International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 07 Issue: 02 | February - 2023 Impact Factor: 7.185 ISSN: 2582-3930

Quantitative Finance Model Visualizer Using High-Frequency Trading Algorithms

Soumya Kumari (1DS19CS161)
Xitiz Verma (1DS19CS193)
Yashaswini Shree (1DS19CS196)
Mohammed Shagil Khan (1DS20CS415)

Guided By – **Prof. Sunanda**
Department of Computer
Science Engineering
DSCE

1. Abstract

Our project is about quantitative finance which refers to the methodology of applying disciplines from mathematics, statistics, and finance with computer programming to create a financial tool for future stock prediction. Stock market prediction is a very tough job as very high precision is needed. So many models have been used earlier but none of them are reliable and consistent. Thus, we propose a solution using deep learning, as it uses current output in further steps and understands the relationship between them.RNN has a vanishing gradient problem which happens due to the use of the same parameters repeatedly. This problem is avoided by the use of different parameters at each step. We balance the situation by adding variable-length sequences, generating variable-length sequences, and keeping the learnable parameters count constant. We are introducing gated cells like LSTM. Also, RNN can't retain the data for a long duration. Hence, we are planning to use the LSTM model which is a kind of RNN that has long-term memory. Lstm works on time series data and time series data can be influenced by any tiny noise. Time series data helps us to track differences over time. For this project, we are using high-frequency data which means everyday data fluctuation is included. Also, we are using the random forest for noise reduction and outlier detection. Random forest is normalizing our output and provides accuracy.

Keywords: Artificial Intelligence; machine learning algorithms; Deep learning;

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 07 Issue: 02 | February - 2023 Impact Factor: 7.185 ISSN: 2582-3930

2. Introduction

Shares of businesses are traded on the stock market for stockbrokers. One of the most difficult tasks is projecting stock prices since accurate forecasting is essential to success in the stock market. Due to the stock market's volatility, a number of strategies are employed to predict price, but none of them has been demonstrated to be a reliable tool for consistent prediction.

As a result, we suggested the Artificial Neural Network (ANN) technique since, after learning from and analyzing the initial inputs and their relationships, ANN can generalize and forecast data. In our Quant Major Project, we set out to develop models on basis of technical indicators and Long Short Term Memory to forecast the price of a stock that varies daily as well as financial statistics that can be seen visually, like Skewness, Kurtosis, Holistic Volatility, etc., that are all compiled from day-end data for any publicly traded stock.

The regression Long-Short Term Memory ANN model is trained after we first obtain a dataset of intraday trading of any publicly traded stock of the Indian Stock Exchange Market from Kaggle. A special type of artificial neural network called Long Short-Term Memory (LSTM) is employed in deep learning and artificial intelligence.

One of the largest innovations to occur in the last 15 years is Algorithmic trading's High-Frequency Trading (HFT) division. The ability of a trader to take orders with extremely little lead time is referred to as HFT or nano trading. This model is run using price history together with technical analysis signals and strategies, and the results are assessed using profitability and performance measures. Based on historical stock data, the neural network model is successfully used to predict the daily highest, lowest, and closing values of business stocks in a short amount of time, however, it is ineffective in predicting the return rate of the stocks. Various features such as stochastic indicators, moving averages, and RSI is extracted from the historical stock data to train the ANN model. The dataset is then divided into training and testing sets which are used for the accuracy of the ANN model.

Thus we are combining both Data mining and neural networks in our proposed system to first collect and refine stock data then analyze these data with the ANN method and provide the result of the input data in prediction using data mining and LSTM algorithm to predict the stock value more accurately.

3. Methodology

3.1 Statistical Indicators

3.1.1 Normal distribution

It is a probability distribution that tells us that data about the mean is more frequent compared to data far from the mean.
It is also called Gaussian Distribution.
Normal distributions have a kurtosis of value 3.

- Normal distribution has two parameters: mean and standard deviation
- It is visually depicted as a bell curve.
- It shows symmetrical data around the mean and its width represents the standard deviation.
- Its mean, median, and mode are all equal to one another.

