Efficient Prediction of Annual Yield from Stocks Using Hybrid Deep Learning

Ashish Papanai [0000-0002-0764-2968]

[1] Maharaja Agrasen Institute of Technology, Delhi, India
ashishpapanai00@gmail.com

Abstract. Predicting and analyzing the stock market has been of primary interest to researchers, investors, and market experts. The technology has been evolving continuously from manual to automated collection, tuning, and data analysis to generate insights and predict the rise or fall of a stock. This work presents stockDL, a deep learning solution to analyze, understand the historical stock data and calculate the gross and annual yield for the chosen stock ticker. The proposed solution is comprehensive and user-friendly. It includes data collection and preprocessing and utilizes various mathematical and deep learning techniques for feature extraction combined with state-of-art neural network architectures to predict the market trends. The stockDL algorithm assimilates two traditional stock trading techniques, Buy and Hold strategy and Moving Average ribbon trading strategy, with two Deep Learning Models created using the state-of-art Long Short-Term Memory networks. The first model is a pure LSTM network, whereas the second network is a Mixture of Convolution Neural networks and LSTMs. stockDL uses the data of the past five years from the date of generating the predictions, making the model immune from any sudden fluctuations in the historical data. When evaluated on the four stock symbols (AAPL, GOOGL, HDFCBANK.NSE, RELIANCE.NSE), the model has attained state-of-art for deep learning backed algorithmic trading in a controlled computational environment. The novel solution introduced in this study is faster and more accurate than any existing deep-learning solutions available. It is immune from any sudden dramatic decline among significant sections of the stock market. This work contributes to the stock analysis and research community of technical and financial domains.

Keywords: LSTMs, RNN, CNN, Financial Deep Learning, Stacked LSTMs, Stock Price Prediction, Time Series prediction

1 Introduction

The stock market is an area of high profit and high risks, and this is considered while devising and generating a quantitative investment strategy to predict and judge the stock's future price using the historical stock data. The market governed by various financial and non-financial factors poses a new challenge to the researchers to develop a best-fit solution for predicting the stock prices or the annual yield by investing in

a particular stock. The three significant factors measuring the outcome of investing in a company or business are Environment, Social, and Governance (ESG). Considering ESGs while making investing decisions leads to a favorable outcome in most cases [1].

The Environment of an investment market involves Assessment and Investment vehicles, Financial Markets, Market Structure, Market Intermediaries, Investment process, Regulation, and Economy. The Indian Financial markets include the Credit Market, Money Market, Foreign Exchange Market, Debt Market, and Capital Market. The Social factors include the interactions between neighbors, friends, colleagues, advice given by analysts, planners, bankers [2]. Social media and Interactions are prominent areas to analyze and predict the market trends based on recent activities of significant investors, executives, and influencers [3]. Corporate Governance drafts the rules and regulations for the companies and the shareholders and assists in the smooth day-to-day functioning of trade, The Governance also handles illegal trade practices like Insider Trade, Security Frauds, et cetera [4].

This study follows the idea of conquering one step (or problem) at a time. It focuses on providing a solution to predict the fluctuations on a stock based on the Environment (the Historical Data). With the availability of a massive amount of data, the challenge is to use it and draw meaningful conclusions from it. We have analyzed the data from the date of the initial issue to the date of making predictions. The two deep learning algorithms introduced in this study include a Short-Term Long Memory (LSTM), a recurrent neural network architecture for time-series prediction [5], and a Convolution Neural Network (CNN) and LSTM mix architecture for making the predictions. The baseline comparison of the deep-learning strategy is with two traditional stock trading strategies, the Buy and Hold system and the Moving Averages.

The deep learning model is trained end-to-end on new data (between a training window of the past six years from the day of making the predictions). This strategy prevents any sudden old fluctuations in the data from contaminating the projections. All outliers are scaled using a min-max scaler to stop them from dominating in the results.

Our results suggest a substantial promise in integrating the traditional and computer-generated strategies for developing an improved quantitative investment strategy. The algorithms outperform the baselines set by the conventional approach, and the enhanced LSTM-CNN mix model provides better performance with reduced computational cost. Integration of this model with other deep learning strategies to handle other governing factors like Governance and Social can bring a new revolution in algorithmic trading using Deep Learning.

