Automated Stock Trading using Machine Learning Approaches

Abhishek Verma*

Department of Computer Science

New Jersey City University

Jersey City, NJ 07305

E-mail: averma@njcu.edu

*Corresponding Author

Ambar Ruiz

Department of Computer Science

New Jersey City University

Jersey City, NJ 07305

E-mail: ambarruiz12@gmail.com

Haresh Singh

Department of Computer Science

New Jersey City University

Jersey City, NJ 07305

E-mail: haresh.singh.97@gmail.com

Karim Elbahloul

Department of Computer Science

New Jersey City University

Jersey City, NJ 07305

E-mail: kelbahloul1994@gmail.com

Abstract: A new automated stock trading method based on deep learning approaches is presented to improve stock market prediction. The usage of machine learning methods applies regression and neural network models for more accurate detection and efficiency. Historical data will be fed and a prediction of the stock price and its movement will be outputted.

I. Introduction

Stock markets have become one of the biggest financial institutions of the United States economy. It allows big companies to pool capital from the investing public to assist in funding their growth and operations. Even though it may create a greater rate of return, there is still a large amount of risk involved. Due to the volatile nature of the stock market, large investment firms and/or private banks have attempted to create some sort of strategy to predicting the stock market. With such information, it would help to advise buy and sell decisions with volatile fluctuations. With the development of artificial intelligence in the modern world, this allowed the use of computers in assisting with the task of predicting the market with the use of statistical methods in data analysis [1]. The study of machine learning can be looked upon as creating a computer program or system that can inherit data from a past or streaming source and learn from it. Traditional approaches to machine learning have often focused upon performance improvement. It uses various techniques to attempt to extract the most promising features from the dataset. One of the most popular approaches is the use of support vector machines. Support vectors are kernel-based machine learning algorithms that uses supervised learning to create a prediction. Even though it embraces the topic, they are not as effective as neural networks, especially deep learning neural networks. They are very powerful because they make use of multiple layers in order to interpret and determine an output of prediction.

Once an accurate model has been produced from the existing data, future users will be able to feed various time periods of stock price data and be able to receive an output as a prediction for the next few days, months or even years depending on the parameters given. With this information, a short-term prediction can be assessed by stakeholders such as investors, financial advisors, and investment bankers. Our objective of this project was to do a time-series analysis and attempt to show a predicted price and price movement for the next year. For this research, we selected the SPDR Standard & Poor's 500 ETF (SPY) stock, that is publicly available on Yahoo Finance [2].

II. Related Work

A. *Automation of Stock Market*

Traders and/or investors of the stock market attempt to use systems from financial institutions to automate a prediction of buy sell prices within a given time period. These methods are quite advantageous as they bring assistance to investment decision making. This benefits the normal investor as it provides the advantages of better response times to market changes, more accurate trading operations, and reduced risk of loss due to repeated or mistaken operations [4].

B. Logistic Regression

Logistic Regression is one of the most common regression model used for machine learning approaches. As a predictive model, it is quite similar to linear regression, except that the output is dichotomous. In the case of stock price prediction, it can be used to help determine whether to buy or sell the stock at a given point in time. If the closing price the day after were higher than the present, it would result in an output 1, which predicts a good buy. If a -1 output is produced, the model would predict that it may be best to sell.

C. Support Vector Regression

While doing analysis on the historical data of a stock's price, it is also important to study the temporal dependency. In [3], the authors attempted to use a machine learning approach called stock's time-series data.

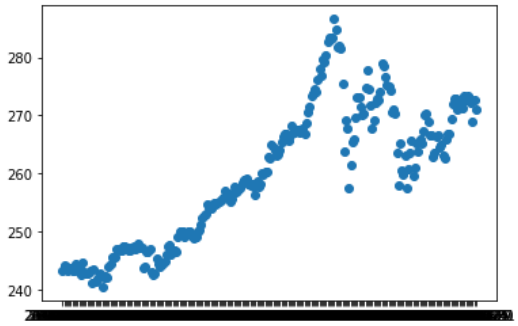


Figure 1. This scatter plot shows daily closing price of ticker symbol SPY from the time period of June 1, 2017 to May 31, 2018 [2]

TABLE I

RESULTS FROM FOUR BEST PERFORMING SUPPORT VECTOR REGRESSION MODELS ON SPY DATASET

Kernel	Mean Squared Error	Accuracy
linear	0.006459	0.99361
linear	0.007439	0.99468
linear	0.007836	0.99282
linear	0.071894	0.99584

Investors and traders today have developed methods to predict future stock prices. One method is using the business value of a stock. Factors such as assets, operating expenses, management and revenue. That information is always readily available on Yahoo Finance [2]. Another study that is very popular is the usage of technical analysis. Its popularity is assumed by the belief that the market may move in cycles. Such indicators such as RSI, are used based on this belief due to the fluctuation of price data and the volume. With modern day advancements in artificial intelligence, larger datasets can be analyzed [4].