Normal Distribution Formula

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

Here, μ = the mean
x = value of the variable
σ = the standard deviation

3.1.2 Skewness

Asymmetry is a measure of the asymmetry or distortion of a symmetrical distribution. It measures the deviation of a given distribution of a random variable from a symmetric distribution, such as a normal distribution. A normal distribution is not skewed because it is symmetric on both sides. Therefore, a curve is considered biased if it is shifted to the right or to the left. Skewness tells us **the direction of outliers**.

There are two types of skewness as shown in Fig 1. If the given distribution is skewed or shifted to the left and its tail to the right then it is called positive skewness. Its skew value is greater than zero. Tails are the end of the curve that indicates extreme values and are infrequent.

And, if the given distribution is shifted to the right and its tail to the left then it is called negative skewness and has a skew value less than zero.

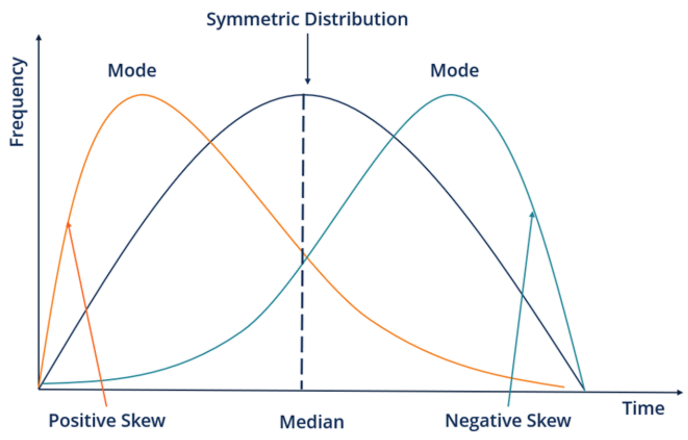

Fig 1: types of skewness

Measuring skewness:

The formula for Pearson mode skewness:

$$\text{Skewness} = \frac{\bar{X} - M_o}{s}$$

Where:

- **X** = Mean value
- $\mathbf{M_O}$ = Mode value
- s = Standard deviation of the sample data

3.1.3 Kurtosis

Kurtosis represents the degree of outliers and extreme values present in any data distribution.

Kurtosis is considered a statistical measure of whether the data is heavy-tagged or light-tagged in a normal distribution. Kurtosis is used to measure financial risks and threats.

A large kurtosis value means that the chances of extreme returns are very high. It also means that there are chances of substantial and extremely low returns which can lead to loss and be considered as a fiscal threat.

Tails are the ends of the distribution. It tells us about the presence of extreme values compared to the mean.In other words, tails represent how frequently outlier occurs.

While a small kurtosis value represents a moderate threat because the chances of high or low returns are fairly less.

Excess kurtosis = kurtosis -3

Excess kurtosis is used in statistics and probability calculation for comparison of the kurtosis value of the normal distribution. Excess kurtosis can be positive, negative, or zero.

Types of kurtosis as shown in Fig 2:

- **Mesokurtic**

- It is a medium-tailed distribution.
- value = 3.
- Distributions with medium tails are called mesokurtic.
- So, any distribution with excess kurtosis equal to zero is mesokurtic.

- **Platykurtic**

- It is a distribution with a low kurtosis value i. e. thin-tailed.
- It has less distribution than a normal distribution.
- It has a kurtosis value of less than 3 and an excess kurtosis value of less than 0.
- It is also known as negative kurtosis.
- As a result of the playkurtic's lower tail and being stretched around the center tail, most of the data points are close to the mean point.
- An advantage of a platykurtic distribution is that it is flatter than a normal distribution.

- **Leptokurtic**

- It is a distribution with a high kurtosis value
- Many outliers are present in it since it is fat-tailed
- It is known as called positive kurtosis.
- There are thick tails and a peak in the distribution when the value is positive.
- There are more outliers in the Leptokurtic since its tails are disproportionately long and skinny.
- When a value is extreme, it means that many of the figures are distant from the mean and located in the tails of the distribution.