2 Related Work

Our work connects several relevant pieces of literature. Recent work highlights how Deep Learning can be used in algorithmic trading. With researchers working on individual factors affecting the stock prices, as the social factor, Environment, and Governance [3, 4, 6, 7], a new vision is being added to the analysis and projection of stock

prices based on the contributing factors. Reinforcement Learning libraries like finRL (Financial Reinforcement Learning) and TA-Lib (Technical Analysis Library in Python) have set a new benchmark in Artificial Intelligence for finance.

The use of machine learning algorithms like Support Vector Machine (SVM) [8] and a hybrid feature selection method provided a detailed parameter adjustment procedure with performance under different parameter values. The performance of this algorithm is significantly less than the state-of-art LSTMs.

Various other LSTM based models on long-term data fail to address the computational complexities and the efforts required to train it [9]. Our study focuses on the training aspect of the model and tries to reduce the training time even in a limited computational environment.

The limitation of these developments is that they fail to scale because of high computational requirements [10]. The creation of new replicas of the trading environment and dummy data generation fail to use the existing data. The current deep learning solutions using LSTMs and LSTM-CNN Mix are tested only on short-term predictions ranging from 1 day to 10 days [11].
The main contribution of our work to this problem is to:

1. Use existing, publicly available historical data of stocks to train and test the model.
2. Develop a computationally efficient, less complex, and simple deep-learning solution to improve performance and scalability.
3. Comparing various investment strategies (Traditional and Deep Learning) makes it easier for the user to create a rational decision.

Our work uses the stock market to focus on a crucial issue, to align the human devised strategies with the computer-generated strategy for algorithmic trading and quantitative finance.

3 Data and Background

3.1 Problem Formulation

The current study aims to develop a deep-learning-backed automated In and Out Trading strategy to maximize the annual yield from a single stock. The In and Out trading strategy involves two import decisions:

1. In: Trade in the market if the stock price is predicted to rise for the current month.
2. Out: Sell out and exit the market if the prize is predicted to decrease for the current month.

This study introduces a neural network architecture that learns the historical stock data and predicts the price of the fluctuation in the stock price for the current month thus, solving the market's most important question: "Will the stock soar or crash?". The model receives eight features (highest price, lowest price, opening price, closing price, the volume traded, the first day of the current month, moving average for 12 months, moving average for 24 months) for each month in the period of six years

considered. In the training phase, the model tries to find a pattern in the historical data, which is tested in the test data. Finally, the model makes an In and Out prediction for the next 12 months from the prediction date to provide the user with an investment strategy. This is later used to predict an efficient annual and gross yield. The yield predicted by the deep learning strategy is compared with baseline and popular statistical prediction and trading strategies, which explains the deep learning strategy's efficiency.

3.2 Background

Our work leverages three radically different approaches. The moving average method is mathematical, and the Buy and Hold strategy is positional and confidence-based. In contrast, stockDL focuses on two deep learning architectures, a black-box, and tries to understand the pattern in the historical data and derive meaningful insights from it.

Moving Average Strategy. It follows a set of rules to decide whether to trade In or Stay Out of the market for the month considered. This strategy focuses on smoothening out the price trends by filtering out the noise from short-term predictions. It acts as a support in case of an uptrend of the time frame taken. This method helps understand where Moving Averages will offer support and resistance. Support indicates a price level where we can expect a downtrend, whereas resistance suggests an increase in the price level, that is, an uptrend. Traditional investors have developed various tools to use the moving average to indicate upcoming trends in the prices.

$$\frac{\sum_{i=m}^{n} Price}{m-n} \tag{1}$$

In the formula (1), 'm' is the starting date (Six years before the current date), And 'n' is the final date (date of making the prediction, Current Date)

Buy and Hold Strategy. The investor using this strategy buys stocks and holds them for a long time irrespective of the market fluctuations. This approach is generally long-term and relies on the confidence of the investor. It is a passive investment strategy, and the investor might not sell the possessions at the optimal time.

Deep Learning (LSTM Strategy). LSTM is the backbone of the two architectures defined in this study. LSTM learns to keep the relevant information and forget non-relevant data. RNNs can retain the information at time t about the input seen many timestamps before t; this fails in practical implementation due to the problem of vanishing gradients (the gradient gradually becomes zero because of multiplication of long series of numbers less than zero). LSTM saves the information for later and prevents the older signals from being lost during the processing. The LSTM cells allow the past information to be reinjected later, overcoming the vanishing gradient problem.