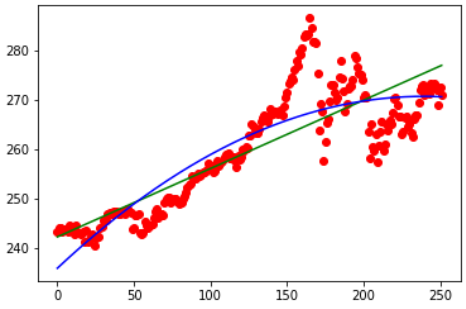


Figure 2. This graph displays a scatter plot of the S&P 500's closing prices. The green line represents a line of linear regression. The blue line represents a line of polynomial regression with a degree of 2.

D. Random Forest

It is evident that the stock market's price movement is entirely random and volatile, with fluctuations that are infeasible to avoid. Decision Tree algorithms in machine learning however, are common to help build classification and regression and help overcome the obstacles of predicting the stock market. But these algorithms come with drawbacks of their own. Overfitting is the main problem with decision trees because as the machine makes decisions to come up with a definite conclusion, it is at risk of increasing the specificity within the tree and making it indefinitely complex. Fortunately, random forest helps to overcome the drawbacks of decision trees. The random forest algorithm is made up of many decision trees that attacks the overfitting. The accuracy of this this ensemble model is much greater than one of an individual model with an average of all

TABLE 2

RESULTS FROM CLASSIFICATION REPORT METRICS ON THE SPY DATASET

	precision	recall	f1-score	support
-1	0.84	0.88	0.86	608
1	0.89	0.86	0.87	713
micro avg	0.86	0.86	0.86	1321
macro avg	0.86	0.87	0.86	1321
weighted avg	0.87	0.86	0.86	1321

outputs being considered the final output. That is why Random Forest will be one of the choices to help build our predictive model. The robustness of the model is evaluated by calculating different parameters such as accuracy, specificity as well as precision. Using the predicted result, it can be decided whether to buy or sell stock. [12]

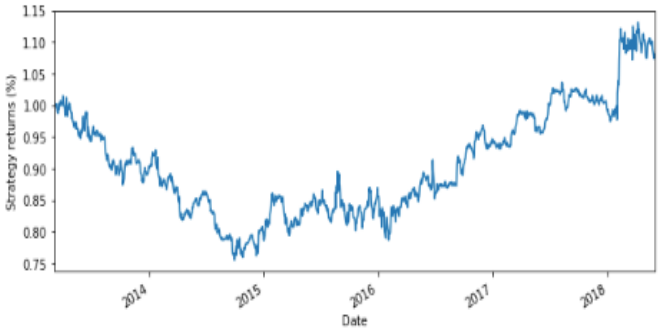


Figure 3. The output displays the strategy returns and daily returns according to the Random Forest Classifier.

III. Description of Dataset

The SPDR Standard & Poor's 500 Trust (SPY) is an exchange-traded fund which trades on the NYSE under the symbol. It is designed to track the S&P 500. It is a market-capitalization-weighted index of the 500 U.S. largest publicly traded companies by market value. This fund is the largest ETF and most popular to be publicly traded. This gives it an excellent choice due to measurement of the overall performance of the stock market.

The SPY dataset is publicly available on Yahoo Finance [2]. It consists of historical data primarily of opening, closing, adjusting and other parameters that may be useful. Our data consists of daily closing prices from June 1, 2017 to May 31, 2018. It contains daily data for five

weeks for the following year. The dataset contains of opening, closing, high and low prices as well as daily volume traded. The reason why this particular ETF was chosen was because of the strong stability. With traditional company stocks, high volatility is present causing unpredictability in the moving price.

A second dataset was also used for validation purposes. A range of 21 years of the S&P500 ETF (SPY) was derived from Yahoo Finance[2]. As regards to the technical indicators for the prediction model, the following were included: correlation, the difference between the opening, closing prices, daily highs and lows of various days. The 10-day moving average was used a technical indicator to show a simple movement of the stock before applying any implementation of machine learning.

