

Group1_SimilarityReport

by Mostafa Mahatabe

Submission date: 07-Sep-2022 01:58AM (UTC+0530)

Submission ID: 1893843921

File name: Group1_Report.pdf (702.74K)

Word count: 3663

Character count: 19374

Time Series Analysis for Stock Price Prediction

Fatima Juairiah, Aysha Siddika, Hasan Bin Jamal, Mostafa Mahatabe
Department of Computer Science and Technology
Ahsanullah University of Science and Technology, Dhaka, Bangladesh

Abstract—Predicting future stock market volatility has always been a demanding task for research studies. Individuals around the world have long regarded the stock market as a substantial profit. This research study seeks to comprehend historical stock market information and extract analysis from it in order to remedy the disparity between price movements and risk attitudes. A stock data set contains numerous precise terms that are difficult for an individual to comprehend when considering stock market expenditures. This thesis tries to fill that knowledge gap. This study aims to provide a future market scenario supported by statistical data. To foresee stock market volatility, we used time series analysis with the LSTM (Long Short-Term Memory), Autoregressive Integrated Moving Average (ARIMA), Hidden Markov Model (HMM), and Transformer model.

Keywords—Stock market prediction, LSTM, ARIMA, HMM, Transformer model, Time series analysis.

I. INTRODUCTION

The stock exchange refers to the network of trades where shares of major corporations can be traded. A stock price prediction is an attempt to forecast the future value of an individual stock. The Stock Market Prediction is an indication of time-series forecasting in which prior data is quickly examined and subsequent data values are estimated. A time series is an accumulation of analyses of data items over a specific time period.

Accurate stock market prediction is essential for any company as it allows it to build and optimize strategies, explore new scopes, find insights from the business data, and help achieve tangible business results. The principal target of every business is making a profit; for this purpose, time series analysis for their stock data is fundamental. A few other research on forecasting stock prices have really been undertaken. Similar approaches like Linear Regression and Support Vector Regression have been utilized, however the effectiveness was inadequate. In this research, we implemented the time series analysis using the LSTM (Long Short-Term Memory), Autoregressive Integrated Moving Average (ARIMA), Hidden Markov Model (HMM), and Transformer model to predict future stock prices.

The Transformer model and Hidden Markov models are rarely employed in time series analysis, some scholars have already presented novel methods for forecasting stock prices utilizing several classical models. In order to accurately

anticipate stock price, we used HMM and Transform models, which gave us good accuracy.

We're working with the Microsoft Corporation Stock dataset from 2012 here. We are primarily concerned with the date and the closing price of the stock. In stock price forecasting, the closing price of a stock has a different meaning. The Closing Price assists the shareholder in understanding the stock market viewpoint over time. It is the most precise matrix for determining stock valuation until the market resumes trading the following day.

II. RELATED WORKS

Professional investors utilize the share price analysis technique to decide whether to buy or sell a stock. By critically thinking, examining, and evaluating both historical and recent data, traders and investors try to obtain an advantage in the markets. There are extensive research papers on this topic, which makes it more convenient.

The paper [1] emphasizes the importance of time series by analyzing historical stock market data and thoroughly examining its fundamental principles to forecast stock price movements utilizing the LSTM neural network model's proved deep learning function for selective memory. The input layer, hidden layer, attention layer, and output layer are the four layers that make up this model. Model training is accomplished using the gradient descent approach. They selected 300 indexes of representative relevance for China's stock market for the dataset, which included six parameters such as trading date, opening price, closing price, lowest price, maximum price, and daily volume of the stock. After normalization, they divide the input data into training and test sets in a 7:3 ratio after normalization, and they transform each input data element into the interval [0,1]. Following calculation, the stock deep learning model's hidden layer and Attention layer forecast the next trading day's closing price. These 300 Chinese stock market data are first preprocessed by removing null and invalid data and data standardization. They employed mini-batch gradient descent to train the model, using a learning rate of 0.001. LSTM gives 95.70% accuracy on the training dataset and 80% on the test dataset.

