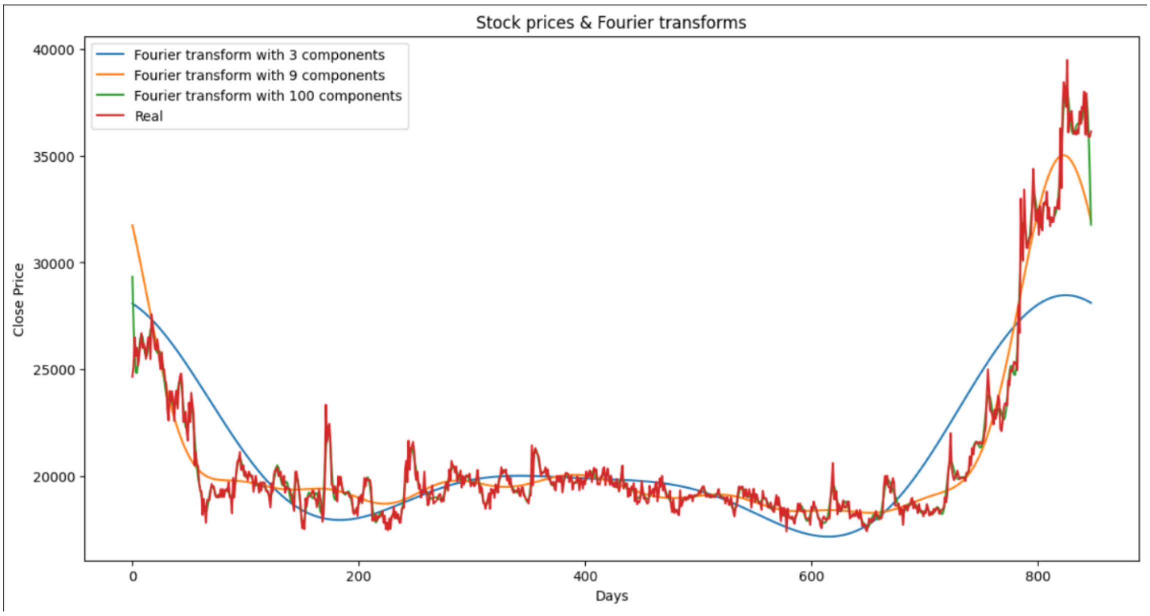

Fig.1: Fourier Transform to Extract features.

As we see in Fig.2, the more components from the Fourier transform we use the closer the approximation function is to the real stock price (the 100 components transform is almost identical to the original function - the red and the purple lines almost overlap). We use Fourier transforms for the purpose of extracting long-term and short-term trends. We select the transforms with 3, 6, and 9 components. For this study, the transform with 3 components serves as the long-term trend.

3.2 Support/Resistance Buy-Sell Strategy using K-means for Support/Resistance level identification.

K-means clustering is a popular unsupervised machine learning algorithm used for partitioning a dataset into K distinct, non-overlapping clusters.

Here's how the algorithm works:

1. Initialization: Choose K initial centroids randomly from the data points. These centroids represent the centers of the clusters.
2. Assignment: Assign each data point to the nearest centroid. This is typically done by calculating the Euclidean distance between each data point and each centroid and assigning the data point to the nearest centroid.

3. Update: After all data points have been assigned to clusters, recalculate the centroids of the clusters by taking the mean of all the data points assigned to each cluster.
4. Repeat: Repeat steps 2 and 3 until convergence, which occurs when the centroids no longer change significantly, or a maximum number of iterations is reached.
5. Convergence: The algorithm converges when the centroids no longer change significantly between iterations.

The Support/Resistance Buy-Sell Strategy involves identifying key support and resistance levels in the stock price chart. The algorithm typically includes steps to identify historical price points where the stock has consistently reversed direction. The Buy-Sell signals are generated based on breaches or bounces off these levels. For example, a Buy signal might be triggered when the stock price crosses above a resistance level, while a Sell signal may be triggered when it falls below a support level.

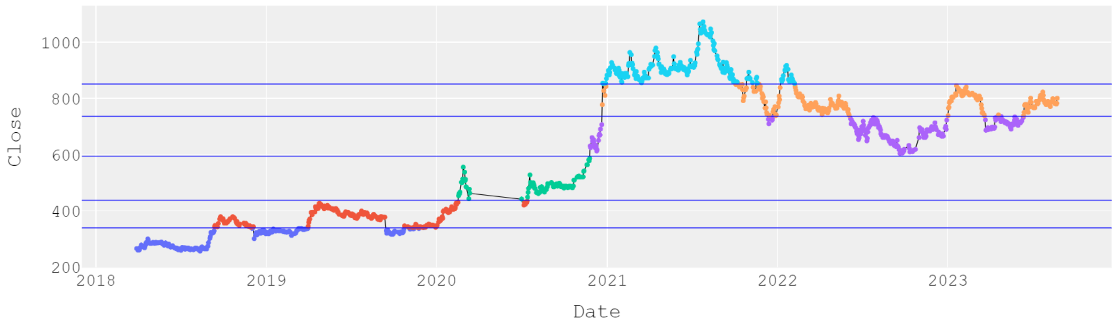

Fig.2: Support Resistance based on k-means.

