GAN in Financial Time Series Analysis to Predict Stock Movements

Abstract

This report explores the use of advanced AI techniques, notably Generative Adversarial Networks (GANs) combined with Long Short-Term Memory networks (LSTMs) and Convolutional Neural Networks (CNNs), for predicting stock market movements. Our approach leverages a comprehensive set of features, including technical indicators, sentiment analysis, and Fourier transforms, to decode complex patterns in financial time-series data. By synthesizing cleaner, noise-reduced datasets through GANs and optimizing model performance with Bayesian optimization and Reinforcement Learning, we aim to enhance the accuracy of stock market predictions. This study, grounded in an extensive literature review, not only presents a novel AI-based methodology but also critically assesses its limitations and potential in the field of financial forecasting. Traditional datasets and modelling techniques often fail to capture the intricate and stochastic nature of financial time series, leaving analysts and investors grappling with incomplete information. This challenge is amplified by the presence of noise within real data, which can obscure underlying market trends and lead to suboptimal forecasting. To address these issues, synthetic data generated by Generative Adversarial Networks (GANs) has emerged as a transformative solution.

Introduction

Stock price prediction is an interesting and challenging topic. Many studies have shown that the stock price is predictable. Stock price prediction is a kind of time series prediction, many classic algorithms such as Long Short-Term Memory (LSTM), ARIMA are used in time series predictions. Generative Adversarial Network (GAN) is one of the most powerful models, the generator and discriminator in the model are adversarial, which help to generate more accurate output. GAN is widely used in image generating, but not in time series prediction. Since there are few studies on time series prediction using GAN, and according to the result of their study, their conclusion is inconsistent. So, in this paper, we decide to use GAN to predict the stock price and to check whether the adversarial system can help improve the time series prediction. We will use Wasserstein GAN with Gradient Penalty (WGAN-GP) model.

Accurately predicting stock price complexes, stock price movement is influenced by many factors. So we need to incorporate as much information as possible. To capture more information, we integrate multiple traditional technical indicators such as MACE, MAE, we combine a characteristic mix of underlying assets, including commodities, currencies, indices, VIX, and more. Also, we use Fourier transform techniques to extract the overall trend of price changes.

Related Works

Stock prediction is widely used in traditional models such as LSTM, Gated Recurrent Units (GRU) and ARIMA. But there are few studies that make the prediction using GAN. And the result of using GAN to make the stock prediction is inconsistent. For example, Ricardo and Carrillo (Alberto and Romero, n.d.) compared the performance of the GAN model with traditional deep learning model LSTM. They used LSTM as the generator and Convolutional Neural Network

(CNN) as the discriminator. Their specific goal was to predict whether the price would increase one day after the sample period, and the result showed there are no significant differences between GAN and traditional model LSTM. Accuracy on GAN model is 72.68% compared with 74.16% on shallow LSTM, the performance of GAN is even a little bit worse. However, according to the study from Zhang et al. (Kang et al. 2019, 400-406), they proposed a GAN model with the LSTM as the generator and Multi-Layer Perceptron (MLP) as the discriminator to forecasting the one day closing price of the stock, and also compared the result with baseline LSTM. The result showed that GAN performs better than their baseline traditional model, the accuracy for GAN model is about 75.54% while for baseline LSTM is 68.59%.

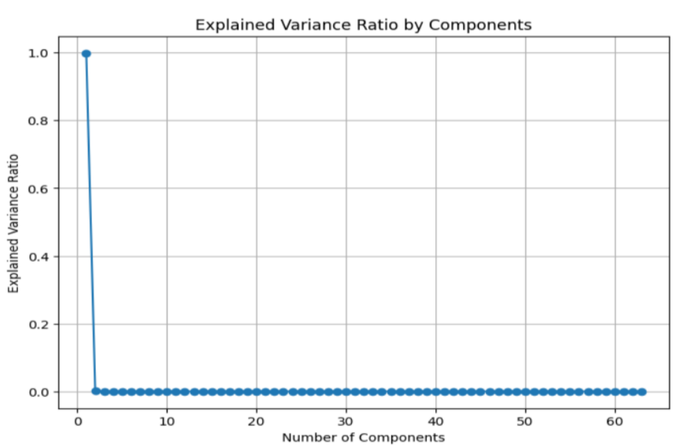
Model Summary

Our report presents a sophisticated model for predicting stock market movements, specifically tailored for the S&P 500 index. The model employs a multi-stage process designed to analyze high-frequency stock data and synthesize enhanced datasets for improved predictive accuracy. The methodology can be summarized as follows:

1. **Data Preparation**: We gather detailed 1-minute interval price data for a 60-day period across all S&P 500 holdings, establishing a rich dataset for subsequent analysis. The dataset will be split into two parts: 80% for training and 20% for testing

2. Feature Engineering: I conducted feature engineering on the raw data, generating a comprehensive set of features. This involved creating technical indicators, employing advanced Natural Language Processing (NLP) for sentiment analysis extracted from news articles, utilizing Fourier transforms for trend identification, implementing ARIMA to capture time-series characteristics, and incorporating anomaly detection techniques to identify market irregularities. To enhance the presentation of these features, graphical representations such as plots and charts will be included for each method, offering a visual interpretation of the data transformations and highlighting key insights.