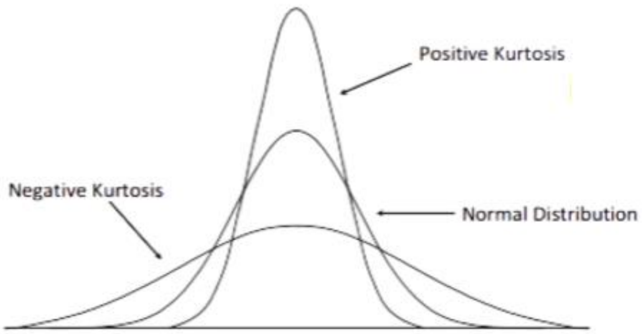

Fig 2: Types of kurtosis

3.1.4 Holistic volatility

It is a statistical indicator that measures the dispersion of returns. Generally, the higher the volatility higher is the risk as the probability of extremely high or extremely low returns is high. In the case of small volatility, the risk is moderate as the probability of extreme returns is low. In other words, we can say that it indicates uncertainty. It shows how an asset's price changes around the mean value.

A higher volatility rate also means that the price of the security can change rapidly over a period of time. While low volatility indicates it won't change dramatically in a short period of time.

Formula:

$$vol = \sigma\sqrt{T}$$

where:

- v = volatility over some interval of time
- σ =standard deviation of returns
- T = number of periods in the time horizon

3.2 Prerequisites for Quantitative Finance

Despite the fact that everyone can perform their own quantitative analysis, institutional investors and money managers usually supervise it. Complex modeling is required for the analysis, which also uses data that can be hard to find. Institutional traders and investment firms already have access to the human and financial resources needed to integrate Quant models into trading strategies. Because there is a lot of money at risk, institutional investors typically need to backtest to identify risk.

Before we start studying a trading technique, we need to consider the following factors:

1. Money logic
2. Market industry
3. Data
4. Programming dialects

Let's take a closer look at each of these components.

3.2.1 Trading logic/hypothesis for quantitative finance

Quantitative finance will be used to analyze a trading strategy, but before we analyze anything, we need to have a clear idea of what we're trying to analyze. That is the analysis's main trading point or thesis.

The most important step in the study is when we calculate the historical time returns of the securities and are confident in our trading logic since only then can we backtest the trading strategy.

3.2.2 Choosing the right market or asset segment for quantitative finance

We can consider a number of variables to determine which request or method will be fashionable for the type of trading we're intending to do.

The considerations may include the risks it is willing to face, the rewards it hopes to achieve, and the amount of time it will commit, whether long- or short-term.

Trading in cryptocurrencies, for instance, may be riskier than trading in other asset classes, but it can also result in higher rewards, and vice versa. Choosing the appropriate request and asset class to trade in is therefore crucial.

3.2.3 Data for quantitative finance

Analyze the trading strategy after the means have been given. The next step is to select the stock's literal data. Data is gathered from any broker or from open source platforms like Kaggle, Yahoo Finance, etc. by the data seller.

It is crucial to choose high-quality data or data that is error-free.

Then, we're using time-series data which is taken by the top 50 company NIFTY stocks from 2000 to 2021. Our dataset is taken from Kaggle which is a dependable and secure source.
We choose to use Indian Nifty 50 Stocks because of their high liquidity and abundance of data. Since a lack or lack of trading data can impede the learning process, the abundance of data that characterizes Indian Nifty 50 Stocks, even in brief time intervals, is crucial for training our models.
Date, symbol, series, previous close, high, low, open, VWAP (volume weighted average price), volume, turnover, transactions, deliverables, and percent deliverables are all included in our dataset.
The time and date the trade occurred, its price, and its size are all included in the raw trade data (in stocks).