3.5 GAN (Generative Adversarial Network)

GANs consist of two neural networks, a generator, and a discriminator, engaged in a competitive training process. The generator creates synthetic data, and the discriminator tries to distinguish between real and generated data. In stock trading, the GAN algorithm involves training the generator to produce realistic stock market data, while the discriminator learns to differentiate between real and synthetic data. This adversarial process improves the generator's ability to create data that closely resembles actual market conditions.

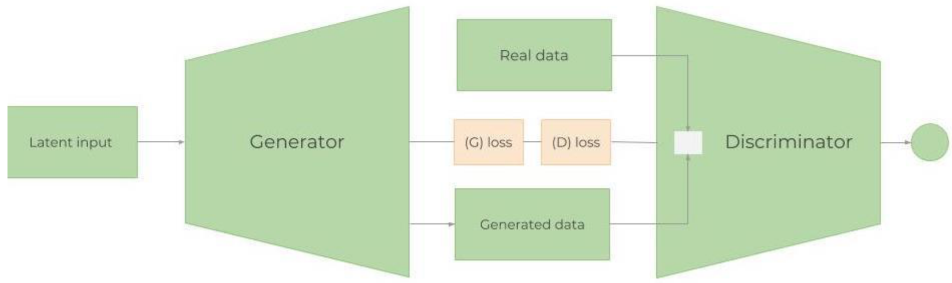

Fig.3: Generative Adversarial Network [1]

3.5.1 LSTM Generator

As mentioned before, the generator is a LSTM network, a type of Recurrent Neural Network (RNN). RNNs are used for time-series data because they keep track of all previous data points and can capture patterns developing through time. Due to their nature, RNNs many times suffer from *vanishing gradient* - that is, the changes the weights receive during training become so small, that they don't change, making the network unable to converge to a minimal loss (The opposite problem can also be observed at times - when gradients become too big. This is called *gradient exploding*, but the solution to this is quite simple - clip gradients if they start exceeding some constant number, i.e. gradient clipping). Two modifications tackle this problem - Gated Recurrent Unit (**GRU**) and Long-Short Term Memory (**LSTM**). The biggest differences between the two are:

1) GRU has 2 gates (update and reset) and LSTM has 4 (update, input, forget, and output),
2) LSTM maintains an internal memory state, while GRU doesn't, and
3) LSTM applies a nonlinearity (sigmoid) before the output gate, GRU doesn't [1].

Our LSTM implementation looks like:

```
# LSTM Generator
generator = Sequential()
generator.add(LSTM(200,activation='relu',
input_shape=(X_train.shape[1], X_train.shape[2])))
generator.add(Dense(3, activation='softmax'))
generator.compile(loss='categorical_crossentropy',
optimizer=Adam())
```

We use this LSTM model as a generator to generate several features that can be used by CNN discriminator to predict the stock trading actions. The LSTM model of the GAN takes extracted features as input which includes the normalized OHLCV, fourier transform, auto-encoder and technical analysis values. The generator outputs 3 layers which signifies the confidence for buy, sell, or hold signal.

3.5.2 CNN Discriminator

CNNs are primarily used for image recognition but can also be applied to sequential data like stock market time series. The algorithm involves convolutional layers that extract spatial features from the input data. In the stock trading context, CNNs can be used to identify spatial patterns or features within the OHLCV data that may contribute to predictive signals. The CNN network comprises Convolutional, Pooling and Fully Connected Layers.

