MACHINE LEARNING AND TECHNICAL ANALYSIS FOR STOCK MARKET PREDICTION

Date: 23rd December 2018

Team: Sharkdodo

Team Members:

1. **Fyeeza Fyruz**
   ID:1510152042
   Section: 02
2. **Mohammad Sadman Islam**
   ID:1510151642
   Section: 02
3. **Mohammad Rashedul Alam**
   ID:1511384642
   Section: 02
4. **Shegufta Zaman**
   ID:1511146042
   Section: 02

MACHINE LEARNING AND TECHNICAL ANALYSIS FOR STOCK MARKET PREDICTION

Fyeeza Fyruz
1510152042
North South University
Dhaka, Bangladesh
ffyeeza.987@gmail.com

Mohammad Sadman Islam
1510151642
North South University
Dhaka, Bangladesh
sadman.islam@northsouth.edu

Mohammad Rashedul Alam
1511384642
North South University
Dhaka, Bangladesh
rashedul.alam@northsouth.edu

Shegufta Zaman
1511146042
North South University
Dhaka, Bangladesh
shegufta.zaman@northsouth.edu

***Abstract*— Providing opportunities for investment and capital acquisition, capital markets, namely the Dhaka Stock Exchange (DSE) and Chittagong Stock Exchange (CSE) have made a significant impact on the economy. In recent times, unpredictability in the share market has come to be a serious concern for investors. In this paper, we aim to predict and analyze the stock market so as to gain a more intuitive understanding of the pattern and thus make more profitable investments.**

Keywords—stock, market, prediction, machine learning, deep learning ,RNN,ARIMA,MSE

I. The Problem

Financial services and banking industries employ analysts dedicated over attempting to quantify data from news and daily exchange reports to predict stock price fluctuations happening the next day. The vast amount of quantitative data means that it is nearly impossible for an analyst to cover more than just a few companies at once.

As there are daily ups and downs of the market, there must be patterns that may aid analysts and data scientists to predict the stock market better and avoid falling into pitfalls.

Small investors generally rely on trends in data and news from sources with varying levels of legitimacy when trading, with very little credible data that may help aid their decisions. Forecasting stock prices should theoretically give investors and analysts to work better with the random-walk behavior of stock prices.

II. Motivation For Our Choice

In December 2010, the Dhaka Stock Exchange crashed to record lows. Initially, the General Index or DGEN of Dhaka Stock Exchange reached record highs. However, in a short span of time, the bubble created by the record highs burst, showcasing the biggest fall in a day ever recorded in its history. The market became stale with no positive movement in stock prices.

A study [4] identified four psychologies of local investors. These traits specifically portrayed greed, envy, speculation, and overconfidence that contributed to the formation of the huge bubble, while on the other hand, four loss-minimizing traits such as panic, frustration, lack of self-confidence, and distrust caused the bubble to burst concurrently. The bankers, brokers and manipulators were the biggest gainers, whereas the most unaware and greedy small investors encountered heavy loss. It is believed that the market will be less volatile, more mature, and sustainable when most of the investors will be conscious about the potential risks and returns

This research intends on making information more visible to the domestic investor to aid in stock price forecast, taking into consideration factors that may have skipped mind due to the aforementioned human psychology issues.

III. Background

Previous work on stock market forecasting has made use of linear models such as Auto Regression (AR), Auto Regressive Moving Average (ARMA) and Auto Regressive Integrated Moving Average (ARIMA). However, the limitations of these models were that they were specific to one company i.e. it only performed well for the company whose dataset it was trained by.

For multiple decades, deep neural networks also referred to as Artificial Neural Network (ANN), have been utilized for predicting stock prices. ANNs have proven successful because of their ability to examine input and output relationships even in cases where the dataset is complex. Another useful trait of ANN is that they are able to identify newly introduced test samples post training. Iterative and directive methods have been tested for forecasting by Coskun Hamzacebi [1]. ANN was also formally applied to predict NASDAQ's stock value given input parameter of stock market by Yunus Yetis [2]. Neural networks have been applied to technical analysis as a prediction model in the works of Mizuno [3].

In most cases ANNs suffer from over-fitting problems due to the large number of parameters to fix, and the little prior user knowledge on the relevance of the inputs provide above-average results. Support vector machines (SVMs) had been developed as an alternative that avoids such limitations.