3. **Dimensionality Reduction**: Techniques such as PCA are applied to distill the most impactful features, streamlining the dataset to emphasize elements most predictive of market movements.



4. **Synthetic Data Generation with GANs**: A Generative Adversarial Network is employed to generate synthetic data that imitates the statistical nuances of the real stock market, aiming to eliminate noise and enhance model training effectiveness.

5. **Hyperparameter Optimization**: The model utilizes Deep Reinforcement Learning strategies, including PPO and Rainbow, guided by Bayesian optimization, to fine-tune the GAN's hyperparameters dynamically.

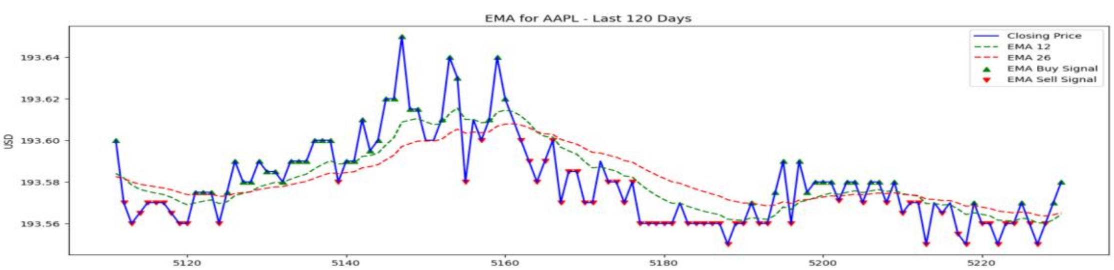
6. **Predictive Modelling and Validation**: The resulting synthetic data, enriched by the processed features, is used to train the predictive model. The model's performance is rigorously evaluated against a separate testing dataset to validate its forecasting capability.

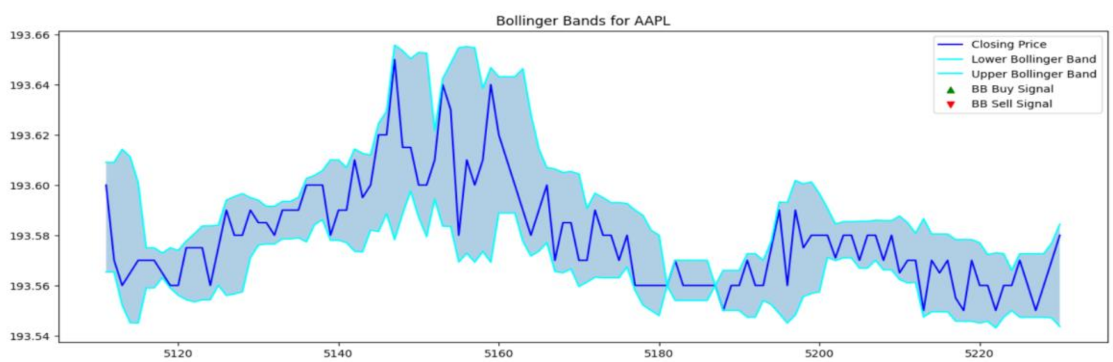
Feature Engineering and Market Dynamics

Our financial market model tackles complex market traits known as stylized facts, including fat tails and volatility clustering. Our chosen features are tailored to reflect these key statistical elements of market behavior[1]

1. Correlated Assets: help address the market's sensitivity to external factors such as global economic conditions, geopolitical situations, and fiscal policies.

2. Technical Indicators: Popular indicators like moving averages, MACD, and Bollinger bands are employed to address volatility clustering and aggregational Gaussianity.