The hypothesis that merging RNNs with useful input variables might offer a more effective method for anticipating stock indicating the following day is examined by Gao,

Tingwei et al. (2017) [2]. The goal of this study was to find whether the stock prediction model could be implemented using long short term memory (LSTM) and main trade data. For the study, Yahoo Finance stock trading data were gathered in order to assess and validate the efficacy of several models. A case study was presented based on Standard & Poor's (S&P500). Five different models were tested to demonstrate the utility of our system. The mean absolute error (MAE), root mean square error (RMSE), mean absolute percentage error (MAPE), average mean absolute percentage error (AMAPE) was conducted to evaluate the performance of the model. LSTM model performs better result for predictive experiments.

The author [3] Gourav Bathla, attempted to enhance stock price prediction using ARIMA in this work. Deep learning techniques are used due to their proven accuracy in numerous analytical sectors because of the very significant volatility in stock values. Recurrent neural networks were used in this case to boost prediction accuracy even further. For stock price prediction, they employed Linear Regression, Support Vector Machines, LSTM, CNN, RBM, ANN, ARIMA (Auto Regressive Integrated Moving Average), and standard machine learning approaches. Jan 2015 to June 2019 was used for training, and July 2019 to Jan 2020 was used for testing and validation. They employed Jan 2015 to Jan 2020. In this research, LSTM is applied to several stock indexes and compared to ARIMA and linear regression. The stock indexes used are the NYSE, NSE, NASDAQ, Dow Jones Industrial Average, and S&P500. In this study, LSTM performs better than SVR and offers more accurate prediction. On several stock markets, convolutional neural networks achieve an accuracy of 92%.

In order to find the most precise trained algorithm for forecasting future values, Adil Moghar et al. (2020) [4] explored the ML algorithm based on LSTM RNN to anticipate the adjusted closing prices for a portfolio of assets. The outcome of this paper's model has indicated some encouraging results, demonstrating that the constructed model is capable of tracking the evolution of opening prices for both assets. The model takes 80% of the data for training and the remaining 20% for testing. Mean squared error is used to optimize the model during training. Additionally, increasing the number of training data epochs (12, 25, 50, and 100) improves precision.

III. BACKGROUND STUDY

A. Autoregressive Integrated Moving Average (ARIMA)

ARIMA is an acronym that stands for AutoRegressive Integrated moving average, and it's miles a category of the version that captures a collection of various trendy temporal systems in time collection information. ARIMA version is used to be expecting destiny time collection motion via way of means of analyzing the variations among values within the collection in place of thru real values and additionally It may be used for any nonseasonal collection of numbers that

famous styles and isn't always a chain of random events. The parameters of the ARIMA version are described as p,d,q. In desk-bound time-collection information, the houses or cost of a pattern statement does now no longer rely on the timestamp at which its miles are observed. To cast off the correlation, ARIMA makes use of differencing to make the information desk-bound. ARIMA is super healthy to seize ancient trends, seasonality, randomness, and other non-static conduct that people miss.

B. Long Short-term Memory (LSTM)

A specific type of recurrent neural network called an LSTM is designed to accommodate time-series data. The LSTM is a more sophisticated version of the Recurrent Neural Network. Accuracy diminishes because the states of many hidden layers cannot be preserved in RNN. In the long run, LSTM can save cell state. Because previous days' stock prices can be held, it can effectively forecast the stock price. In the LSTM model, the input gate, forget gate, and output gate are used to verify previous states. It has a feedback line that solves the problem of vanishing gradients. The feedback line is capable of tracking the entire data sequence. This demonstrates outstanding performance on a wide range of successive data. A cell state, three gates, and an LSTM module allow them to selectively learn, unlearn, or retain information from each of the units. It has an output, a forget gate, and three inputs. Here, a sigmoid is used to create a number between 0 and 1. A forget gate erases the past value at a specific time step, and the result is passed through an activation function that returns 0 when the information should be erased or 1 when it must be retained for use in the future. The input gate is used to assess the significance of fresh information carried by the input, the output gate determines which information should be passed on to the next hidden state, and finally, the forget gate erases the past value. The vanishing gradient issue can be resolved via long short-term memory.