Choosing the best features for any machine learning problem is quite difficult. For extracting best features 8 different techniques was used in a paper by Ganesh[5]. NN. SMO, Bagging using SMO and M5P were used to test these features. "Volume + Company" and "Nasdaq + S & P + Company" were the two best features that outperformed the others.

News effect the change in stock prices significantly. In a paper by Xiao [6] they extracted events from news text. The events were structured using NLP and OpenIE. EB-CNN gave 65.8% accuracy predicting the index.

Previous work shows that predicting prices based solely on past prices and volumes did not give desirably accurate outcomes because stock prices were also influenced by external factors such as news about the stock market and the company. Thus, to improve accuracy, technical indicators were combined with machine learning by Alice Zhen, Jack Jin from Stanford University [7].

Through heavy testing Amin Hedayati Moghaddama, Moein Hedayati Moghaddamb and Morteza Esfandyaric have found that artificial neural networks with four or more hidden layers were unable to generate a robust model and accurately predict outcomes. Their studies how that networks with two to 3 hidden layers and 40 to 70 neurons in the hidden layers functioned best in predicting appropriate outputs [8].

IV. Description of Data and Data Source

Our model will be trained using dataset populated by data directly from the official website of Dhaka Stock Exchange Ltd (https://www.dsebd.org). Aside from that, another dataset consisting of stock prices from the Microsoft Corporation was used as a quality benchmark.

We will primarily be working with information like Opening Price, Highest Price of the Day, Lowest Price of the Day, Closing Price, Volume Traded, Difference between High and Close.

For the feature selection we will extract the day-wise closing price of each stock because that is the feature that holds most relevance when it comes to the decision-making process for the investors.

V. Data Preprocessing

Feature scaling will be done via normalization to unify the range of all the features in the training set and test set. Data may contain gaps and therefore, previous prices will be carried over to fill the gaps.

Because the features in our dataset exhibit many diverse ranges, we will need to do feature scaling by normalizing each data before we feed it into our algorithm so that all out data belong to the same range and is, thus, easier to process.

$$x\_normalized = (x - x\_min) / (x\_max - x\_min)$$

For now, we have worked on American stock market dataset obtained from Kaggle. This Dataset contains Date, Open, High, Close, Volume, OpenInt price for each company in a csv file. We have created three new columns of diff_openclose, diff_highclose and inc_volume.

Below we have shown the visualization for current two most active stocks on NASDAQ: Advanced Micro Devices, Inc.(AMD) and Invesco QQQ Trust