3.2.4 Choosing the programming language for Quantitative Finance Analysis

It is obvious with trading sense, naming the appropriate asset for trading, and acquiring the required asset data.
Choosing the computer language that will be used to analyse a trading strategy is the last step. We then employ Python and its collection of machine learning (ML) libraries, including Keras, Pandas, MatplotLib, Dense, Tensorflow, Scikit, Pytorch, Numpy, Django Tastypie, etc.

3.3 Long short-term memory (LSTM)

Here, We are proposing the ANN technique as ANN takes the current output for further rounds of processing along with some other parameters. It also analyzes the relationship between them. We are using the Long short-term memory model which is a special kind of RNN, capable of learning long-term dependencies. In an RNN, the output of the last step is provided as the input to the current step. LSTM was designed by Hochreiter & Schmidhuber. It solves the problem of RNN, where RNN can't retain the data for a long period of time but can make more correct predictions from recent information. As the distance length increases, the RNN performance decreases. By default, LSTM can store information for a long period of time. It is used for processing, prediction, and classification and it works on time-series data. Due to the RNN block's repetitive usage of the same parameters at each step, the gradient can vanish, which is a concern. In order to avoid this issue, we are utilizing different settings at each phase. In addition to maintaining the learnable parameters constant, we are adding additional parameters at each step that are generalized to strings of varying lengths.
An LSTM with a near RNN cell is being introduced. Internal variables, or ports, are found in closed cells. The value of each gate is determined by the information for each time step, including the beginning values. The individual variables will then be modified by multiplying the gate's value by those variables. In order to follow changes over time, LSTM uses time-series data, which is gathered at various intervals.
We can monitor development in milliseconds, seconds, days, and years using time series data. Our initial interpretation of time series data was intended to be more static; it included temperature-based daily highs and lows as well as the frequency of stock market openings and closings. We are importing a few Keras modules to create the LSTM, and we are using Matplotlib to create a graph that compares the projected and actual stock prices.
Here, we are implementing LSTM on our dataset.

To build our LSTM-based Recurrent Neural Network (RNN) to predict any stock price step by step. It is split into 7 parts as below.

1. Problem statement
2. Data processing
3. Model building
4. Model compiling
5. Model fitting
6. Model prediction
7. Result visualization

Data Pre-processing:
We want to preprocess these statistics earlier than making use of stock variation through the use of LSTM. Transform our statistics values with the use of the fit_transform function. Minmax scaler is used to scale the statistics in order that we will scale all of the price values to a not-unusual place scale. Then we use 80% of the statistics for schooling and the closing 20% for trying out and assign them to split variables.
Splitting data for training:
A function is created to create training and test sequences.

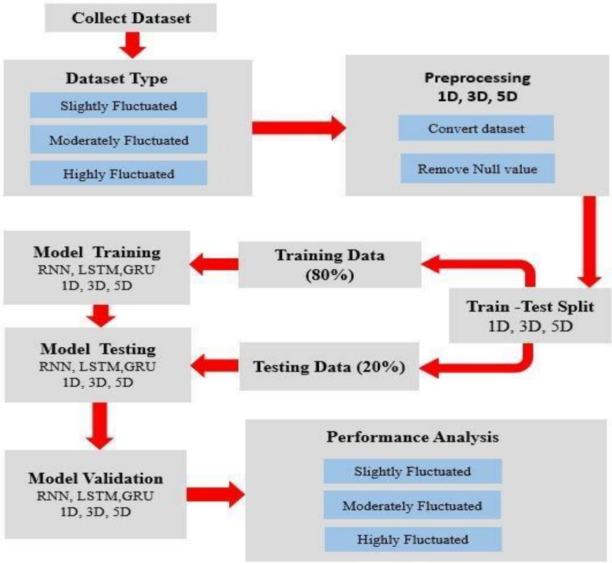

Fig 3: LSTM model steps

Implementation of LSTM model shown in Fig 3

The following step is creating our LSTM version. We can use the Sequential version that was imported from Keras and the needed libraries have been loaded.