C. Bidirectional Long-short Term Memory (Bidirectional LSTM)

The RNN version is Long-Term Short-Term Memory (LSTM). To remember a particular part of the input sequence, the LSTM uses a combination of different RNNs. Sometimes it's important to remember a certain block of data, but that's not always the case, and there are ways to deal with that. When trying to recall specific calculations, LSTM is used. Estimates can be used to control how much he should keep. The RNN can also reasonably recall part of the previous input, but it has two disadvantages. Gradients fade and explode. Then LSTM - embedding a memory element called a cell in the array - was used to solve this problem. Any neural network may keep the sequence in both directions: backward (future to past) or forward (past to future), utilizing the bi-LSTM approach. Our input works bidirectionally with bidirectional LSTM, unlike conventional LSTM.

D. Hidden Markov Model (HMM)

HMM includes statistical models to capture confidential information from observable sequential emblems. In an HMM, the task is to extract the hidden parameters from the visible parameters in a system considered a Markov process with unknown parameters. According to this statement, an observed event will relate to a set of probability distributions rather than its step-by-step state. Making probabilistic models of problems involving the "labeling" of linear sequences is formally supported by hidden Markov models (HMMs). Additionally, they offer a conceptual toolkit for creating intricate models by creating an understandable image. As-transition probability, Emission probability, and starting probability are the parameters of the HMM model. In HMMs, learning entails estimating the state transition probabilities A and the output emission probabilities B that increase the likelihood of an observed sequence.

E. Transformer Model

The words in this statement serve as an example of how a transformer model learns context and meaning by examining relationships in sequential data. Transformer models apply an evolving set of mathematical techniques, known as attention or self-attention, to uncover the subtle ways in which even data items are distant from the chain of influence and interdependence. Transfiguration translates text and speech near real-time, opening meetings and classes to diverse and hearing-impaired participants.

F. Performance Matrices

Performance metrics are used to evaluate the quality of machine learning models. Here, Three different error matrices were used to evaluate the study such as Mean absolute error, Mean squared error(MSE), and Root Mean squared error(RMSE).

Below x and y are D dimensional vectors, and x_i denotes the value on the i th dimension of x .

$$\begin{aligned}\text{Mean Absolute Error(MAE)} &= \sum_{i=1}^D |x_i - y_i| \\ \text{Mean Squared Error(MSE)} &= \sum_{i=1}^D (x_i - y_i)^2 \\ \text{Root Mean Square Error (RMSE)} &= \sqrt{MSE}\end{aligned}$$

IV. DATASET

The dataset is collected from yahoo finance, which is a media property of Yahoo corporation. We used the 2022 Microsoft Corporation Stock dataset which is consisting of 9166 data. The dataset contains 7 features: date, high, low, opening price, closing price, Adj close, and volume. Our selected features will be 'date', and 'closing price'.

V. METHODOLOGY

ARIMA, LSTM, HMM, Transformer, and Bidirectional layers are used in this research. Adam optimizer and relu activation function are also used. Furthermore, epochs are set to 10 for LSTM and bidirectional LSTM. For experiment setup, the

python libraries such as Keras, Scikit-learn, Pandas, Numpy, and Matplotlib are utilized. These analyses are executing on the Google Colab GPU. Microsoft stock indexes are used.

A. Dataset Preprocessing

- **Eliminating Unnecessary Columns:**
There were several columns that were irrelevant to our work such as high, low, opening price, Adj value, and volume. We have eliminated those columns.
- **Null Value Handling:**
We have deleted the rows having null values. We have discarded all of that. Number of null values were so less. So there was any impact after deleting those columns in our dataset.

