Stock Market Trend Prediction using Supervised Learning

Asad Masood Khattak, Habib Ullah, Hassan Ali Khalid, Ammara Habib, Muhammad Zubair Asghar, Fazal Masud Kundi

2019

Presented In Proceedings of the 10th International Symposium on Information and Communication Technology

This report is presented as a survey of a previous work [1].

Jason Ellerbeck

ITCS – 5156 Spring 2023 – Desai

May 4, 2023

1 Introduction

1.1 Problem Statement

The stock markets of today allows financial professionals to invest into companies by trading and selling stocks. In this trading, there is money to be made. Professionals spend countless years of their lives perfecting the craft, memorizing trends, and taking risks. As someone who is interested in the stock market but has no training or knowledge of it, I would also like to make money trading stocks. The following report determines if easy to understand data can used by machine learning algorithms to predict stock trends.

1.2 Motivation

Ever since I started computer science, I imaged software that would be able to trade for me automatically on the stock market. By simply running a program, stock trends could be analyzed and predicted, creating a hands-free system that could make money without a user. I believe that any complex ideas or problems can be broken down into data to find a cause-and-effect relationship that a computer could use. While I could simply only imagine a program capable of this when beginning programming, machine learning makes this fantasy obtainable.

1.3 Questions of Domain

There is a countless amount of researching that has gone into utilizing machine learning to predict the stock market. However, a considerable sum of this research utilizes complex financial data in which a high degree of knowledge is required to understand it. This is no surprise as the stock market can be volatile and difficult to predict so using the best data available leads to better results. In this, I have never studied finance or learned about the stock market, making this endeavor more difficult.

That leads to the main question of this report: Can stock market trends be accurately predicted with easy-to-understand data? Features that are used within the model are the stocks daily open and close price, the daily high and low of the stock, and the volume of the stock traded that day which are fed into the model to predict the target of stock return [1]. Differing from other research, this research uses data that is easy to understand.

1.4 Approach

"*Stock Market Trend Prediction using Supervised Learning*" uses a K-Nearest Neighbor algorithm by breaking the problem down into the following three steps: Data Collection, Data Normalization, and Apply KNN [1]. As seen, two of these steps focus on collecting the data for the model along with normalizing it. When reproducing the report, these two steps took the most time and had the most impact on the accuracy of the model. The data normalization section was where the data was filtered, and the necessary target value was created.

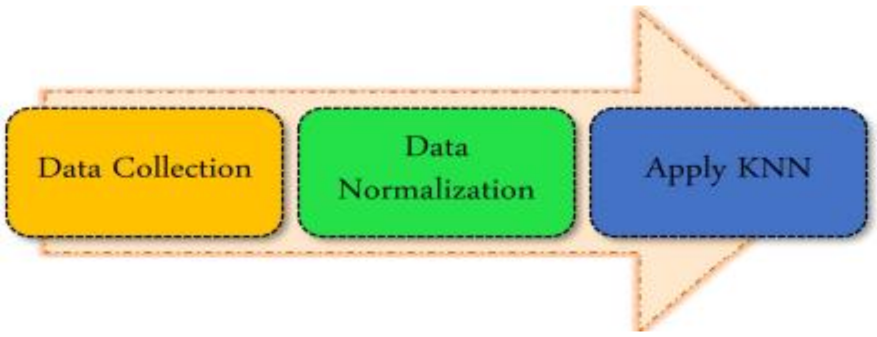


Fig. 1: Approach [1]

Additionally, Khattak's team focused on evaluating the K-Nearest Neighbor with Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) [1]. These metrics, along with Logcosh, were used to evaluate the model against different data and classifiers.

2 Background

The first supporting paper used for this report is "*Application of machine learning in stock selection*". Secondly, "*A comparative study of supervised machine learning algorithms for stock market trend prediction*" is the additional supporting paper utilized. Both research papers explore how machine learning can predict stock trends. The following sections highlight the pros and cons of each on these papers in relation to this report.

2.1 Pros

The main benefit of "*Application of machine learning in stock selection*" is comparing how different algorithms perform when predicting the stock market. Li's research team used six commonly used methods including: linear model, clustering, support vector machine, random forest, neural network, and deep learning [2]. By understanding the comparisons of the different models, a similar approach can be taken.

Secondly, "A comparative study of supervised machine learning algorithms for stock market trend prediction" provides the necessary steps to breakdown a problem such as this. Also, this paper uses different metrics in evaluating the models such as accuracy and F–measure [3].