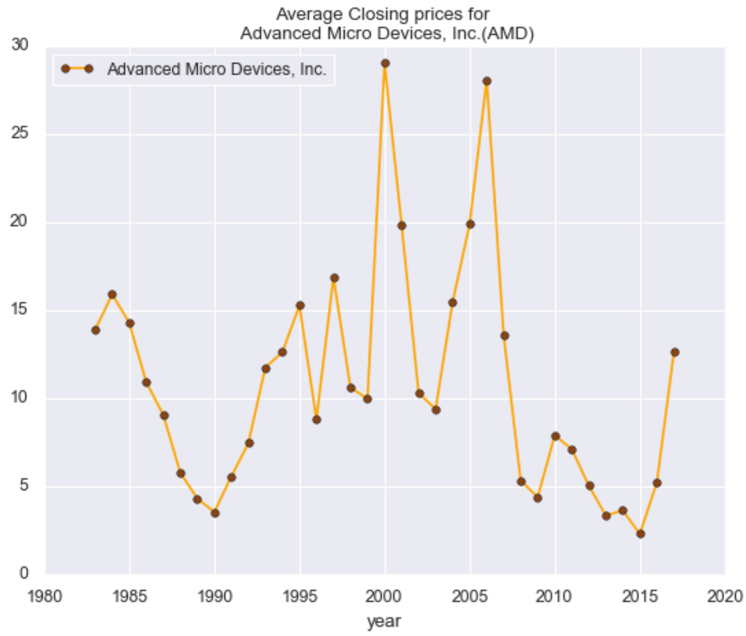

Fig 1: Average Closing Price of AMD

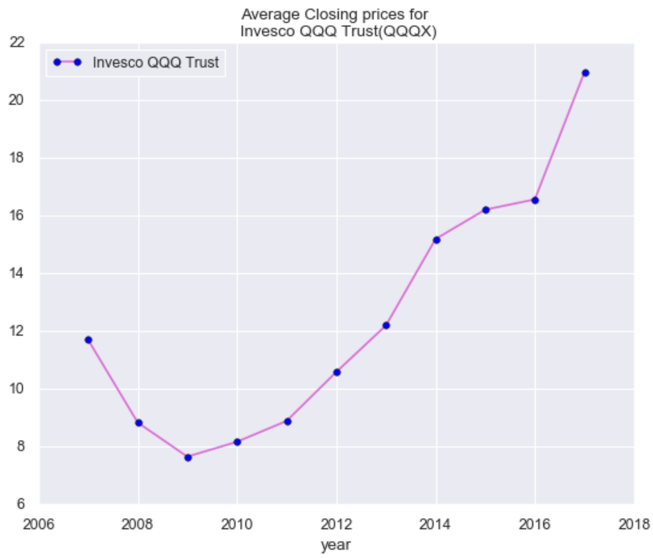

Fig 2: Average Closing Price of Invesco QQQ

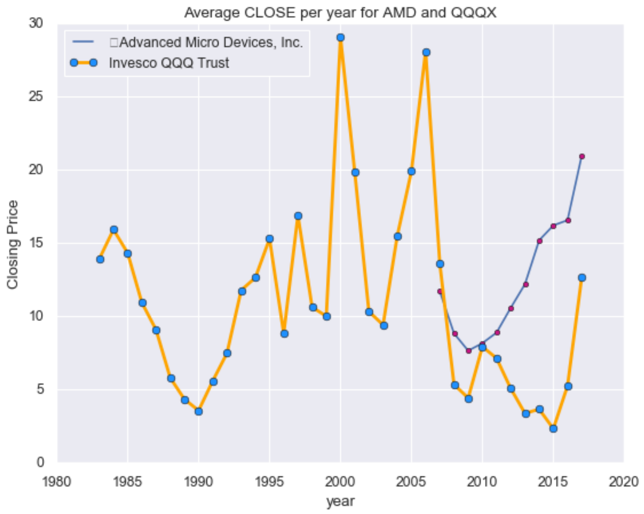

Fig 3: Average Close Per year of AMD and QQQX

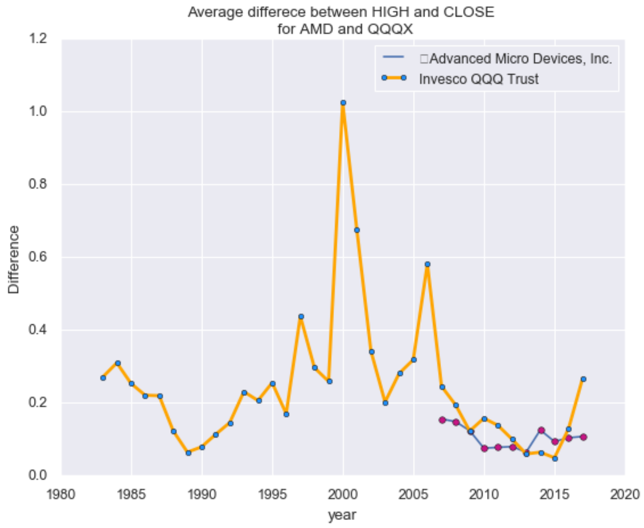

Fig 4: Average difference between high and close

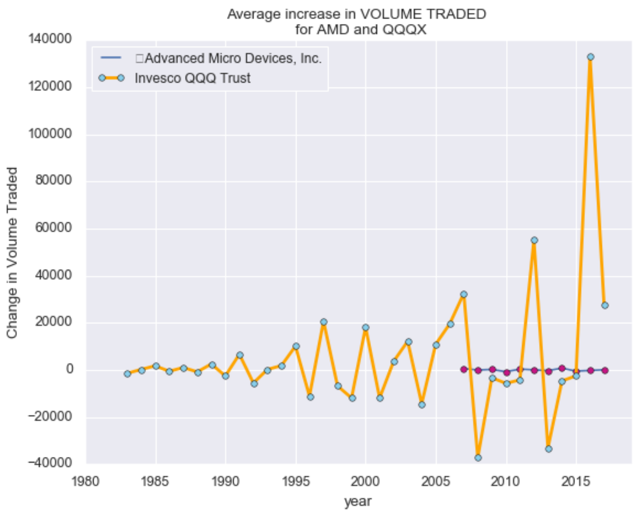

Figure 5: Average increase in Volume traded

VI. ALGORITHMS

For our first experiment, we have applied Multilayer Perceptron (MLP) to our dataset. In the first experimental design we implemented a model to see how well the predictions correlated with the headlines. For that we implemented a MLP and made use of the nltk.sentiment package to analyze the sentiments of the headlines so as to gain insight on whether each headline was positive, negative or neutral. The SentimentIntensityAnalyzer module has four sentiment metrics: Positive, Negative, Neutral and Compound. The module analyses a sentence and sets each metric accordingly. The compound metric is the sum of all the lexicons normalized to one. Looking at the compound metric we or our function can judge how positive or negative a certain sentence is.

For our dataset, we used the stock prices for Microsoft Corporation primarily to find an appropriate dataset consisting of news headlines. However, there was no dataset containing the headlines relevant to just Microsoft. As a result we had to use a dataset consisting of all headlines from The New York Times. As the dataset contained headlines of various world topics, our model failed to show acceptable results.