VI. RESULT AND ANALYSIS

We used ARIMA, LSTM, BiLSTM, HMM, and Transformer models to predict the stock market price and plot it to understand the difference between actual and predicted prices. We also figured out the data by performance matrix.

• ARIMA:

After applying the ARIMA model to our dataset, we get 5.467 MAE, and 44.539 MSE. In our research ARIMA model gives rmse 6.674, which is pretty good. 'AutoRegressive Integrated Moving Average' is a forecasting algorithm for stock market prediction. It has the capability to capture sudden ups and downs in stock price. ARIMA uses a moving average where random jumps to the time series plot that has an impact over two or more successive periods. These leaps show our ARIMA model's erroneous calculations and what the MA component would lag for. Similar to the exponential smoothing strategy, a model that uses MA would smooth out these abrupt jumps.

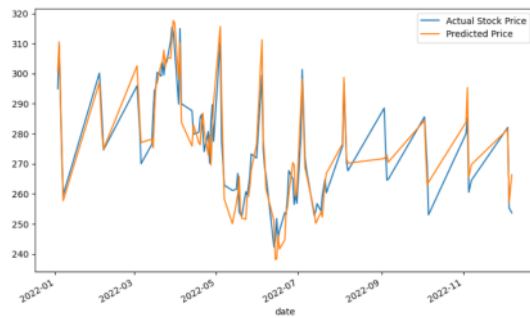


Fig. 1. Stock prediction by ARIMA

• LSTM:

After applying the LSTM model to our dataset,

we get 18.541 MAE, and 461.942 MSE. In our work, LSTM gives the rmse 21.493. So, LSTM could not perform well because our dataset is stationary, which means no trends or patterns concerning time. There are so many fluctuations in our dataset which couldn't be perfectly predicted by LSTM. LSTM works better when a considerable amount of sequential and pattern data is available. Time period from March 1986 to July 2022 is considered in which 13th March 1986 to 28th February 2022 is used for training and 3rd January 2022 to 22th July 2022 is used for testing. Here, we can see that the ARIMA model performs better than the LSTM model.

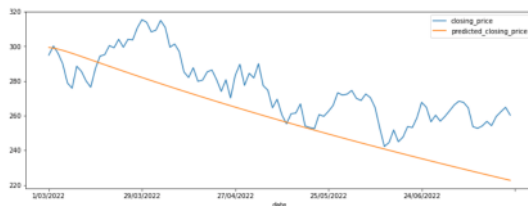


Fig. 2. Stock prediction by LSTM

- Bi-LSTM:

After applying the BiLSTM model to our dataset, we get 11.256 MAE, and 217.860 MSE. We have gained rmse 14.760 using Bi-directional LSTM gives better optimization and prediction result. The method of enabling any neural network to retain sequence information in both directions, either backward or forwards, is widely used as bidirectional long-short term memory (BiLSTM). Our input is bidirectional and flows in both directions, setting a BiLSTM apart from a conventional LSTM. To store past and future information of stock at any time step, we can have input flow in both directions. Here, we also can see that the BiLSTM model performs better than the LSTM model. Moreover, the ARIMA model gives better performance than BiLSTM and LSTM.

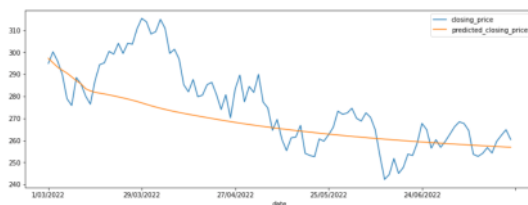


Fig. 3. Stock prediction by Bi-LSTM

- HMM:

HMM are Legos of computational sequence analysis. We have used three performance evaluation metrics on the HMM model. After applying HMM model to our dataset, we get 0.48553775 for MAE, 0.040804 for MSE, and 0.20203499 for RMSE. Due to inherent randomness, stock prediction is difficult. To identify hidden patterns, the Hidden Markov Model is employed for stock prediction. Here, the 10 hidden components (or states) of our data were quickly identified by the Hidden Markov model. The test data for the preceding 50 days will be used to test all potential outcomes for each of the days for which we will predict closing prices; we take the Cartesian product across a range of values for each feature to determine which sequence produces the highest score. The predictions for that day's closing price are then based on the result that receives the highest score. To quickly determine the level of the sequence and if the value increases the following day, hidden states and sequences have been constructed. Additionally, it was determined whether the amount is moderately high, high, or very high, as well as whether it is moderately low, low, or deficient. For both long-term and short-term investors, this model will be constructive.

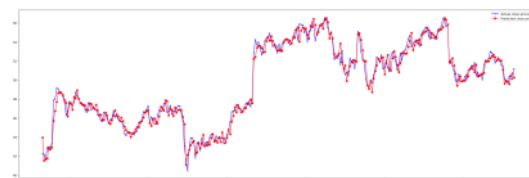


Fig. 4. Stock prediction by HMM

- Transformer Model:

After applying the transformer model, we get 0.09936 for MAE, 0.0234 for MSE, and 0.15309 for RMSE. Transformers are an excellent method for determining the semantic relationships between the components of a short or lengthy sequence. The transformer model is so strong for time series analysis in the case of both stationary and non-stationary data. It can capture trends as well as sudden ups and downs in the stock movement. In our transformer model, we used 60 attention heads, and the Embedding size for attention was 46. The number of transformer blocks was 5, which is well fitted for our ten thousand data. Adam optimizer and 0.14% dropout were used. The transformer model is made by a multi-headed attention mechanism. The attention mechanism was created to allow the decoder to employ the most pertinent parts of the input sequence in a flexible manner by combining all of the encoded input vectors in a weighted way, with the most relevant vectors obtaining the highest weights. A module for attention mechanisms called 'Multi-head Attention' cycles repeatedly and simultaneously through

an attention mechanism. The predicted dimension is then created by linearly combining the separate attention outputs.

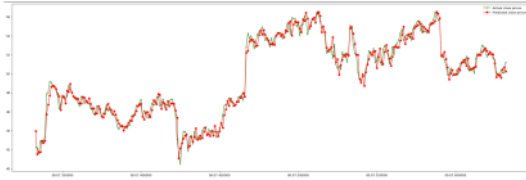


Fig. 5. Stock prediction by Transformer Model

The performance of different architectures are summarized in the table given below.

TABLE I
PERFORMANCE OF ALL THE MODELS.

| Model | MAE | MSE | RMSE |
|-------------|--------|---------|--------|
| Transformer | 0.099 | 0.023 | 0.153 |
| HMM | 0.486 | 0.041 | 0.202 |
| ARIMA | 5.467 | 44.539 | 6.674 |
| BiLSTM | 11.256 | 217.860 | 14.760 |
| LSTM | 18.541 | 461.942 | 21.493 |

In our research transformer model performs best. The Attention module of the Transformer performs its calculations repeatedly and concurrently. These are referred to as Attention Heads each. Attention models, also called attention mechanisms, are deep learning techniques that provide an additional focus on a specific component. In deep learning, attention relates to focusing on something in particular and noting its obvious importance. So multiple attention modules can easily predict the fluctuation in the time series. The Hidden Markov model also gives a good result with rmse 0.202. HMM offers a strong stochastic foundation and high intellectual methods, allowing acquisition to occur straight from the raw sequence analysis. It supports variable length inputs and enables for systematic processing of insertion and deletion costs in the context of selectively surmountable algorithms. Here we used the Gaussian Hidden Markov model with 10 hidden states. ARIMA model ranks third in our research with a rmse. ARIMA model has been used extensively in finance, having the qualities to be robust and efficient and offers a promising short-term share market forecast method. In comparing LSTM and Bidirectional LSTM, Bidirectional LSTM performs better than LSTM. The input flows in both directions, which makes it capable of utilizing information from both sides. So accuracy comes better.