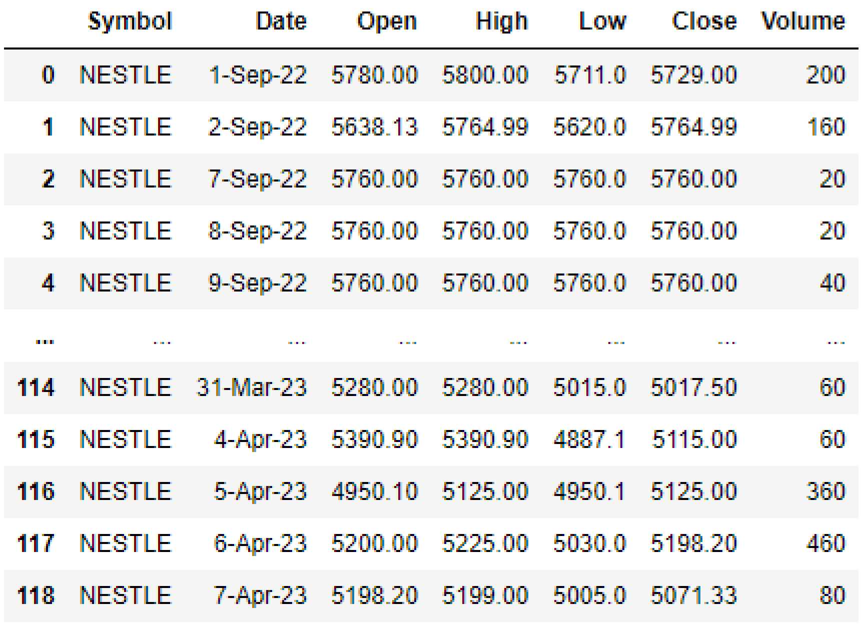
2.2 Cons

The main drawback of these two research papers is that they both use more complex data. Without knowledge of finance, it is difficult for me to understand the data used, yet alone see how they impact machine learning. In this, Li's team utilized features such as excess return rate, information ratio and max drawdown [2]. Kumar's team used features of Moving average, Rate of Change, Relative Strength Index, and Volatility, just to name a few [3]. As mentioned, the goal of this report is to determine the legitimacy of an accurate model using simple to understand features.

3 Method

3.1 Data Collection

The first step in reproducing “*Stock Market Trend Prediction using Supervised Learning*” work is to collect the data for the K-Nearest Neighbor algorithm. Presented in the paper, Khattak’s team utilized , https://www.ksestocks.com/QuotationsData, to gather 8 years of historical data from the Karachi stock exchange [1]. I pulled data from September 1st, 2022, to April 7th, 2023, from NESTLE to use. Highlighted in (Figure 2), 119 daily entries were collected where each daily entry contains the stock symbol, date, open price, daily high, daily low, close price, and volume traded.



	Symbol	Date	Open	High	Low	Close	Volume
0	NESTLE	1-Sep-22	5780.00	5800.00	5711.0	5729.00	200
1	NESTLE	2-Sep-22	5638.13	5764.99	5620.0	5764.99	160
2	NESTLE	7-Sep-22	5760.00	5760.00	5760.0	5760.00	20
3	NESTLE	8-Sep-22	5760.00	5760.00	5760.0	5760.00	20
4	NESTLE	9-Sep-22	5760.00	5760.00	5760.0	5760.00	40
...	...	...	...	...	...	...	...
114	NESTLE	31-Mar-23	5280.00	5280.00	5015.0	5017.50	60
115	NESTLE	4-Apr-23	5390.90	5390.90	4887.1	5115.00	60
116	NESTLE	5-Apr-23	4950.10	5125.00	4950.1	5125.00	360
117	NESTLE	6-Apr-23	5200.00	5225.00	5030.0	5198.20	460
118	NESTLE	7-Apr-23	5198.20	5199.00	5005.0	5071.33	80

Fig. 2: Initial Dataset Collected and Loaded

3.2 Data Normalization

Now that the data has been obtained and loaded into a Pandas Data Frame, it is time to normalize the data. Following Khattak’s team, normalizing the data contains three steps: calculating stock return, calculating Z-Score, and removing outliers [1]. Following these steps builds the target variable along with preprocessing the data to improve the model.