The perform metrics used to evaluate our design were the Mean Squared Error and the Coefficient of Determination (R^2). Our first implementation had values Mean Squared Error (MSE): 0.846 and R^2 Score: -4.7826 suggesting the model is clearly unacceptable.

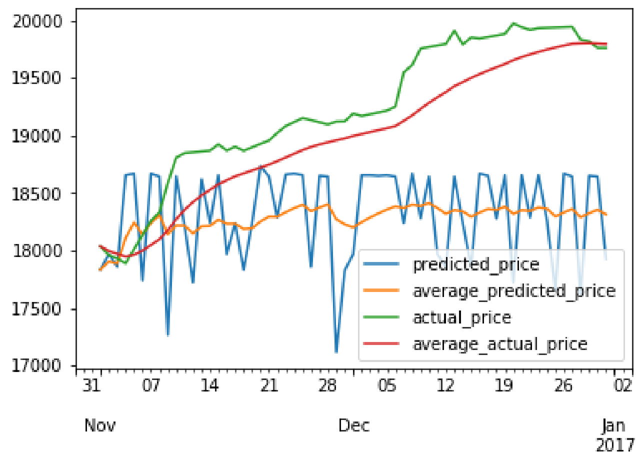

Fig 6: Predicted and Actual Comparison

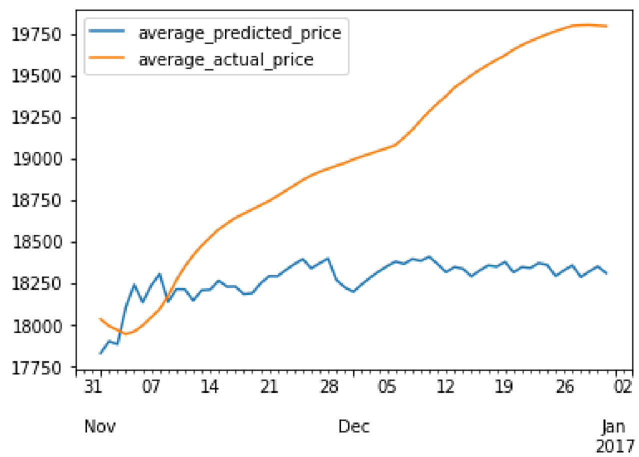

Fig 7: Results from MLP using Sentiment Analyzer

For our second design, we implemented an MLP without using sentiment analysis. Our neural network took 5 inputs - open, high, low, volume, difference_between_high_close.
At first shot the model showed very poor performance due to model over-fitting and excessive count for features. To improve that, we firstly dropped two features - volume, difference_between_high_close – because they provided least significance to the output. After that, the cross validated testing and training inputs were normalized using sklearn.preprocessing package. Once we had our data preprocessed and ready to use, we ran a number of experimental tests, this time to find the optimal values for the number of hidden layers, number of neurons in each hidden layer, the regularization parameter and the learning rate.