We use LSTM layers in our version and put into effect removal within the center for normalization. The variety of gadgets allotted withinside the LSTM parameter is fifty. with a dropout price of 10%. The root manner of the rectangular mistakes is the loss of features to optimize the hassle with the Adam optimizer. The suggested absolute mistake is the metric utilized in our LSTM community due to the fact it's miles related to time collection data.

Visualization:
After the data has been fitted into our model, predictions are made using the model. To obtain the initial value using the modified function, we must use the inverse transformation. Using this data, we can now visualize predictions.

3.4 Random Forest

It is a supervised classification machine learning algorithm that classifies the data by building multiple Classifiers with the goal to achieve prediction with higher accuracy. It consists of multiple decision trees and helps solve the problem of decision tree overfitting. A decision tree is a hierarchical structure consisting of nodes and directed edges used to test specific attributes of an object. The components of a decision tree are:

- Root node: The root node represents the entire population or sample, further divided into two or more homogeneous groups. our starting point.

- Split: Taking a node and splitting it into two or more subnodes. For example, split by gender.

- Decision Node: A decision node is a subnode that has been divided into several other subnodes.

- Leaf/End Nodes: Leaf or end nodes are nodes that do not split.

- Pruning: Pruning is the process of deleting a decision node's subnode. It can also be described as the opposite of division.

- Branch/Subtree: A branch or subtree is a section of the entire tree.

- Parent node and child node: The parent of a subnode is a node that has been divided into subnodes, and the subnodes are the parent node's offspring.

This is important because the decision tree algorithm looks for the best criteria for transforming the dataset into a homogeneous subset (i.e. contamination) rather than a heterogeneous subset. Random forest in classification results in better accuracy in prediction than other classification algorithms

Entropy is a measure of the contamination of data points in a dataset. A node is purest when there are only instances of one class and can be calculated using the following formula:

$$Entropy = \sum_{i=1}^{C} -p_i * \log_2(p_i)$$

Where c=number of features

i=feature

P= probability of I

Disturbances at nodes can be calculated by entropy calculations. Information retrieval uses entropy to determine the features that provide the maximum information about the classification by reducing the entropy and its partitions accordingly.

$$IG(Y,X) = E(Y) - E(Y|X)$$

Like entropy, the Gini index varies between values of 1 and 0. It calculates the classification impurity by measuring the deviation between the probability distributions of the target attribute's values. An output of 0 is uniform and 1 is impure, indicating that new random data is likely to be misclassified. Once this is achieved, a node split is performed that reduces this computed pollution. Here are the Gini expressions for various classes 'C':

$$G = \sum_{i=1}^{C} p(i) * (1 - p(i))$$

3.5 Buying Points

A buy point is a price position at which a stock is most likely to begin a significant advance. It also points to an area of the map that offers the least quantum of resistance to price progress.

Quantitative Finance Analysis against Data

- In quantitative finance analysis, the delicacy of the data is assessed by applying a strategy or prophetic model to the literal data.
- Dealers can test trading techniques without putting their own money at risk.
- Net profit/loss, return, threat-acclimated return, request exposure, and volatility are examples of common financial analysis metrics.

Quantitative finance analysis is used by analysts and traders to assess and test various trading strategies without putting plutocrats at risk. The argument is that since their strategy performed poorly in the past, it is unlikely that it will do so going forward (and vice versa). The overall profitability and the threat position taken are the two key variables examined throughout testing.

However, an analysis will consider how well a strategy performs in relation to a wide range of variables. Traders will be presented with a strategy that has historically produced positive outcomes if the analysis is successful. Backtesting is based on the assumption that stocks move in identical patterns to how they did historically, even when the request in no way moves the same.