3.2.1 Calculating Stock Return

As it stands, only information about a stock is present but there is no indicator in how the stock is performing. Measuring performance of a stock is necessary in predicting its trend. To do this, Khattak’s team utilized the following formula to calculate stock return (Figure 3):

$$stock\_return = (current\_day\_closing\_price - closing\_price\_of\_previous\_day) / previous\_day\_closing\_price\_of\_the\_stock$$

Fig. 3: Stock Return Formula [1]

Stock return is a daily, simple representation if a stock's price went up or down on a day-to-day basis. This new variable is the target for the K-Nearest Neighbor algorithm. It uses the daily features and determines the effect on stock return, being able to predict it.

3.2.2 Calculating Z-Score and Removing Outliers

The next step of normalizing the data was calculating the Z-Score of an entry. To ensure the best performance of a model, it is critical that the data given accurately depicts the normal trend. In this, removing outliers helps the model as it is not negatively impacted by irregular data. To determine outliers, an entries Z-Score is calculated with the following formula (Figure 4):

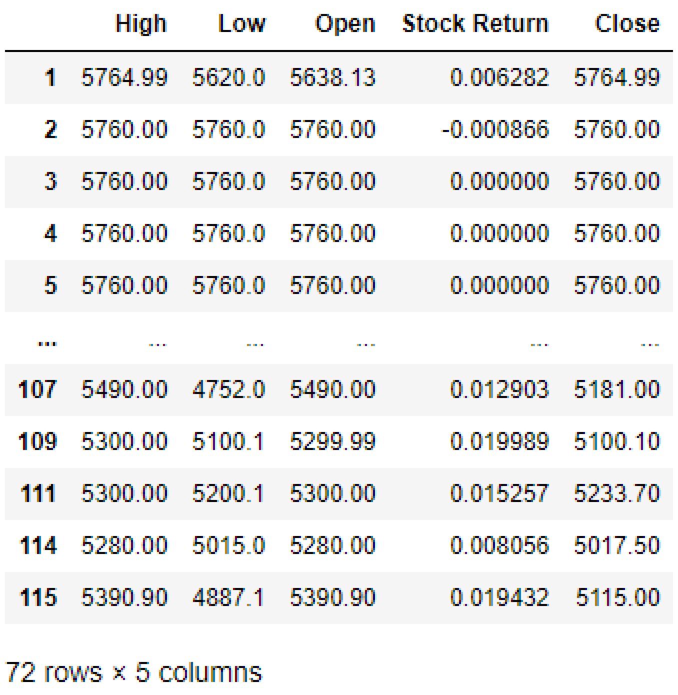
$$Z - score = Open - \frac{Mean\ of\ Open}{Stdv\ of\ Open}$$

Fig. 4: Z-Score Formula [1]

To identify the outliers, each entries Z-Score was evaluated above. An entry is marked as an outlier if the Z-Score was greater than one or less than negative one [1]. If it fell outside of this range, it is removed from the dataset. After following the steps to calculate and apply the Z-Score to identify outliers, my dataset went from 119 entries to 72 entries.

3.3 Resulting Dataset

(Figure 5) is my resulting dataset after following the above steps. With the data now normalized and the outliers removed, I split the data on a .80/.20 split for training and testing [1]. This is the data used to train, test, and evaluate the K-Nearest Neighbor model.



