Analysis and forecasting of Time-Series data using S-ARIMA, CNN, and LSTM

Subhash Arun Dwivedi
subhashaks95@gmail.com
Sharda University, India

Amit Attry
2018012875.amit@ug.sharda.ac.in
Sharda University India

Darshan Parekh
darshparekh03@gmail.com
Sharda University, India

Kanika Singla
kanika.singla@sharda.ac.in
Sharda University, India

Abstract—**Analyzing the behavior of stock market movements has often been an area of interest to machine learning and time-series data analyst. It has been very challenging due to its immense complex nature, chaotic, and dynamic environment. With the advent of machine learning and deep learning algorithms, this paper aims to significantly reduce the risk of trend prediction. This study compares models for Time – Series forecasting i.e. SARIMA (Seasonal Auto-Regressive Integrated Moving Average), CNN (Convolutional Neural Network), and LSTM (Long Short-Term Memory) for predicting Nifty-500 indices trend. The results that were obtained are promising and the evaluation unveils the power of Deep Learning through CNN and LSTM but also empowers the S-ARIMA model, making a great impact on the Machine Learning paradigm.**

Keywords—Stock Market Prediction; Time – Series Forecasting; S-ARIMA; CNN; LSTM.

I. INTRODUCTION

The major factor which encapsulates Stock Market is "Prediction"**[6].** The old-school predicting methodologies aren't reliable as the risk factor can behave exponentially; as in the past stock market trends were typically forecasted by financial analysts specifically "Brokers" **[7].** Although many evaluators of the effective market hypothesis think that it is difficult to reliably forecast stock prices, there are systematic proposals **[17]** that show that precise modeling and design of acceptable variables will contribute to models that can be quite accurately predicted using stock prices and stock price movement trends.

However, with the advent of learning methods, the role of data scientists came into action addressing the financial current and thereby implying logic and analytic outlook for forecasting the trend of any listed companies over any Stock Exchange. To increase accuracy rates, they tried and tested various Machine Learning Models and thereby improved the accuracy of trend prediction. As a result, Deep Learning Models gave an edge to the forecasters improving the accuracy rates.

In this regard **[18]**, another line of work has been proposed by the authors which used decomposition of time series data. A highly robust and reliable productive framework has been presented by the authors **[10]** stock prediction analysis by leveraging text mining and natural language processing.

Inspired by the current success of methods in machine learning and deep learning for analysis and prediction **[8],** this current work proposed the gamut of both models. The historical index values of NIFTY 500 for January 2000 till March 2020 has been used as the training dataset.

The paper has been divided into different sections which are as follows:

1. **Section 1:** This section contains the introduction to the paper, aim, motivation, and objectives of the problem statement.
2. **Section 2:** In this section, we discuss the related work or the literature survey of the concepts used during the research.
3. **Section 3**: This section includes methods to achieve the objectives of the research problem.
4. **Section 4**: This section includes implementation details tools used and different python frameworks required to achieve the objectives of the research problem.
5. **Section 5**: Performance metrics and experimental analysis has been conducted in this section respectively for the S-ARIMA model, Convolutional Neural network **(CNN)** model, and Long Short-Term Memory **(LSTM)** model.
6. **Section 5:** This section contains the analysis and discussions about the result, contribution of the work to the existing research.
7. **Section 6:** Possible future scope has also been discussed in this section.

II. LITERATURE SURVEY

The models used in this project have been widely engulfed in various analyses and predictions. S-ARIMA model has been integrated into Out-patient visits forecasting **[1],** Aircraft Failure Rate **[2]**, Prediction of Crop Yield using Temperature and Rainfall parameters by the concept of Fuzzy logic **[3],** finding out the count of people due to road traffic accident on highways per hour **[4],** Hydrological Time-Series Forecast **[5]**, etc. Comparing the Root Mean Squared Error (RMSE) and Mean Absolute Percentage Error (MAPE) error values between the actual and forecast results, the performance model is tested. Convolutional Neural Network (CNN), Long Short-Term Memory (LSTM) models **[12]** have been proved to show a crucial difference in growth in Time-Series analysis due to their hiked complexity and better tuning.