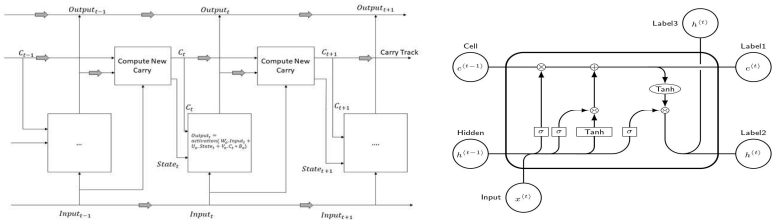

Fig. 1. Anatomy of an LSTM with three input cells [12], Single LSTM Cell

The Fig. 1. Shows the architecture of an LSTM network, it gives a pictorial representation of the activation functions used in the network and the flow of computation of carry by incorporating the past data into the calculating formula.

3.3 Data

The data module of stockDL collects the data required for the study. The data is retrieved from the Yahoo Finance API based on a unique symbol provided to each stock (this unique symbol is called the stock ticker), the starting date, and the end date.

The data table considered in the study (shown as Table 1.) includes information such as opening rate (of the day), the highest rate (of the day), the lowest rate (of the day), the closing rate (of the day), and the volume traded (on that day). As we consider dividends and stock split in calculating the annual yield, the columns comprising it are dropped.

Table 1. First Five Rows of The Historical Stock Data for Further Preprocessing of HDFC Bank Limited

Date	*Open*	*High*	*Low*	*Close*	*Volume*
1996-01-01	2.458312	2.458312	2.373122	2.417745	350000
1996-01-02	2.417746	2.454255	2.393406	2.413689	412000
1996-01-03	2.413689	2.429916	2.393406	2.421802	284000
1996-01-04	2.421802	2.417746	2.385293	2.405576	282000
1996-01-05	2.405576	2.417746	2.393406	2.401519	189000

Table 2. Stock Symbols (Ticker) of the Stocks Used in the Study

Stock Name	*Stock Symbol (Ticker)*	*Stock Exchange*	*Currency*
Alphabet Inc. (Google)	GOOG	NASDAQ Global Select Market	USD
Apple Inc	AAPL	NASDAQ Global Select Market	USD
HDFC Bank Limited	HDFCBANK.NS	National Stock Exchange	INR
Reliance Industries Limited	RELIANCE.NS	National Stock Exchange	INR

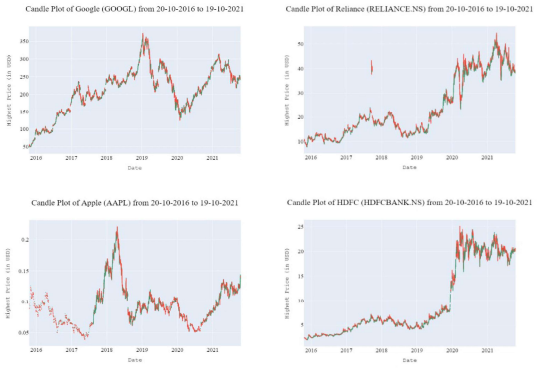

Fig. 2. Candle plots for the stocks considered Apple Inc. (AAPL), Google (GOOGL), HDFC Bank Limited (HDFCBANK.NS), and Reliance Industries Limited (RELIANCE.NS)

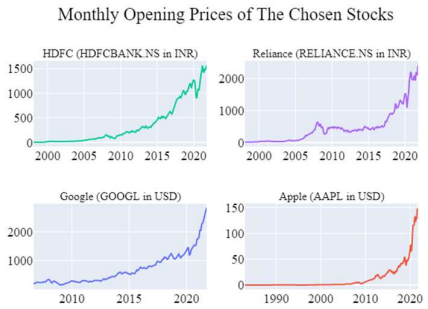

Fig. 3. Opening Prices of the considered stocks for the first trading day of each month

We split the monthly data (shown in Table 1, Plotted in Fig. 1 as candle plots and Fig. 2 as Line Curves) into training and testing data to train and evaluate the model. The split data was normalized separately to prevent any data leak. We normalized the data to values between -1 and 1 using min-max scaling, to ensure that no feature is falsely prioritized based on its value.

Normalization replaces the value in each column with the following formula:

$$m = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{2}$$

In formula 2, m = new cell value, x = initial cell value, x_{max} = maximum column value, x_{min} = minimum column value. Fig. 4. Shows the plot of normalized mothly opening prices.