Frequently Used Quantitative Finance Measures

- Net Income/Loss
- Return: The portfolio's overall return during a specific time period
- Return with Risk Adjustment: The portfolio's return with a level of risk taken into account.
- Market Exposure: The extent of exposure to certain market segments
- Volatility: The range of a portfolio's returns

Confidence Interval

In statistics, a confidence interval is an estimate of a range that could include a population parameter. An estimate of a sample parameter derived from the tested data is used to set up the population parameter that is unknown. As an example, the sample mean x is used to draw up the population means.
The lower and upper boundaries of the interval serve as broad definitions. When represented as a chance, the confidence interval is (the most constantly quoted probabilities are 90, 95, and 99). The adjustment represents the position of confidence.
Since it is utilized as a measure of the query, the conception of the confidence interval is extremely crucial in statistics (thesis testing).
Since it is employed as a gauge of uncertainty, the confidence interval notion is crucial to statistics (hypothesis testing).

How to Calculate

The following steps are used to compute the interval:
1. assemble the test data.
2. Determine the sample mean by x.
3. Identify whether the standard deviation of a population is known or unknown.
4. We can use a z-score to get the relevant confidence level if we are aware of the population's standard deviation.
5. We can use a t-statistic to calculate the corresponding confidence level when the standard deviation of a population is not known.
6. Utilize the following formulas to determine the lower

and upper boundaries of the confidence interval:

The interval is calculated using the following steps:

1. Gather the sample data.
2. Calculate the sample mean $\bar{x}$.
3. Determine whether a population's standard deviation is known or unknown.
4. If a population's standard deviation is known, we can use a z-score for the corresponding confidence level.
5. If a population's standard deviation is unknown, we can use a t-statistic for the corresponding confidence level.
6. Find the lower and upper bounds of the confidence interval using the following formulas:

a. Known population standard deviation

$$\text{Lower bound} = \bar{x} - z \times \frac{\sigma}{\sqrt{n}}$$

$$\text{Upper bound} = \bar{x} + z \times \frac{\sigma}{\sqrt{n}}$$

b. Unknown population standard deviation

$$\text{Lower bound} = \bar{x} - t \times \frac{s}{\sqrt{n}}$$

$$\text{Upper bound} = \bar{x} + t \times \frac{s}{\sqrt{n}}$$

- Value at threat(VaR) is a statistic that quantifies the quantum of implicit loss that could occur within an investment, a portfolio of investments, or an establishment over a specified time period.
- The VaR uses both the confidence interval and confidence position to make a threat assessment model.
- A VaR assessment helps fiscal institutions identify high-threat investments and determine the cash reserves they will need to cover implicit portfolio losses.
- A confidence interval is two set values that indicate a parameter will fall between.
- The confidence level reflects the level of probability (expressed as a percentage) that the confidence interval would contain the population parameter.

Fiscal institutions can use VaR to assess how important cash they should keep in reserve to neutralize probable portfolio losses. Volatility is generally used by threat directors to statistically measure threat. Still, VaR is generally used by investment and marketable banks to calculate the total pitfalls associated with largely connected positions held by several divisions of the business.

The institution can estimate with a high degree of certainty the maximum quantum or chance that could conceivably be lost on an investment over a specific period of time using the VaR analysis. With the help of VaR modeling, directors can spot investments with advanced- then-respectable pitfalls, giving them the occasion to cut threats or exit positions as necessary.

Sharpe ratio

The redundant return, or Sharpe rate, is determined as portfolio returns reduce the standard divagation's threat-free rate of return per unit of return. A threat-free return is often a return on risk-free assets akin to government bonds.

To determine how our approach is compensating for the risk taken on the investment, the Sharpe rate can be utilized to compare the portfolio with the industry average.

Sharpe rate is calculated as (Portfolio Returns Threat-Free Returns) / Portfolio Returns Standard Deviation.

4. Applications

Cost-effective. RF can be trained much more quickly and inexpensively than neural networks. Its accuracy doesn't significantly deteriorate in the while. Because of this, random forest modeling, for instance, is utilized in mobile applications. resistant to overfitting If an outlier in the training set causes one tree to make an incorrect prediction, another will most likely make up for that prediction with the opposite outlier. Therefore, a group of uncorrelated trees performs better than any one of them when compared individually. high rates of coverage and little bias.
Because of the aforementioned, the random forest classifier is perfect in scenarios where our dataset may contain some

missing values or if we want to know how much variance there is between various types of data output (such as college undergraduates who are likely to finish their studies and leave, continue on to a master's degree, or drop out). applicable to both regression and classification. Both sorts of tasks have yielded similarly accurate results using RF. has the ability to manage missing values in features without biasing predictions. Simple to understand, we can examine any one tree to understand its prediction by looking at the forest as a whole, where each tree makes predictions on its own.