A line of work has been proposed **[6,9]** by the authors which are based on CNN that predicts the movement of the price of stocks. They created an ensemble stock predictor for NSE India using RNN, LSTM, and CNN **[11].** Further, research was done upon Stock Prediction using Technical indicators and financial news articles thereby generating an output through RNN, LSTM, and CNN models **[14].**

The integration of American Stock Exchange and NASDAQ Composite for forecasting their trend using Stacked LSTM Networks **[15]** but the volatility of stock market sustains being the main hindrance in future prediction. Later, some of the researchers proposed the Stock Ensemble-based Neural Network (SENN) model and incorporated it over Boeing historical stock data as well as sentiment score extracted from Stock Twits microblog text data in 2019, showing great results and outperforming similar architecture models **[20].** These results prove that, when stock market volatility is overlooked, results we get from Deep Neural Network models **[13]** are exhilarating in future trend prediction.

III. METHODOLOGY USED FOR ANALYSIS

Evolving through various methodologies it's clear that there exist, models, that are exemplary for Time – Series Analysis in Machine Learning, as well as Deep Learning, approaches. The proposed methodology in this project revolves around SARIMA, CNN, and LSTM models which are highly competent for Forecasting.

A. **Seasonal Auto-Regressive Integrated Moving Average (SARIMA):** ARIMA model describes how each successive observation is related to the previous observation.

ARIMA processes $\{X_t\}$

Let,

$$Y_t = \nabla^d X_t \qquad (1)$$

∇ = Difference Operator,

d = Order of Differencing,

X_t = ARIMA

Then,

$$Y_t = \emptyset_1 Y_{t-1} + \emptyset_2 Y_{t-2} + \ldots + \emptyset_p Y_{t-p} + Z_t + \theta_1 Z_{t-1} + \cdots \theta_q Z_{t-q} \qquad (2)$$

Can be written as,

$$\emptyset(B) Y_t = \theta(B) Z_t \qquad (3)$$

Auto-Regressive Polynomial = Moving Average Polynomial

Where,

$$\theta(B) = 1 + \theta_1 B + \cdots + \theta_q B^q \qquad (4)$$

$$\emptyset(B) = 1 - \emptyset_1 B - \emptyset_2 B^2 - \cdots - \emptyset_p B^p \qquad (5)$$

Data might contain a seasonal periodic component in addition to a correlation with recent lags. It repeats every s observations. For a time series of monthly observations, X_t might depend on annual lags i.e. X_{t-12}, X_{t-24},...This state's Seasonal ARIMA model.

SARIMA (p, d, q)(P, D, Q)$_m$ **(6)**

p – denotes trend autoregressive order,

d – denotes trend difference order,

q – denotes trend moving average order,

P – denotes the number of seasonal autoregressive terms,

D – denotes the number of seasonal difference terms,

Q – denotes the number of seasonal moving average terms.

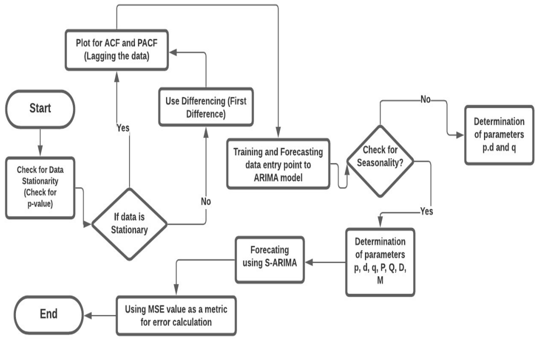

Plot for ACF and PACF (Lagging the data)
Start
Yes
Use Differencing (First Difference)
No
Determination of parameters p.d and q
Check for Data Stationarity (Check for p-value)
Training and Forecasting data entry point to ARIMA model
Check for Seasonality?
No
If data is Stationary
Yes
Determination of parameters p, d, q, P, Q, D, M
Forecating using S-ARIMA
End
Using MSE value as a metric for error calculation

Fig .1: Working of S-ARIMA

B. **Convolutional Neural Network (CNN):** CNN models are extremely vivid, as well as complex in nature and, are composed of the following layers:

Convolution Layer (CONV): The convolution layer inculcates filters that perform convolution operations as it scans input *I* concerning its dimensions. The hyperparameters include the filter size *F and* stride *S.* The resultant output *O* is called Feature Map or Activation Map.