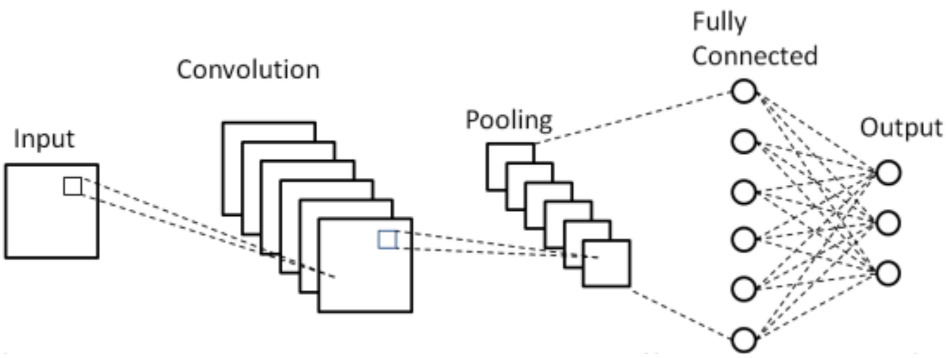

Fig.4: CNN network model

The primary reason for using CNN is that CNNs work well on spatial data - meaning data points that are closer to each other are more related to each other, than data points spread across. This should hold true for time series data. In our case each data point (for each feature) is for each consecutive day. It is natural to assume that the closer two days are to each other, the more related they are to each other. One thing to consider (although not covered in this work) is seasonality and how it might change (if at all) the work of the CNN [1].

The CNN code implementation for our work looks like:

```
# CNN Discriminator

input_shape = (1,3)

discriminator = Sequential()

discriminator.add(Conv1D(32,kernel_size=3,strides=2,input_sh
ape=input_shape, padding='same'))
```

```
discriminator.add(LeakyReLU(0.01))

discriminator.add(Dense(1, activation='sigmoid'))

discriminator.add(Conv1D(64,    kernel_size=3,    strides=2,
padding='same'))

discriminator.add(LeakyReLU(0.01))

discriminator.add(BatchNormalization())

discriminator.add(Conv1D(128,    kernel_size=3,    strides=2,
padding='same'))

discriminator.add(LeakyReLU(0.01))

discriminator.add(BatchNormalization())

discriminator.add(Dense(220))

discriminator.add(LeakyReLU(0.01))

discriminator.add(Dense(220))

discriminator.add(LeakyReLU(0.01))

discriminator.add(Dense(3,activation='softmax')

discriminator.compile(loss='categorical_crossentropy',
optimizer=Adam(),

metrics=['accuracy'])
```

The discriminator is another neural network model that acts as a binary classifier. It is trained to distinguish between real data samples (from the dataset) and fake data samples (generated by the generator). The discriminator takes a data sample as input and outputs a probability indicating whether the input is real or fake. During training, the discriminator's parameters are adjusted to improve its ability to differentiate between real and fake data. The discriminator provides feedback to the generator by indicating how realistic the generated data samples are. This feedback is used by the generator to improve its output.

4. Result Analysis

The predictive model's stock signal prediction mechanism integrates various algorithms to enhance accuracy and adaptability. Starting with Fourier Transform analysis, the model extracts dominant frequencies from historical OHLCV data, providing insights into cyclical patterns. The Support/Resistance Buy-Sell Strategy adjusts levels based on recent price movements, enhancing adaptability to changing market conditions. LSTM modeling captures long-term dependencies, enabling the model to discern intricate patterns for accurate predictions. The CNN architecture extracts spatial features, improving the recognition of fluctuating market dynamics. The GAN model generates synthetic data, diversifying the training set for better adaptability. In the signal generation process, data preprocessing and feature generation precede model training. The dynamic Buy-Sell strategy evaluates evolving levels, and predictions from individual components are integrated for consolidated stock trading signals. This collaborative approach ensures the model's effectiveness in dynamically analyzing market data and generating actionable buy-sell signals.

The Confusion Matrix of the prediction using GAN based on test data is as follows:

On training the historical data with the extracted features for 50 iterations, the resulting accuracy metrics is given as:

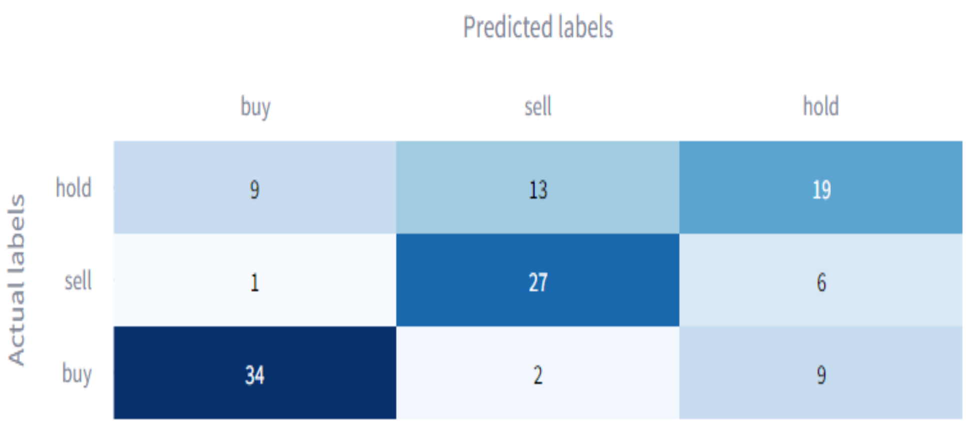

Fig.5: Confusion Matrix of prediction (using 50 epochs for training)