Overall, we can say that the transformer model performs better for our data set. Also, we can write the sequence of

performance as:

Transformer > HMM > ARIMA > BiLSTM > LSTM

VII. LIMITATIONS AND FUTURE WORK

When we first started using the models, we ran into a number of restrictions. These shortcomings also show where further developments can be made.

Firstly, the dataset we selected for our research was not so enormous. Our approach was univariant. We only worked on a single company's datas. So, in future, we will work with multivariant approach. We will try to work with a extensive dataset. We will work with multiple companies to make it more convenient.

VIII. CONCLUSION

The price of a stock indicates its present value to buyers and sellers. Despite volatility, stock prices are more than just randomly generated numbers. Therefore, they can be analyzed as a series of discrete-time data. In other words, time series observations are made at consecutive points in time. Time series forecasting applies well to stock forecasting. The main topic is stock price prediction, and we have used LSTM, Bi-LSTM, HMM, ARIMA, and Transformer models. By treating stock market data as a time series, one can use past stock prices to forecast stock prices for the coming month. The advantage of stock price forecasting is that it reveals the future behavior of the market, always helping investors understand when and what types of stocks can be purchased to grow their investment.

11

REFERENCES

- [1] D. Wei, "Prediction of stock price based on lstm neural network," in *2019 International Conference on Artificial Intelligence and Advanced Manufacturing (AIAM)*, pp. 544–547, IEEE, 2019.
- [2] T. Gao, Y. Chai, and Y. Liu, "Applying long short term memory neural networks for predicting stock closing price," in *2017 8th IEEE international conference on software engineering and service science (SESS)*, pp. 575–578, IEEE, 2017.
- [3] G. Bathla, "Stock price prediction using lstm and svr," in *2020 Sixth International Conference on Parallel, Distributed and Grid Computing (PDGC)*, pp. 211–214, IEEE, 2020.
- [4] A. Moghar and M. Hamiche, "Stock market prediction using lstm recurrent neural network," *Procedia Computer Science*, vol. 170, pp. 1168–1173, 2020.

Group1_SimilarityReport

ORIGINALITY REPORT

17%

SIMILARITY INDEX

11%

INTERNET SOURCES

13%

PUBLICATIONS

9%

STUDENT PAPERS

PRIMARY SOURCES

- | | | |
|--|--|---|
| <div style="background-color: red; color: white; width: 40px; height: 40px; display: flex; align-items: center; justify-content: center; margin: 5px;">1</div> | <p style="color: red;">Gourav Bathla. "Stock Price prediction using LSTM and SVR", 2020 Sixth International Conference on Parallel, Distributed and Grid Computing (PDGC), 2020</p> <p style="color: gray; font-size: 0.8em;">Publication</p> | <p style="color: red; font-size: 2em;">2%</p> |
| <hr/> | | |
| <div style="background-color: magenta; color: white; width: 40px; height: 40px; display: flex; align-items: center; justify-content: center; margin: 5px;">2</div> | <p style="color: magenta;">Dou Wei. "Prediction of Stock Price Based on LSTM Neural Network", 2019 International Conference on Artificial Intelligence and Advanced Manufacturing (AIAM), 2019</p> <p style="color: gray; font-size: 0.8em;">Publication</p> | <p style="color: magenta; font-size: 2em;">1%</p> |
| <hr/> | | |
| <div style="background-color: purple; color: white; width: 40px; height: 40px; display: flex; align-items: center; justify-content: center; margin: 5px;">3</div> | <p style="color: purple;">Submitted to Athlone Institute of Technology</p> <p style="color: gray; font-size: 0.8em;">Student Paper</p> | <p style="color: purple; font-size: 2em;">1%</p> |
| <hr/> | | |
| <div style="background-color: teal; color: white; width: 40px; height: 40px; display: flex; align-items: center; justify-content: center; margin: 5px;">4</div> | <p style="color: teal;">journalofbigdata.springeropen.com</p> <p style="color: gray; font-size: 0.8em;">Internet Source</p> | <p style="color: teal; font-size: 2em;">1%</p> |
| <hr/> | | |
| <div style="background-color: green; color: white; width: 40px; height: 40px; display: flex; align-items: center; justify-content: center; margin: 5px;">5</div> | <p style="color: green;">Md. Anisur Rahman Mia, Mohammad Abu Yousuf, Rupon Ghosh. "Business Forecasting System using Machine Learning Approach", 2021 2nd International Conference on Robotics, Electrical and Signal Processing Techniques (ICREST), 2021</p> <p style="color: gray; font-size: 0.8em;">Publication</p> | <p style="color: green; font-size: 2em;">1%</p> |
-