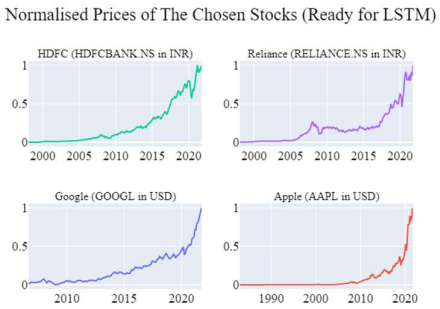

Fig. 4. Normalized Monthly Opening Prices

4 Method

StockDL uses a Long Short Term Memory network. This study explains the two approaches of using the LSTM Network. The accuracy is tested against activation functions like hyperbolic tangent (tanh), Rectified Linear Unit (ReLU), Leaky Rectified Linear Unit (Leaky ReLU). The following flowchart represents the pipeline of predictions from stockDL.

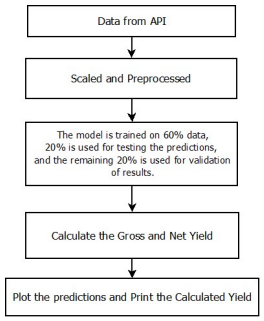

Fig. 5. Pipeline of stockDL

stockDL trains two models for comparisons with the baseline statistical models, the pure LSTM model is computationally expensive and slower than the Ensembled CNN and LSTM model.

4.1 LSTM Network

The preprocessed data is scaled and fed to the model's input layer. The following image represents the architecture of the LSTM model used in stockDL.

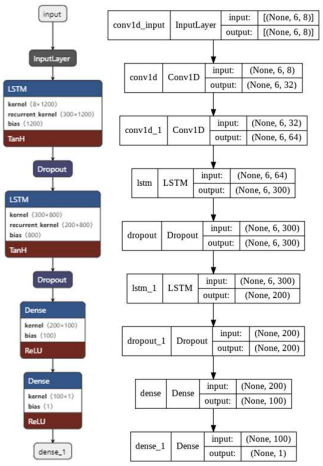

Fig. 6. StockDL Pure LSTM Architecture

The input later of the LSTM model (as shown in Fig. 7.) focuses on six data features: Date, Opening Price, Highest Trading Price, Lowest Trading Price, Closing Price, and the Volume Traded in a day. After each LSTM layer, a dropout layer is added, which drops 50% of the parameters. This step is added to ensure that the complexity of the model doesn't result in the model overfitting the data while it is trained. The model takes around 90 seconds to train daily data for the 6-year time frame and another 30 seconds for validation.

Various loss functions (Mean Squared Error, Quadratic Loss, L2 Loss) are tested for the model, and Mean Squared error loss comes out to be the best parameter for testing the divergence of the predictions from the actual value.

The formula for Mean Squared Error (MSE): $$\sum_{i=1}^{D}\left(x_i - y_i\right)^2 \quad (3)$$

4.2 Hybrid CNN-LSTM Model:

This is the novel approach introduced in this study which is ten times faster than the Pure LSTM model. The LSTM model takes time in understanding the pattern in the time-series data. To reduce the preliminary time taken by LSTM in pattern recognition, a layer of convolution neural network is employed for pattern recognition, and subsequent LSTM layers follow this layer for time series prediction, the result of the

prediction is further used in the pipeline to decide if the model should trade in or out of the market for the particular month to maximize the annual yield.

The following image shows the model architecture for the Hybrid CNN-LSTM model used by stockDL:

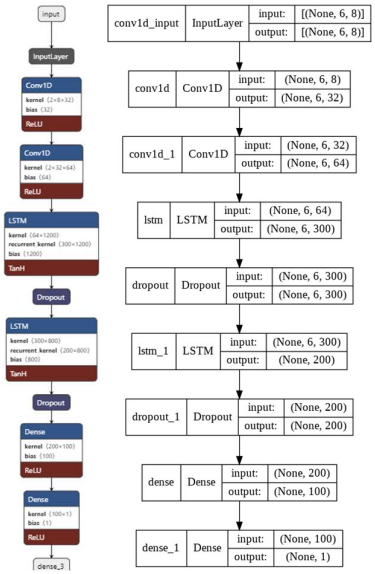

Fig. 7. Hybrid CNN-LSTM Model architecture of stockDL

The additional CNN layers are shown in Fig. 8., are added to the model to improve feature extraction and model training time. Dropout layers are added after the LSTM network to prevent overfitting. This model also uses the MSE loss function to check the actual and predicted value variation for providing better annual and gross yields.