Pooling (POOL): To reduce the number of parameters and computation in the network, its purpose is to gradually reduce the spatial size of the representation. The pooling layer works independently on each function map.

Fully connected (FC): Fully connected layer is the last layer that operates on a flattened input where each of the inputs is connected to all neurons, which typically has a sigmoid activation function or softmax function.

Stride(S): Strides is a filter hyperparameter. It denotes the number of pixels the window will pass thro ugh after each operation.

Via an asterisk * symbol, convolution is mainly expressed mathematically. If we have an input X and filter as *f* then the expression would be:

$$Z = X * f \quad \textbf{(7)}$$

1D - Convolution is given as,

$$h_i = f((W*x)_i) \quad \textbf{(8)}$$

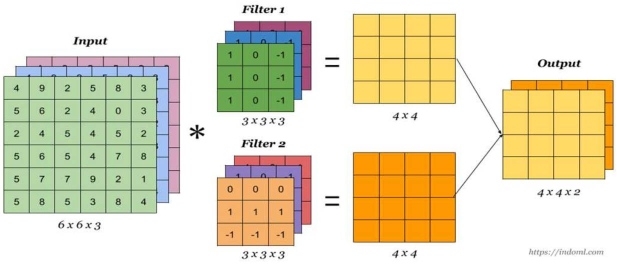


Fig .2: Convolutional Neural Network

The CNN layer may include a sub-sampling layer that reduces noise in learned features i.e. the feature maps. This layer is followed by a regression layer.

C. **Long Short–Term Memory (LSTM):** A key shortcoming of CNNs for time series data is that they do not secure information from the sequential nature of independent data and this drawback is nullified through the LSTM network i.e. it remembers what it has seen. The heart of LSTM is its cell state which helps it to do so. In LSTMs we have three gates i.e.,

$$i_t = \sigma\left(w_i\left[h_{t-1}, x_t\right] + b_i\right)\ldots\text{Input Gate} \quad \textbf{(9)}$$

$$f_t = \sigma\left(w_f\left[h_{t-1}, x_t\right] + b_f\right)\ldots\text{Forget Gate} \quad \textbf{(10)}$$

$$o_t = \sigma\left(w_o\left[h_{t-1}, x_t\right] + b_o\right)\ldots\text{Output Gate} \quad \textbf{(11)}$$

where,

$\sigma =$ It represents a sigmoid function

$w_x =$ It's the weight for the respective gate (x) neurons.

$h_{t-1} =$ The output of the previous LSTM block (at timestamp t – 1)

$x_t =$ The input at the current timestamp

$b_x =$ The biases for the respective gate (x)

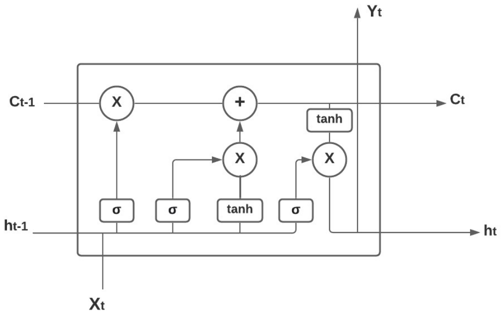


Fig –3: ***Long Short Term Memory Working***

IV. IMPLEMENTATION AND TOOLS

A. **Dataset used and Data Source:** Nifty-500 (This index sits at the top. It represents the top 500 companies focused on full market capitalization. The dataset comprises NIFTY-500 index values over the active period. It has been retrieved through CEIC - Census and Economic Information Center.