Based on the resulting confusion matrix in Fig.6, the accuracy metrics are computed as:

Accuracy: 66.67%

Recall: 66.67%

Precision: 66.29%

F1-score: 66.09%

While this result is satisfactory in the dynamic scenario of Nepali stock market, we can achieve greater accuracy by increasing the number of iterations to 100 epochs. The accuracy metric after training the GAN model for 100 iterations looks like:

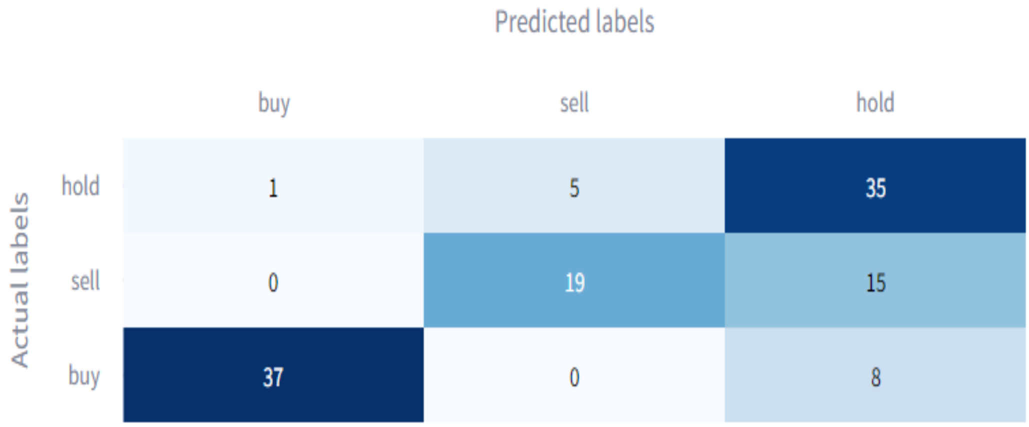

Fig.6: Confusion Matrix of prediction (using 100 epochs for training)

Based on the resulting confusion matrix in Fig.7, the accuracy metrics are computed as:

Accuracy: 75.83%

Recall: 75.83%

Precision: 79.56%

F1-score: 76.16%

The accuracy of the model increased with the increase in the number of training epochs. However, increasing the iterations more would tend to make the model to over fit which in turn affects the accuracy.

5. Conclusion and Future Recommendation

5.1 Conclusion

In conclusion, the exploration of predicting stock trading actions using a Generative Adversarial Network (GAN) and advanced algorithms has been a dynamic journey into financial market forecasting intricacies. The core thesis, focusing on developing a predictive model, integrates Fourier Transform analysis, a dynamic Buy-Sell Strategy, LSTM modeling, CNN feature extraction, and GAN data augmentation. Together, these components create a robust system, adapting to market dynamics and offering insights into stock trading signals. Summarizing key points, the model excels in extracting meaningful patterns from historical data, dynamically adjusting to market changes, and generating profitable trading signals.

Reflecting on broader implications, this research not only advances financial market prediction but also highlights the potential of technologies like GANs in enhancing traditional analytical methods. The synthesis of artificial intelligence and financial modeling paves the way for further exploration. As we navigate stock trading complexities, this study encourages consideration of broader applications for predictive models and the ongoing evolution of technology-driven financial strategies.

5.2 Future Recommendation

To enhance the current predictive model, future efforts should focus on integrating reinforcement learning techniques to fine tune the generator and discriminator. This approach allows for continuous adaptation and optimization of trading strategies in real-time, offering improved performance in dynamic market conditions. Additionally, inclusion of sentiment analysis of news or algorithmic optimizations is recommended for improving the accuracy of the prediction. Expanding the dataset to include a wider range of financial instruments and economic indicators would contribute to a more comprehensive understanding of market dynamics. Also, integration of live data in the project can help to make it more interactive as users can get the predictions on the go as they are trading in the market.

REFERENCES

[1] B. Banushev, "Github," 2019. [Online]. Available: https://github.com/borisbanushev/stockpredictionai. [Accessed 14 12 2023].

[2] J. D. K. Z. Guoqiang Zhong, "Stock Market Prediction Based on Generative Adversarial Network," Ocean University of China, Qingdao, China, 2018.

[3] C.-H. H. J.-L. W. Xian-Rong Tang, "A prediction model of stock market trading actions using generative adversarial network and piecewise linear representation approaches.," Springer Nature, Germany, 2022.

[4] MeroLagani, "Mero Lagani," [Online]. Available: https://aichart.merolagani.com/AICharts.aspx. [Accessed 12 2 2024].