5 Performance and Evaluations

5.1 Training Performance

The training time of training the LSTM model of stockDL is 90 seconds using NVIDIA K90 GPU on Google Colab, which is reduced to 30 seconds for the CNN-LSTM hybrid network. The following plot shows the training plot of the Hybrid CNN-LSTM Network.

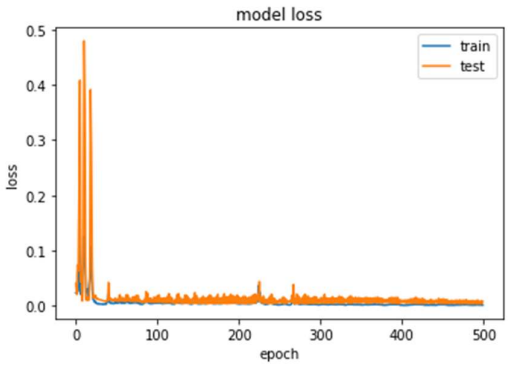

Fig. 8. Model loss for the Hybrid CNN-LSTM network

The training loss stabilizes at 0.0001, whereas the testing loss plateaus at 0.0003. The above plot (Fig. 9.) validates that the model isn't overfitting on the data, based on comparable training and testing loss. The following predictions plots of the model testify that stockDL is a novel, efficient and improved version of all existing LSTM Deep Learning Strategies that can be employed for the stock market or any regression-based time-series predictions.

5.2 Evaluations

The performance of the trained model is evaluated on four stock options as explained in the introduction of the study:

Table 3. Predicted net yield for each method used in predicting the trading strategy.

Method	***Predicted Net Yield (GOOGLE)***	***Predicted Net Yield (HDFC)***	***Predicted Net Yield (APPLE)***	***Predicted Net Yield (RELIANCE)***
Buy and Hold	23.25	20.25	20.25	28.42
Moving Average	10.9	10.9	20.9	21.14
LSTM	19.97	19.97	19.97	19.17
Hybrid CNN-LSTM	17.99	19.99	19.99	19.31

The methods employed for the predictions predicted a similar strategy for deciding In-Out trading months for the stocks of GOOGL, HDFCBANK.NS, AAPL, and RELIANCE.NS.

It can be inferred from Table 3. The Buy Hold Strategy, if incorporated, is predicted to provide the investor with the best yield for the stocks of Alphabet Inc. (GOOGL). For HDFC Bank, The Moving average strategy predicts the annual yield

to be 10.9%, whereas the Buy Hold, LSTM, and Hybrid CNN-LSTM have a comparable prediction with 202.5%, 19.97%, and 19.99%, respectively. The stocks of Apple Inc. provided similar Net yield from the four methods, with LSTM and Hybrid CNN-LSTM providing 19.97% and 19.99% respectively and the Buy and Hold, Moving Average providing comparable net yields of 20.25% and 20.90%, respectively. For Reliance (RELIANCE.NS) stocks, The LSTM and Hybrid CNN-LSTM Network provide similar net yields of 19.17% and 19.31%. The Buy Hold Strategy again predicts the maximum net yield of 28.42%.

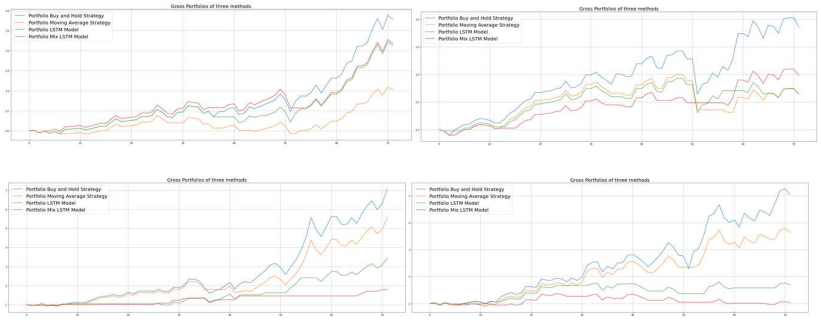

Fig. 9. Comparison of the predictions made by the strategies on the stocks of GOOGL (top left), HDFCBANK.NS (top right), AAPL (bottom left), RELIANCE.NS (bottom right)

From Fig. 9, both deep learning strategies have similar predictions of the net yield, and this provides the investors with a trade-off between minimizing the training time or maximizing the Net Yield from the stock. The Average Moving Strategy predicts 10.9% as net yield, the lowest compared to the other statistical method and the two deep learning methods.