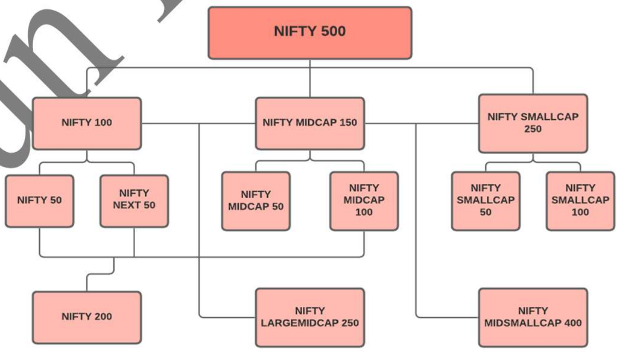


Fig – 4: NIFTY-500 Hierarchy

B. ***Frameworks used:*** *Tensorflow and Keras*

Tensorflow is an open-source machine learning framework. It's extremely useful for numerical computation and large-scale machine learning Keras is a high-level API for the neural network, capable of running on top of Tensorflow, enabling fast experiments.

C. **Making the Data Stationary:** Stationarity is a property of time series data stating that the distributional property (mean and standard deviation) of the data series has not changed across time. For forecasting, the data must be stationary because, in the absence of stationarity, one is asking the model to predict data that is nothing like anything it has seen before.

A common test of stationarity is the Dickey-Fuller Test.
If the p-value associated with the Dickey-Fuller test statistic is greater than 0.05, we state that the data is not stationary.

D. **Lagging the Data:** Next, because sequential/time series data is autoregressive - i.e. the outcome today depends on the outcome yesterday and the outcome the day before yesterday and so on - we need to create lagged versions of each independent variable.

The maximum number of lags of the dependent variable to use can be decided from the Autocorrelation Function (ACF) and/or Partial Autocorrelation Function (PACF) plot or decided heuristically based on domain-specific cycles (E.g.: a business cycle, seasons, etc). The maximum number of lags of other independent variables to include is rather arbitrary - can be decided heuristically based on domain-specific cycles (E.g.: a business cycle, seasons, etc).

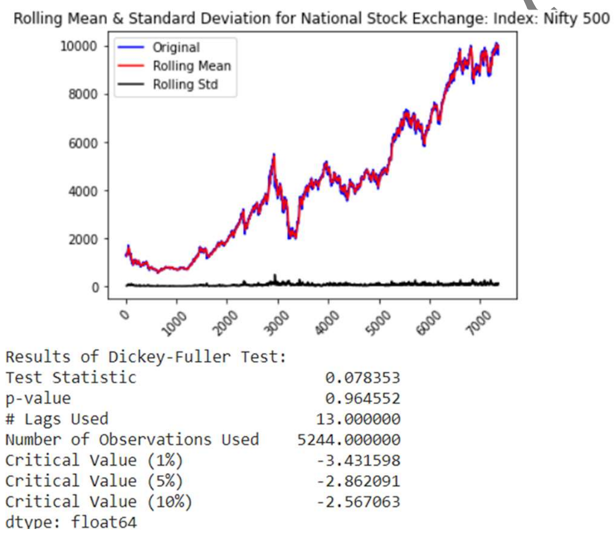
The minimum number of lags to include depending on the forecasting horizon. If you want to forecast h steps, you exclude the first h lags.

E. **Metric to be used to gauge the accuracy of forecasts**: The most commonly used metric for gauging the accuracy of time series forecasts is the Mean Squared Error (MSE)

The higher the MSE, the worse is the predictive accuracy of the model. Thus for each model, it is desired that the MSE be minimized.