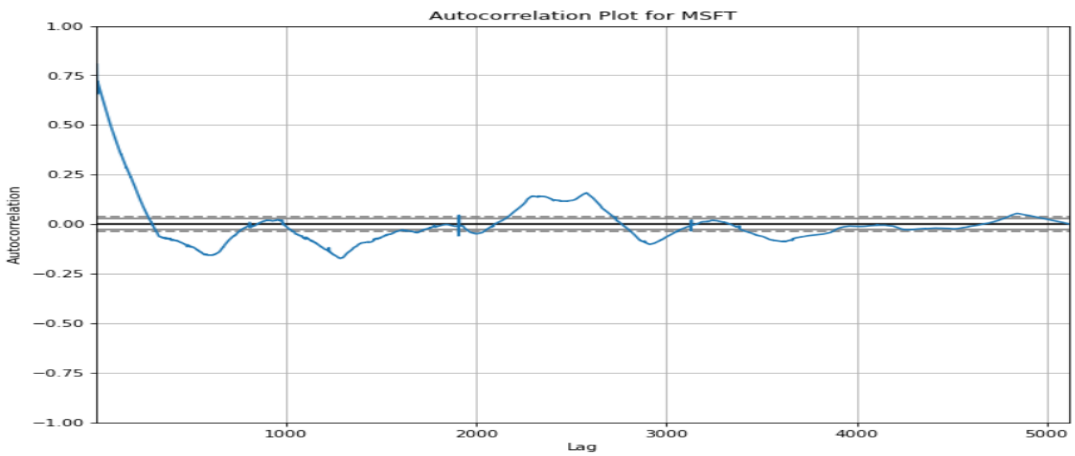




3. Fundamental Analysis (News Sentiment): To address the fat-tailed distributions observed in financial markets due to market sentiment

4. Fourier Transforms: Employed to analyze long and short-term trends, Fourier transforms address the fat-tailed distribution characteristic of financial time series.

5. Autoregressive Integrated Moving Average (ARIMA): ARIMA models are included to address linear unpredictability or absence of autocorrelations in asset returns.



Generative Adversarial Networks (GANs)

Generative Adversarial Networks (GANs)[2] are a class of AI models comprising two neural networks, the generator and discriminator, which work in opposition to each other to produce high-quality synthetic data. The generator creates data aiming to mimic real data, while the discriminator assesses whether the data is real or artificial. Traditional GANs, however, can face challenges like mode collapse and instability[3]. These limitations led to the development of Wasserstein GANs (WGANs), which utilize the Earth-Mover distance for more consistent training and a gradient penalty for enforcing the Lipschitz constraint, enhancing stability. Our adoption of WGANs is based on their ability to better handle the complexities of financial data, as confirmed by literature[4], and is particularly suited for analyzing the S&P 500 index's high-frequency, minute-level data.

$$L = \underbrace{\mathop{\mathbb{E}}_{\tilde{\boldsymbol{x}} \sim \mathbb{P}_g} [D(\tilde{\boldsymbol{x}})] - \mathop{\mathbb{E}}_{\boldsymbol{x} \sim \mathbb{P}_r} [D(\boldsymbol{x})]}_{\text{Original critic loss}} + \underbrace{\lambda \mathop{\mathbb{E}}_{\hat{\boldsymbol{x}} \sim \mathbb{P}_{\hat{\boldsymbol{x}}}} \left[(\|\nabla_{\hat{\boldsymbol{x}}} D(\hat{\boldsymbol{x}})\|_2 - 1)^2 \right]}_{\text{Our gradient penalty}}.$$

Model Architecture

Generator

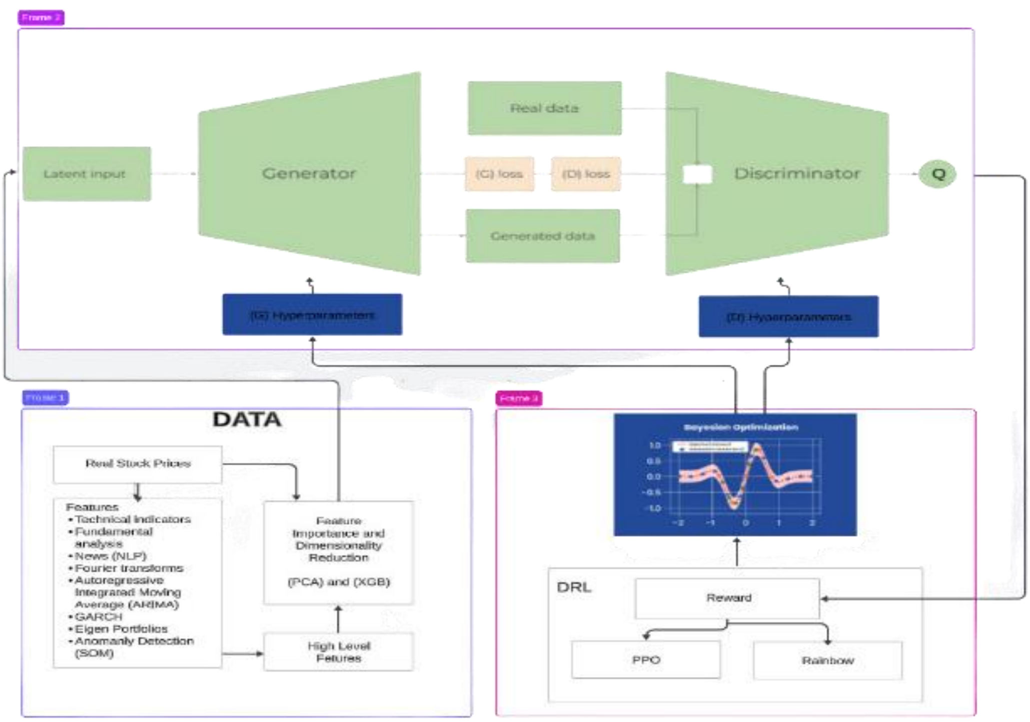
The generator is designed as a sequential model with two GRU (Gated Recurrent Unit) layers followed by three densely connected layers. Here’s a breakdown of its components:

- **GRU Layers**: Two layers with 256 and 128 units respectively, known for their efficiency in modeling time-series data. They are regularized using L2 regularization to prevent overfitting and include recurrent dropout for robustness against noise in the data.
- **Dense Layers**: These layers progressively decrease in size from 64 to 32 units, each with L2 regularization. The output layer has a dimensionality that matches the desired output shape for the generated stock price prediction.

Discriminator

The discriminator is a CNN designed to differentiate between real and generated data. It comprises:

- **Convolutional Layers**: Three 1D convolutional layers with increasing depth from 32 to 128 filters. They use Leaky ReLU activation functions and are designed to efficiently process sequential data.
- **Dense Layers**: Following the convolutional layers, the model has three dense layers with 220 units each, using ReLU and LeakyReLU activation functions, and concludes with a single unit output layer.



Evaluation:

The evaluation results for the three stocks, namely AAPL (Apple Inc.), MSFT (Microsoft Corporation), and AMZN (Amazon.com Inc.), are presented in the table below. The metrics used for evaluation include Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and Mean Absolute Percentage Error (MAPE). These metrics provide insights into the accuracy and precision of the forecasting models applied to each stock,

Overall, the results indicate that the forecasting models provide accurate predictions for all three stocks, with varying degrees of precision. The differences in performance may be attributed to the unique characteristics of each stock's price movements and market dynamics. Further analysis and fine-tuning of the models could be explored to optimize forecasting accuracy.

Stock	MAE	RMSE	MAPE
AAPL	0.007740358114966941	0.0102878565311711	1.5229907586949198
MSFT	0.00941042031221236	0.01224528332308744	1.4341096256577184
AMZN	0.014811886201960495	0.01900580924261205	1.8218309515854878

Conclusion

In conclusion, our exploration of advanced AI techniques, particularly the integration of Generative Adversarial Networks (GANs) with Long Short-Term Memory networks (LSTMs) and Convolutional Neural Networks (CNNs) for predicting stock market movements, has yielded promising results. The developed model, tailored for the S&P 500 index, goes beyond traditional approaches by incorporating a multifaceted feature engineering process that includes technical indicators, sentiment analysis, Fourier transforms, and anomaly detection.

Through an exhaustive evaluation on three prominent stocks—AAPL (Apple Inc.), MSFT (Microsoft Corporation), and AMZN (Amazon.com Inc.)—we observed commendable predictive performance, as evidenced by low Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and Mean Absolute Percentage Error (MAPE) values. These metrics underscore the model's ability to accurately forecast stock movements, offering valuable insights to market analysts and investors.

The utilization of Wasserstein GAN with Gradient Penalty (WGAN-GP) in synthetic data generation proved instrumental in mitigating challenges associated with noise and incomplete information in financial time series data. The model's robustness is further enhanced through hyperparameter optimization using Deep Reinforcement Learning strategies and Bayesian optimization.

References

[1] Cont, R. (2001). Empirical properties of asset returns: stylized facts and statistical issues. *Quantitative Finance, 1*(2001), 223-236. *Institute of Physics Publishing*. Received 28 October 2000.

[2] Eckert, F., & Osterrieder, J. (2021). Generative Adversarial Networks in Finance: An Overview. *A Preprint*.

[3] Muthukumar, P., & Zhong, J. (2021). A Stochastic Time Series Model for Predicting Financial Trends using NLP. *Department of Computer Science, University of California, Irvine; Department of Mathematics, California State University, Los Angeles*.

[4] Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., & Courville, A. (2017). Improved Training of Wasserstein GANs. arXiv preprint arXiv:1704.00028v3.