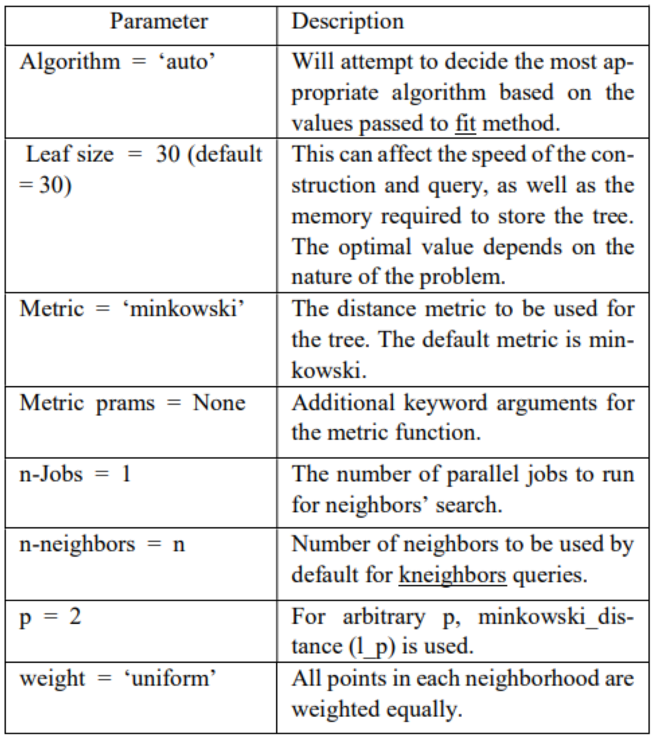
	High	Low	Open	Stock Return	Close
1	5764.99	5620.0	5638.13	0.006282	5764.99
2	5760.00	5760.0	5760.00	-0.000866	5760.00
3	5760.00	5760.0	5760.00	0.000000	5760.00
4	5760.00	5760.0	5760.00	0.000000	5760.00
5	5760.00	5760.0	5760.00	0.000000	5760.00
...	...	...	...	...	...
107	5490.00	4752.0	5490.00	0.012903	5181.00
109	5300.00	5100.1	5299.99	0.019989	5100.10
111	5300.00	5200.1	5300.00	0.015257	5233.70
114	5280.00	5015.0	5280.00	0.008056	5017.50
115	5390.90	4887.1	5390.90	0.019432	5115.00

72 rows × 5 columns

Fig. 5: Normalized Dataset

3.6 Apply K-Nearest Neighbor

The algorithm used for this report was sklearn.neighbors.KNeighborsRegressor. To follow Khattak's team report, the following parameters were given to the model (Figure 6). However, the 'n-neighbors = n' parameter different throughout the iterations. I lowered the number of neighbors that a data point would evaluate due to the significantly less amount of data I had.



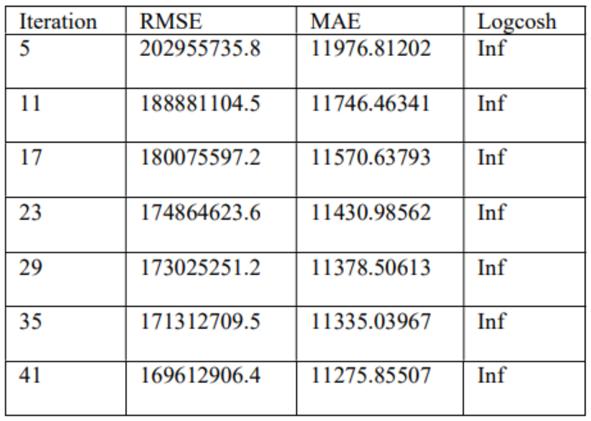
Parameter	Description
Algorithm = 'auto'	Will attempt to decide the most appropriate algorithm based on the values passed to fit method.
Leaf size = 30 (default = 30)	This can affect the speed of the construction and query, as well as the memory required to store the tree. The optimal value depends on the nature of the problem.
Metric = 'minkowski'	The distance metric to be used for the tree. The default metric is minkowski.
Metric prams = None	Additional keyword arguments for the metric function.
n-Jobs = 1	The number of parallel jobs to run for neighbors' search.
n-neighbors = n	Number of neighbors to be used by default for kneighbors queries.
p = 2	For arbitrary p, minkowski_distance (l_p) is used.
weight = 'uniform'	All points in each neighborhood are weighted equally.

Fig. 6: KNN Parameters [1]

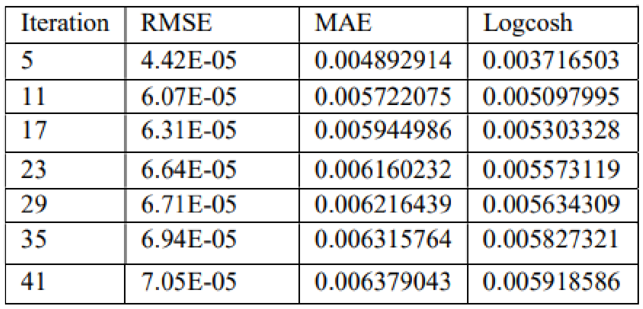
4 Experiment

4.1 Comparing Results of K-Nearest Neighbor

The main results of "*Stock Market Trend Prediction using Supervised Learning*" provides insight into how the K-Nearest Neighbor algorithm performs better will the normalized and filtered data compared to a raw dataset. By evaluating the model using RMSE, MAE and Logcosh, the following two figures from Khattak's team highlight the improvement of the model when using the normalized data (Figure 7, Figure 8):