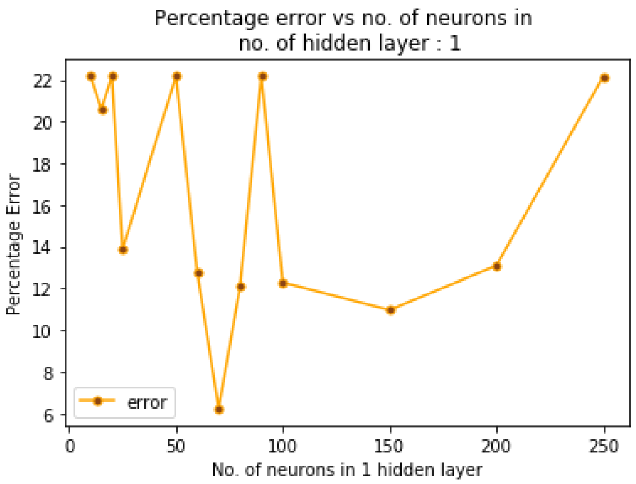

Fig 8: Percentage error vs no. of neuron in 1 hidden layer

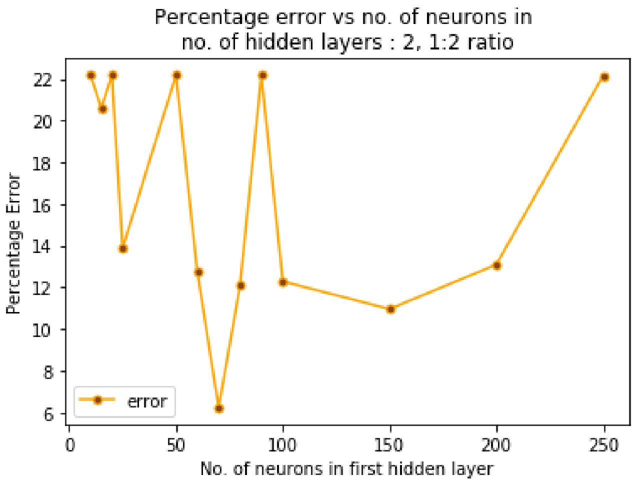

Fig 9: Percentage error vs no. of neuron in 2 hidden layers

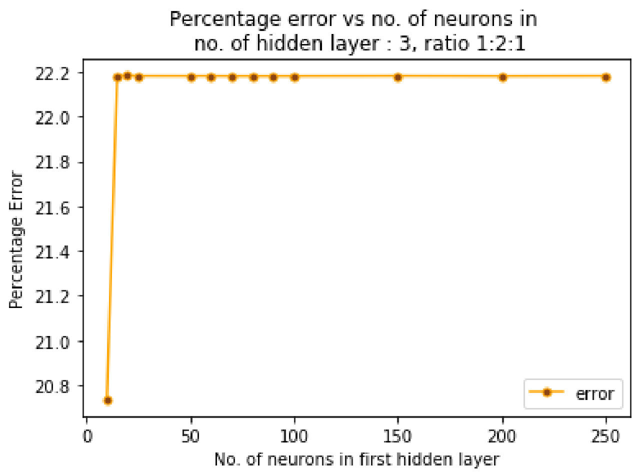

Fig 10: Percentage error vs no. of neuron in 3 hidden layers

For our dataset it was found that a MLP with two hidden layers with a neuron distribution of 60,120 proved to output the best results with the regularization parameter and learning rate set to 0.004 and 0.001 respectively. We used logistic as our activation function which performed much better than the rectifier (ReLu).

$$\phi(z) = \frac{1}{1 + e^{-z}}$$

After completion our model had a Mean Squared Error (MSE) of 0.1475 and an R^2 Score of -1.7826.

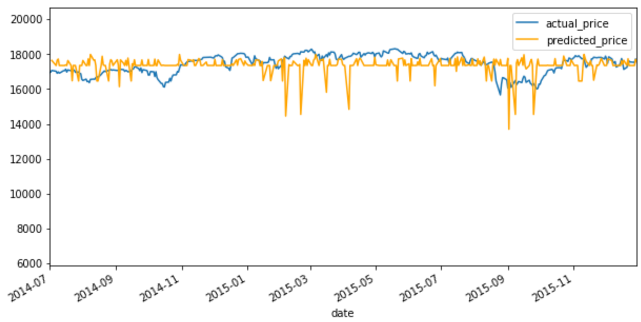

Fig 11: Output of model using MLP

Secondly, we applied RNN (Recurrent Neural Network) which is an Artificial Neural Network where connection between nodes form a directed graph along a sequence. RNN works best for sequential cases which depends on previous results. As stock market prediction is heavily depended on previous days features and closing prices.
The model is a sequential model. The Sequential model is a linear stack of layers. We used GRU (Gated Recurrent Unit) with 512 units. Adam optimizer and mse loss function is used in this model. We also used sigmoid activation function here.

$$f(t) = \frac{1}{1+e^{-t}}$$

We ran 500 epochs with batch size of 250. We got Mean Squared Error of 0.955.