6

Submitted to Universiti Teknologi Malaysia

Student Paper

1 %

7

ijircce.com

Internet Source

1 %

8

www.indeed.com

Internet Source

1 %

9

Tingwei Gao, Yueting Chai, Yi Liu. "Applying long short term momory neural networks for predicting stock closing price", 2017 8th IEEE International Conference on Software Engineering and Service Science (ICSESS), 2017

Publication

1 %

10

Adil Moghar, Mhamed Hamiche. "Stock Market Prediction Using LSTM Recurrent Neural Network", Procedia Computer Science, 2020

Publication

1 %

11

Ashkan Safari. "Data Driven Artificial Neural Network LSTM Hybrid Predictive Model Applied for International Stock Index Prediction", 2022 8th International Conference on Web Research (ICWR), 2022

Publication

1 %

12

www.science.gov

Internet Source

1 %

| | | |
|----|---|------|
| 13 | aclanthology.org Internet Source | 1 % |
| 14 | mobt3ath.com Internet Source | 1 % |
| 15 | www.tuba.gov.tr Internet Source | <1 % |
| 16 | Thomas, Sonia, G.N. Pillai, Kirat Pal, and Pushpak Jagtap. "Prediction of ground motion parameters using randomized ANFIS (RANFIS)", Applied Soft Computing, 2016. Publication | <1 % |
| 17 | Dilip Singh, Bhupendra Kumar Gupta. "Chapter 37 Closing Price Prediction of Nifty Stock Using LSTM with Dense Network", Springer Science and Business Media LLC, 2022 Publication | <1 % |
| 18 | www.hindawi.com Internet Source | <1 % |
| 19 | Amr Zeedan, Khaled Al-Fakhroo, Abdulaziz Barakeh. "EEG-Based Seizure Detection Using Feed- Forward and LSTM Neural Networks Based on a Neonates Dataset", Institute of Electrical and Electronics Engineers (IEEE), 2022 Publication | <1 % |

- | | | |
|----|---|------|
| 20 | Pawee Werawithayaset, Suratose Tritilanunt. "Stock Closing Price Prediction Using Machine Learning", 2019 17th International Conference on ICT and Knowledge Engineering (ICT&KE), 2019 Publication | <1 % |
| 21 | towardsdatascience.com Internet Source | <1 % |
| 22 | Submitted to Florida International University Student Paper | <1 % |
| 23 | iwaponline.com Internet Source | <1 % |
| 24 | Ho, Y.C.. "Comparing ANFIS and SEM in linear and nonlinear forecasting of new product development performance", Expert Systems With Applications, 201106 Publication | <1 % |
| 25 | frontend.lidsen.com Internet Source | <1 % |
| 26 | www.frontiersin.org Internet Source | <1 % |
| 27 | Hendri, Antoni Wibowo*, Rina Novita Sari. "Timeseries Forecasting using Long Short-Term Memory Optimized by Multi Heuristics Algorithm", International Journal of Recent Technology and Engineering (IJRTE), 2019 | <1 % |

Publication

| | | | |
|----------------------|-----|-----------------|-----|
| Exclude quotes | Off | Exclude matches | Off |
| Exclude bibliography | Off | | |