Iteration	RMSE	MAE	Logcosh
5	202955735.8	11976.81202	Inf
11	188881104.5	11746.46341	Inf
17	180075597.2	11570.63793	Inf
23	174864623.6	11430.98562	Inf
29	173025251.2	11378.50613	Inf
35	171312709.5	11335.03967	Inf
41	169612906.4	11275.85507	Inf

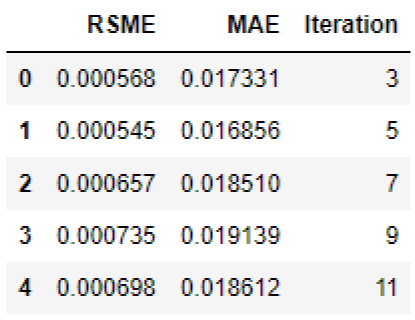


Iteration	RMSE	MAE	Logcosh
5	4.42E-05	0.004892914	0.003716503
11	6.07E-05	0.005722075	0.005097995
17	6.31E-05	0.005944986	0.005303328
23	6.64E-05	0.006160232	0.005573119
29	6.71E-05	0.006216439	0.005634309
35	6.94E-05	0.006315764	0.005827321
41	7.05E-05	0.006379043	0.005918586

Fig. 7: KNN Results on Raw Dataset [1]

Fig. 8: KNN Results on Normalized Dataset [1]

Both error rates, RSME and MAE are significantly lower showcasing the model's improved performance. When running the data of my experiment, I receive comparable results to (Figure 8). Below are the results of my K-Nearest Neighbor model where n-neighbors = iterations (Figure 9):



	RSME	MAE	Iteration
0	0.000568	0.017331	3
1	0.000545	0.016856	5
2	0.000657	0.018510	7
3	0.000735	0.019139	9
4	0.000698	0.018612	11

Fig. 9: My Results of KNN

Plotting these results helps visualize the data over the iterations. (Figure 10) shows the resulting RSME and MAE over the differ iterations. It was interesting to see how the model improved by increasing number of iterations up until 5. Increasing n from there lead to worse results up until 9 neighbors where it seems to improve again. However, I believe that the improvement of RSME and MAE at 9 or more neighbors might because the model overfits the data due to the little sample size.

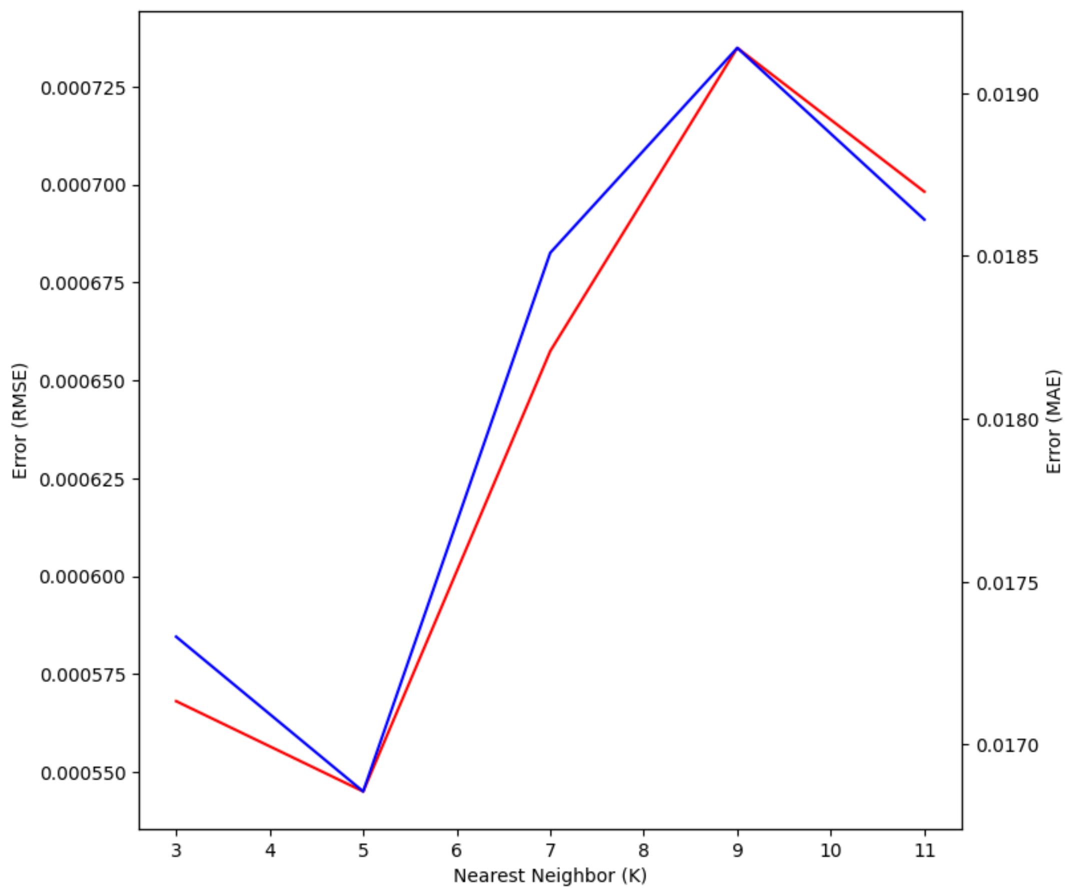
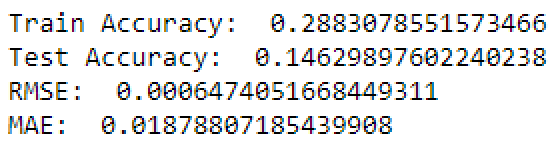