IV. Experimental Environment

The model was created using the programming language Python. While programming, we have used different libraries for different functionalities:

- Pandas [6] was used for importing the dataset, as well as data manipulation.
- Matplotlib [7] was used for plotting graphs for the usage of time-series analysis as well as predicting trends.
- NumPy [8] was used to perform matrix operation upon data. Matrix operations such as reshape were used in preprocessing the data.
- Scikit-Learn [9] was used to prep data by preprocessing so it may be used to find the best parameters. In older versions, 5 cross validation was used. Because of updated software, train test split was used through model selection.
- Keras [10] was used for creating and training the neural network. It is a popular open-source for researchers because it is quite easy to define and train a deep learning model in a short amount of time. It creates a front-end interface to other existing libraries such Tensorflow [11]. It is a great library because it allows the Graphical Processing Unit(GPU) to be used for faster training and prediction.
- Tensorflow [11] is the backend for the library Keras. It is developed and maintained by Google. It was primarily used for assistance in the creation, training and predicting of the convolutional neural network.
- Prophet [14] implements a procedure for forecasting time series data based on an additive model where non-linear trends are fit with yearly, weekly, and daily seasonality, plus holiday effects. It works best with time series that have strong seasonal effects and several seasons of historical data. Prophet is robust to missing data and shifts in the trend,
and typically handles outliers well.

V. Research Results and Analysis

A. *Logistic Regression*

The dataset was split into training (75%) and test (25%) dataset. After fitting these into the model, the decision whether to buy (1) or sell (-1) was predicted. Finally, the model was evaluated to create a stock trading strategy.

Figure 4: This graph represents the prediction output of the Boost results. The blue line represents the actual SPY data. The orange line represents the XG Boost's prediction upon interpreting SPY

The 75-25 split model used showed an accuracy measure of 78.59% and a Mean Squared Error (M SE) of 0.86. The accuracy, which is close to 80%, shows that the model is a relatively good predictor for future stock trading while getting a MSE less than one is an acceptable value.

B. *Support Vector Regression Model Results*

Results from the SVR Model are shown in Figure 5. Cross validation as well as grid search was done to select best kernel values. Mean squared error and accuracy was used as performance metric. Higher value is considered better for accuracy. Lower value is considered for mean squared error. From the list, of
kernels, linear was performed through grid search as one of the best models. Linear and polynomial kernels are beneficial as they create a simple predictive trend that is easy to understand.

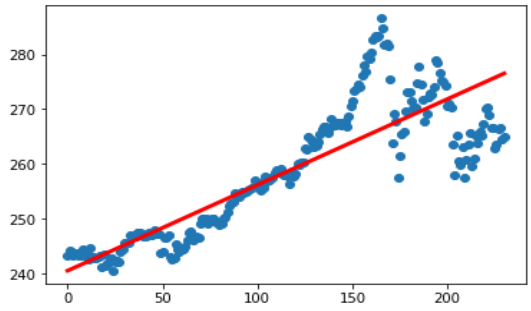


Figure 5. Support Vector Regression predictions of the dataset based by on ticker symbol SPY. The X axis shows the time segment for the period time and prediction. Y axis shows the closing price for each day. Data comes from 2017 and 2018.

C. *Random Forest*

After our data was acquired and smoothed, the random forest will be ready to be trained. The smoothing will be done to remove random variation and noise allowing the model to identify long term price trend in the behavior of the stock market. The input and output variables were created. Input: standard deviation of last 2 days returns and average of the last 2 days returns. Output: 1 indicates to buy the stock and -1

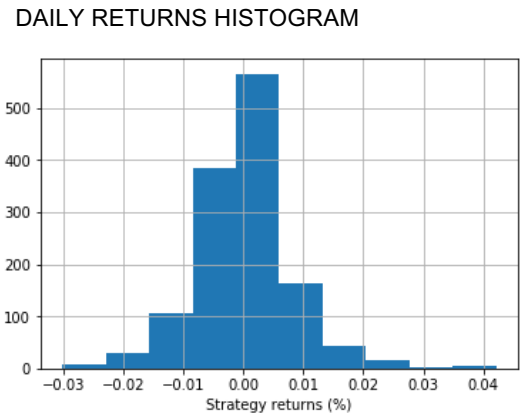


Figure 6. This graph is the output using the Random Forest classifier. With relation to the neural network, a range of returns were found and counted throughout the dataset.

indicates to sell stock. The choices of input and output are completely random with feature selection. The dataset was then split: 75% for training and 25% for testing. [12] When training the machine learning model, the accuracy score, or correct prediction was 86.44%.

D. *XG Boost Results*

After results from various regression were posted, XG Boost was used to verify the results. Before using this method, the data had to be pre-processed to where a training and a testing dataset will be created. Various splits were done in order to create a better prediction. Two parameters were used to test the precision of the prediction. Accuracy was used as a simple method to test how accurate the data using a percentile format. Mean Squared Error was also used to test proximity in how far away the prediction output was in comparison with the actual price.

After testing various splits, the most accurate one was a 75:25 percentage split upon the dataset. Using the 75% training dataset for both Random Forest and XGBoost, the Mean Squared Error (MSE) results showed a lower value for the Random Forest (1.93) versus the XGBoost (2.73).