Applications of statistical indicators for traders and investors:

- Skewness tells about the direction of outliers.
- Kurtosis is used to tell financial risks, and whether the distribution contains any extreme values.
- If the kurtosis value is more there will be more outliers and fluctuations.
- A large value of kurtosis indicates the probability of extremely high or low returns. And, small kurtosis means a small threat as the probability of extreme returns is low.
- Skewness also contains extremes of a dataset and investor notes that while investing.
- Generally, standard deviation is used for forecasting returns by investors which uses normal distribution.
- It is better to consider performance on skewness because of the risks it possesses.
- Hence skewed data will increase the accuracy of financial models.
- A high Hostilic volatility value means security value can change rapidly over a short period of time and if it is low it won't fluctuate dramatically.

5. Conclusion

This study used the Indian Stock Market financing dataset and a variety of methodologies.We have created an application for predicting close stock prices using the LSTM algorithm. We are doing this for the closing stock price of any particular firm. We used datasets from the Nifty50 Stocks and obtained extremely high accuracy for them.

Positive outcomes have been achieved thanks to these strategies, which have improved prediction accuracy.

The study and forecast of stocks using recently released machine learning techniques have produced encouraging results, paving the way for their implementation in lucrative exchange schemes.We have come to the conclusion that by applying machine learning techniques, stock market predictions can be made more effectively and accurately.

Future Enhancement:

By using a dataset that is far bigger than the one being utilized right now, the stock market prediction system can be significantly improved in the future. The accuracy of our prediction models would increase as a result.

The stock market is renowned to be unpredictable, volatile, and nonlinear. It is highly challenging to predict stock prices with any degree of accuracy because of numerous (macro and micro) variables, including politics, global economic conditions, unforeseen events, a company's financial performance, and others. To find out what accuracy rate different machine learning models provide, additional research could be done.

We can expand this program in the future to forecast bitcoin trade, and we can also use sentiment analysis for more accurate forecasts.

6. References

1. Selvin S, Vijayakumar R, Gopalakrishnan EA, Menon VK, Soman KP. Stock price prediction using LSTM, RNN, and CNN sliding window model. In 2017 international conference on advances in computing, communications, and informatics (icacci). 2017:1643-1647. IEEE.

2. Bing Y, Hao JK, Zhang SC. Stock market prediction using artificial neural networks. In the Advanced Engineering Forum. Trans Tech Publications Ltd. 2012;6:1055-1060.

3. Gurjar M, Naik P. Mujumdar G. Vaidya T. Stock market prediction using ANN. Int. Res. J. Eng. Technol. (IRJET). 2018;5:2758-2761.

4.Hiransha M. Gopalakrishnan EA, Menon VK, Soman KP. NSE stock market prediction using deep-learning models. Procedia computer 2018;132:1351-1362. science.

5. Utomo D. Stock price prediction using back propagation neural network based on gradient descent with momentum and adaptive learning rate. Journal of Internet Banking and Commerce. 2017;22(3):1-16.

6.Chandar SK, Sumathi M, Sivanandam SN. Prediction of the stock market price using a hybrid of wavelet transform and artificial neural network. Indian Journal of Science and Technology. 2016;9(8):1-5.

7.Borovkova S, Tsiamas I. An ensemble of LSTM neural networks for high-frequency stock market classification. Journal of Forecasting. 2019;38(6):600-619.

8. Araujo RDA, Oliveira AL, Meira S. A hybrid model for high-frequency stock market forecasting. Expert Systems with Applications. 2015;42(8):4081-4096.