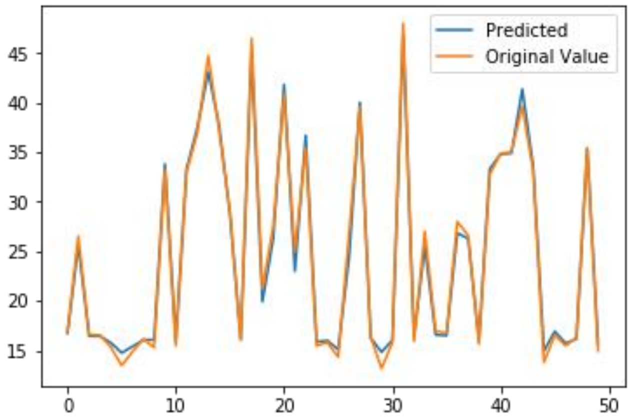

Fig 12: Output of model using ARMA algorithm

As expected RNN produced much accurate prediction for stock prices.

Lastly, we applied the Auto Regressive Moving Average (ARMA), which is a linear model, to our dataset. We normalized and cross validated the data in the same way as all of our prior experiments so that we are able to make a fair comparison as to which algorithm performs best.

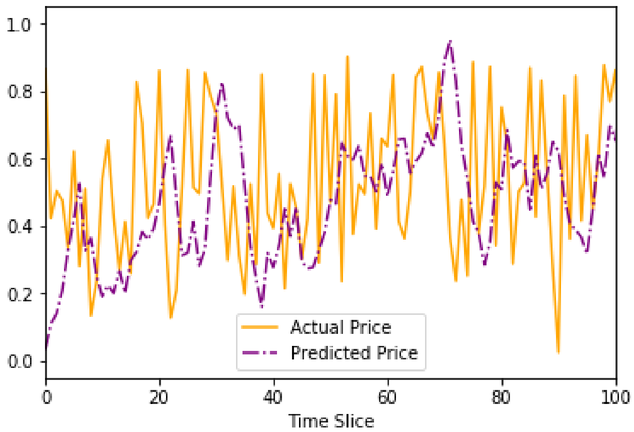

Fig 13: Output of model using ARMA algorithm

Our linear ARMA model outputted a very low MSE of 0.087 but a negative R^2 of -0.64393, which is still an improvement over our MLP model which had an R^2 of -1.7826.

The low MSE and negative R^2 means that given our dataset, a null model, or a horizontal line with the equation y = mean(outputs) would be a *better* fit and thus a better model to predict the outcomes.

References

[1] Hamzaebi C., Akay D. and Kutay F. (2009). ”Comparison of direct and iterative artificial neural network forecast approaches in multi-periodic time series forecasting.” Expert Systems with Applications 36 (2): 3839-3844

[2] Yetis Y., Kaplan H., and Jamshidi M. (2014). ”Stock market prediction by using artificial neural network.” In World Automation Congress (WAC) :718-722

[3] Mizuno H., Kosaka M., Yajima H. and Komoda N. (1998). ”Application of neural network to technical analysis of stock market predic- tion.”Studies in Informatic and control 7 (3): 111-120.

[4] Md. Tahidur Rahman, Syed Zabid Hossain, Md. Habibullah. Stock Market Crash in Bangladesh: The Moneymaking Psychology of Domestic Investors. American Journal of Theoretical and Applied Business. Vol. 3, No. 3, 2017, pp. 43-53. doi: 10.11648/j.ajtab.20170303.12

[5] Ganesh Bonde ,Rasheed Khaled, “Extracting the best features for predicting stock prices using machine learning”

[6] Xiao Dingy, Yue Zhangz, Ting Liuy, Junwen Duan, ”Deep Learning for Event-Driven Stock Prediction”

[7] Amin Hedayati Moghaddama, Moein Hedayati Moghaddamb, Morteza Esfandyaric, “Stock Market Index Prediction Using Artificial Neural Network”

[8] Alice Zheng, Stanford University, Jack Jin, Stanford University, “Using AI to Make Predictions on Stock Market”