VI. Conclusion and Future Work

In this dissertation, a proposal was done to create a new deep learning model that combined the powers of Convolutional Neural Networks (CNN) and LSTM for data analysis. Our model attempted to improve the accuracy of a stock market prediction. This was used a create a better model than using popular methods such a regression and perceptron models. Data variations was also presented to prove how our accuracy increased.

Though regression techniques are used popularly to create a simple prediction, further research would apply deep learning techniques to further explore directional stock predictions. An addition of sentiment analysis would create higher accuracy as trending news would be useful in create a more accurate prediction. With specific keywords streaming from the internet, positive and negative keywords may be identified and used as an assumption towards the current value of the stock.

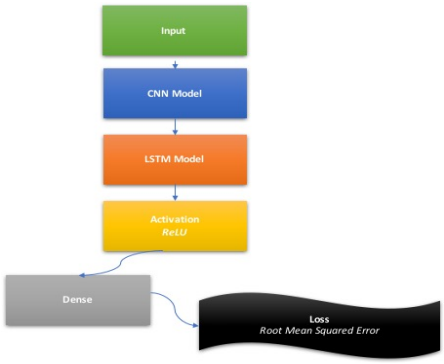
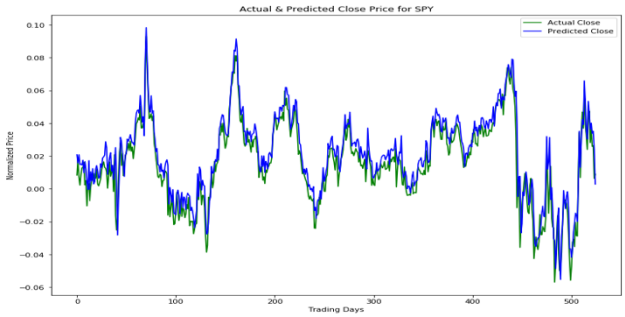


Figure 7 & 8

This displays our CNN+LSTM model using a single pipeline. A graph is outputted to display how accurate the predicted price vs the actual price.



References

[1] J. Hall, G. Mani, and D. Barr, "Applying Computational Intelligence to the Investment Process", in Proc. of 1996 CIFER: Computational Intelligence in Financial Engineering.

[2] “SPDR S&P 500 (SPY) Stock Historical Prices & Data.” *Yahoo Finance*. [Online]] https://finance.yahoo.com/quote/SPY/history?p=SPY

[3] P. Meesad and R.I Rasel, "Predicting Stock Market Price Using Support Vector Regression," in Proc. of Int. Conf. Informatics, Electron. and Vision, (ICIEV), 2013. doi: 10.1109/ICIEV.2013.6572570.

[4] J. Eapen, A. Verma, and D. Bein, "Novel Deep Learning Model with CNN and Bi-Directional LSTM for Improved Stock Market Index Prediction," *the 9th IEEE Annual Computing and Communication Workshop and Conference (CCWC)*, Jan. 7-9, 2019, Las Vegas, NV, USA.

[5]A. Verma, C. Liu, and J. Jia, "Iris Recognition based on Robust Iris Segmentation and Image Enhancement," *International Journal of Biometrics*, vol. 4, no. 1, pp. 56-76, 2012

[6]NA, “Python Data Analysis Library¶.” 2019[Online]. *Pandas*. [Accessed July 11, 2019] https://pandas.pydata.org/.

[7] “Installation¶.” 2019. *Matplotlib*. [Accessed July 11, 2019]. https://matplotlib.org/.[Online]

[8]“NumPy¶.” 2019.[Online] *NumPy*.. https:// www.numpy.org/. [Accessed July 11, 2019]

[9] “Learn.” 2019. *Scikit*.[Online]. https://scikit-learn.org/stable/. [Accessed July 11, 2019]

[10] “Keras: The Python Deep Learning Library.” 2019[Online]. *Home - Keras Documentation*.. https://keras.io/. [Accessed July 11, 2019]

[11] “TensorFlow.” 2019. *TensorFlow*.[Accessed July 11]. https://www.tensorflow.org/.[Online]

[12] Khaidem, Luckyson & Saha, Snehanshu & Basak, Suryoday & Kar, Saibal & Dey, Sudeepa. (2016). Predicting the direction of stock market prices using random forest.

[13] “Random Forest Algorithm In Trading Using Python” [Online]. https://blog.quantinsti.com/ random-forest-algorithm-in-python/. [Accessed 5 Jul. 2019].

[14] “FbProphet.” 2019. [Online] https:// pypi.org/project/fbprophet/ [Accessed July 11, 2019]