9. Deepak RS, Uday SI, Malathi D. Machine learning approach in stock market prediction. International Journal of Pure and Applied Mathematics. 2017;115(8):71 77.

10. Sarioglu E, Aladag CH, Yolcu U, Uslu VR, Basaran MA. A new approach based on artificial neural networks for high-order multivariate fuzzy time series. Expert Systems with Applications.2009;36(7):10589-10594. t14

11. Miah MBA, Hossain MZ, Hossain MA, Islam MM. Price prediction of the stock market using a hybrid model of artificial intelligence. International Journal of Computer Applications. 2015;111(3).

12. G. Box and G. Jenkins, *Time Series Analysis: Forecasting and Control*, Holden-Day, San Francisco, CA, USA, 1976.

13. J. H. Wang and J. Y. Leu, "Stock market trend prediction using ARIMA-based neural networks," *Proceedings of International Conference on Neural Networks (ICNN)*.
View at: Google Scholar

14. C. N. Babu and B. E. Reddy, "Selected Indian stock predictions using a hybrid ARIMA-GARCH model," in *Proceedings of the International Conference on Advances in Electronics*, Venice, Italy, August 2015.
View at: Google Scholar

15. S. Wichaidit and S. Kittitornkun, "Predicting SET50 stock prices using CARISMA (Cross-Correlation ARIMA)," in *Proceedings of the Computer Science & Engineering Conference*, Chiangmai, Thailand, December 2016.
View at: Google Scholar

16. C. Li and T. W. Chiang, "Complex neuro fuzzy ARIMA forecasting—a new approach using complex fuzzy sets," *IEEE Transactions on Fuzzy Systems*, vol. 21, no. 3, pp. 567–584, 2013.
View at: Publisher Site | Google Scholar

17. R. Singh and S. Srivastava, "Stock prediction using deep learning," *Multimedia Tools and Application*, vol. 76, no. 18, pp. 18569–18584, 2017.
View at: Publisher Site | Google Scholar

18. K. Vaibhav and M. Garg, "Prediction of stock closing price by the hybrid deep neural network," *European Journal of Advances in Engineering and Technology*, vol. 5, no. 4, pp. 282–287, 2018.
View at: Google Scholar

19. D. M. Q. Nelson, A. C. M. Pereira, and R. A. de Oliveira, "Stock market's price movement prediction with LSTM neural networks," in *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, Anchorage, AK, USA, May 2017.
View at: Google Scholar

20. S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
View at: Publisher Site | Google Scholar

21. X. L. Shao, D. Ma, Y. Liu, et al., "Short-term forecast of the stock price of multi-branch LSTM based on K-means," *International Conference on Systems & Informatics*, IEEE, 2018.
View at: Google Scholar

22. L. C. Cheng, Y. H. Huang, and M. E. Wu, "Applied attention-based LSTM neural networks in stock prediction," in *Proceedings of the International Conference on Big Data (Big Data)*, Seattle, WA, USA, December 2018.
View at: Google Scholar

23. L. Zhang, C. Aggarwal, and G. J. Qi, "Stock price prediction via discovering multi-frequency trading patterns," in *Proceedings of the International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, Halifax, Canada, August 2017.
View at: Google Scholar

24. Y. Zhao, Y. Shen, Y. Zhu, et al., "Forecasting wavelet transformed time series with attentive neural networks," in *Proceedings of the International Conference on Data Mining (ICDM)*, Singapore, November 2018.
View at: Publisher Site | Google Scholar

25.Y. Zhao, Y. Shen, and Y. Huang, "DMDP: a dynamic multi-source default probability prediction framework," *Data Science and Engineering*, vol. 4, no. 1, pp. 3–13, 2019.
View at: Publisher Site | Google Scholar

26. Korhan Gokmenoglu1, Baris Memduh Eren1,2, and Siamand Hesami1, " Exchange rates and stock markets in emerging economies: new evidence using the Quantile-on-Quantile approach",2021. |Google Scholar

27.AlexSherstinsky, "Fundamentals of Recurrent Neural Network and Long short term memory", 2019.