6 Conclusion and Future Work

StockDL displays prominent and breakthrough results for efficient and accurate time-series predictions on historical stock data, which is evident with the comparisons of the Black-Box deep learning predictions with statistical baselines. Buy Hold Strategy appears to predict maximum net yield in almost all cases considered in this study, but this method is prone to market fluctuations, and this method relies on faith that the price will eventually rise in the long term. This assumption/faith can be risky for investors relying blindly on this strategy. The two Deep Learning strategies, LSTM and the Novel Hybrid CNN-LSTM, predict similar results. However, the Hybrid Architecture introduced in this study is computationally inexpensive and takes much less training and prediction time than the existing LSTM Architecture for time series prediction.

The findings of this study are interesting and surprising because of the accuracy of improved predictions with reduced/incomparable risks compared to the two manual statistical methods tested. Future work of this study can be implementing trading an

alternate stock in the month we trade out from the market to improve the yield. This Multi-Stock trading will help investors make better AI-supported decisions, reducing financial losses and improving the net yield. Another enhancement to stockDL or other such financial libraries can be done by developing a self-improving neural network architecture that learns from the gains or losses made by the past transaction and aims to run as humanly as possible to maximize the yield from investments on single or multiple stock portfolios.

References

1. G. Friede, T. Busch, and A. Bassen, "Journal of Sustainable Finance & Investment ESG and financial performance: aggregated evidence from more than 2000 empirical studies," J. Sustain. Financ. Invest., vol. 5, no. 4, pp. 210–233, 2015, doi: 10.1080/20430795.2015.1118917.
2. R. Shanmugham and K. Ramya, "Impact of Social Factors on Individual Investors' Trading Behaviour," Procedia Econ. Financ., vol. 2, pp. 237–246, Jan. 2012, doi: 10.1016/S2212-5671(12)00084-6.
3. Z. Chen and X. Du, "Study of stock prediction based on social network," Proc. - Soc. 2013, pp. 913–916, 2013, doi: 10.1109/SOCIALCOM.2013.141.
4. B. Prabowo, E. Rochmatulaili, Rusdiyanto, and E. Sulistyowati, "Corporate governance and its impact in company's stock price: case study," Utop. y Prax. Latinoam., vol. 25, no. Extra10, pp. 187–196, 2020, doi: 10.5281/ZENODO.4155459.
5. S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” Neural Comput., vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: 10.1162/NECO.1997.9.8.1735.
6. X.-Y. Liu et al., "FinRL: A Deep Reinforcement Learning Library for Automated Stock Trading in Quantitative Finance," Accessed: Oct. 20, 2021. [Online]. Available: https://github.com/AI4Finance-LLC/FinRL-Library.
7. H. Chen, P. De, Y. Hu, and B. H. Hwang, "Sentiment revealed in social media and its effect on the stock market," IEEE Work. Stat. Signal Process. Proc., pp. 25–28, 2011, doi: 10.1109/SSP.2011.5967675.
8. M. C. Lee, "Using support vector machine with a hybrid feature selection method to the stock trend prediction," Expert Syst. Appl., vol. 36, no. 8, pp. 10896–10904, Oct. 2009, doi: 10.1016/J.ESWA.2009.02.038.
9. T. Fischer and C. Krauss, "Deep learning with long short-term memory networks for financial market predictions," Eur. J. Oper. Res., vol. 270, no. 2, pp. 654–669, Oct. 2018, doi: 10.1016/J.EJOR.2017.11.054.
10. W. Bao and X. Liu, "Multi-Agent Deep Reinforcement Learning for Liquidation Strategy Analysis," Jun. 2019, Accessed: Oct. 20, 2021. [Online]. Available: https://arxiv.org/abs/1906.11046v1.
11. J. Shen and M. O. Shafiq, "Short-term stock market price trend prediction using a comprehensive deep learning system," J. Big Data 2020 71, vol. 7, no. 1, pp. 1–33, Aug. 2020, doi: 10.1186/S40537-020-00333-6.
12. F. Chollet, Deep Learning with Python. 2018.