V. EXPERIMENTAL RESULTS AND ANALYSIS

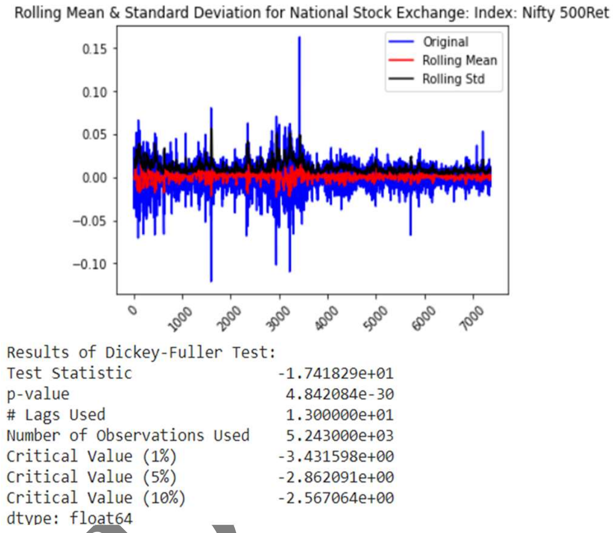
Primarily, as above reported, we have to test for data stationarity as we're dealing with Time Series Forecasting. So using **Dicky-Fuller Test we tested Nifty-500 data's stationarity.**



```
Results of Dickey-Fuller Test:
Test Statistic                    0.078353
p-value                           0.964552
# Lags Used                      13.000000
Number of Observations Used    5244.000000
Critical Value (1%)              -3.431598
Critical Value (5%)              -2.862091
Critical Value (10%)             -2.567063
dtype: float64
```

Fig – 4: ***Non – Stationary Data Representation(Dickey-Fuller Test)***

From the above results, we see that the **p-value is 0.964552**, leading to the conclusion that the Nifty-500 Index series is not stationary. **The easiest way of making the data stationary is to calculate the first difference.**



```
Results of Dickey-Fuller Test:
Test Statistic                 -1.741829e+01
p-value                         4.842084e-30
# Lags Used                     1.300000e+01
Number of Observations Used     5.243000e+03
Critical Value (1%)            -3.431598e+00
Critical Value (5%)            -2.862091e+00
Critical Value (10%)           -2.567064e+00
dtype: float64
```

Fig – 5: ***Visualization of Stationary Data***

From the above graphs, we see that the percentage change in the Nifty 500 Index (i.e. National Stock Exchange: Index: Nifty 500Ret) is stationary as the p-value is less than 0.05.
Now for lagging the data, **ACF and PACF** are plotted.

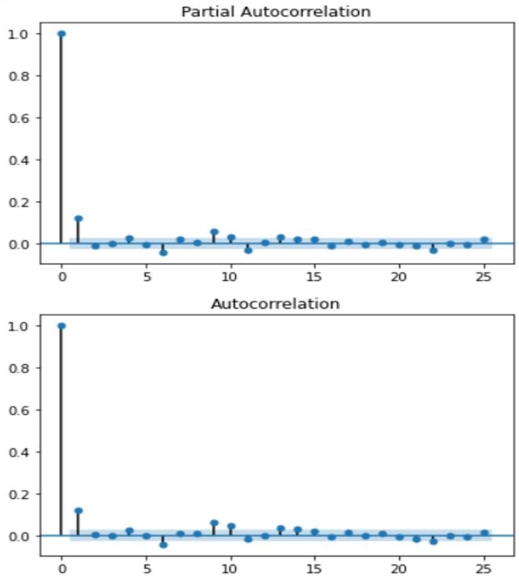
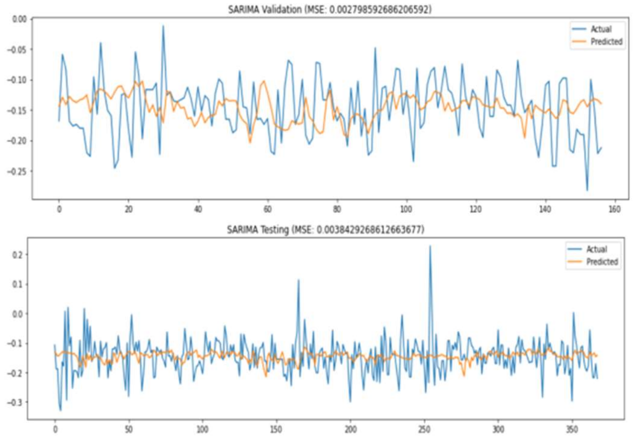


Fig – 6: ***PACF and ACF Plot***

Both the ACF and PACF plots show high serial correlation at the first lag.