Fig. 10: Visualization of my KNN Results

Overall, the evaluation of RMSE and MAE on my model seem similar to the results showcased in Khattak's paper. What is interesting to note however, is that Khattak's team never used accuracy to measure the model's performance. I was curious about this, so I included accuracy within my python script and the train score was 30.35% and a test score of 28.13%.

4.2 Comparing Support Vector Machines

The next additional test I wanted to include was use this data with a support vector machine model. Using scikit-learns support vector regression package, I created a model and evaluated the resulting RSME, MAE and overall accuracy (Figure 11).



```
Train Accuracy:  0.2883078551573466
Test Accuracy:  0.14629897602240238
RMSE:  0.0006474051668449311
MAE:  0.01878807185439908
```

Fig. 11: SVM Results

The SVM had an RMSE of .000647 and MAE of 0.0187. It is also interesting to note that the SVM resulted in worse accuracy overall. The train score was 28.83% and the test score was 14.62%. When comparing these results, the KNN performs better. It had higher training and test scores along with a lower RMSE and MAE at n=5 neighbors.

4.3 Future Experiments

In future experiments I would like to compare these results to additional models such as random forest and naïve bayes. In doing this, the first step I would take would be getting a larger sample size of data. I learned that the limited amount of data that I used does not accurately represent the model's potential in determining stock trends. Additionally, I will include more metrics for evaluation such f-1 score.

5 Conclusion

Overall, I do not believe that I successfully replicated the work in creating an accurate K-Nearest Neighbor model. Although my results of MAE and RMSE seem similar, the train and test accuracy scores seem exceptionally low. In this, I could not successfully implement a machine learning algorithm to accurately predict stock trends.

I think that the biggest problem with the project was the limited data that I used. As I mentioned above, Khattak's team used 8 years' worth of data while I only had 7–8 months of it. This was leading factor in producing an inaccurate model and is the first step that needs to be revisit. Additionally, I would like to further this research by incorporate more metrics to analyze the models. In conclusion, I recognized the value that a dataset holds when using machine learning and need to revisit it before any more progress can be made.

Sharing Agreement

I would not like this assignment to be shared.

References

[1] Asad Masood Khattak, Habib Ullah, Hassan Ali Khalid, Ammara Habib, Muhammad Zubair Asghar, and Fazal Masud Kundi. 2019. “Stock Market Trend Prediction using Supervised Learning”. In Proceedings of the 10th International Symposium on Information and Communication Technology (SoICT '19). Association for Computing Machinery, New York, NY, USA, 85–91. https://doi.org/10.1145/3368926.3369680

[2] Li, Pengfei, Xu, Jungang and AI-Hamami, Mohammad. "Application of machine learning in stock selection" Applied Mathematics and Nonlinear Sciences, vol.0, no.0, 2022, pp.-. https://doi.org/10.2478/amns.2022.1.00025

[3] Kumar, K. Dogra, C. Utreja and P. Yadav, "A Comparative Study of Supervised Machine Learning Algorithms for Stock Market Trend Prediction," 2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT), Coimbatore, India, 2018, pp. 1003-1007, doi: 10.1109/ICICCT.2018.8473214.