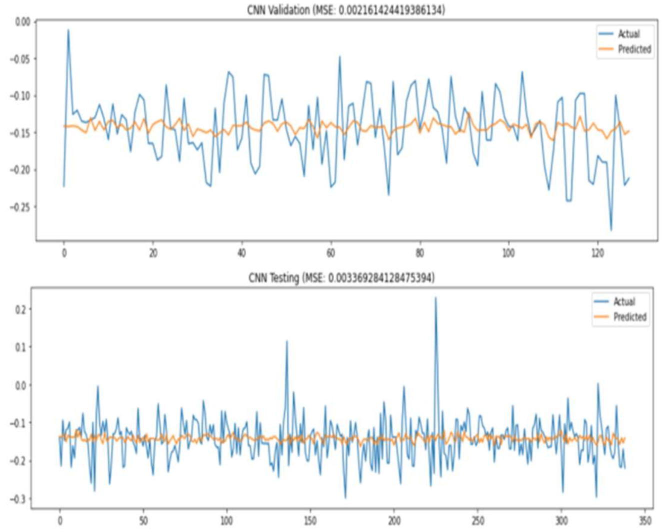
Since we're going to forecast the one-day Nifty stock returns, the minimum lag considered by me is1. Now after lagging the data, we split the data into Training, Validating, and Testing Data and then rescaled the data. Then we scale it to being between -1 and 1 as that is the appropriate scaling for data when being input into a Convolutional Neural Network (CNN) and Long Short Term Memory Network (LSTM).

Now let's see the MSE results of the models:


SARIMA Validation (MSE: 0.002798592686206592)

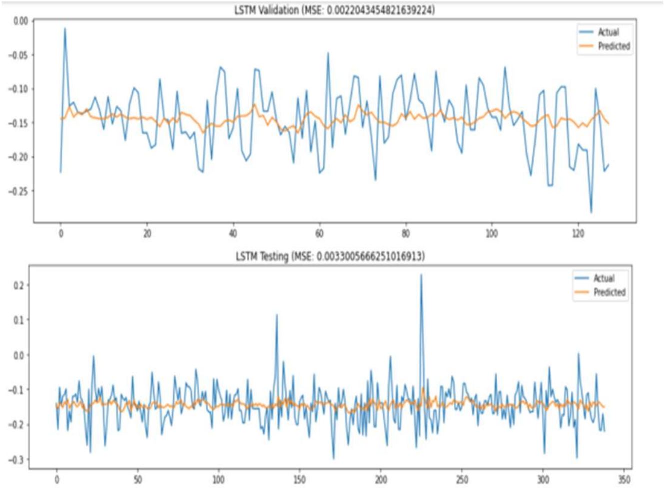
SARIMA Testing (MSE: 0.0038429268612663677)

Fig – 7: *SARIMA Result*


CNN Validation (MSE: 0.002161424419386134)

CNN Testing (MSE: 0.003369284128475394)

Fig – 8: *CNN Result*


LSTM Validation (MSE: 0.0022043454821639224)

LSTM Testing (MSE: 0.0033005666251016913)

Fig – 9:*LSTM Result*

Model	Validation MSE	Testing MSE
SARIMA	0.002798	0.003842
CNN	0.002161	0.003369
LSTM	0.002204	0.003300

TABLE –1: *Comparision of Results Obtained shows MSE results over Validation Data and Test Data.*

VI. CONCLUSION AND FUTURE SCOPE

In this paper, we have presented several approaches to predict stock index values and their movement patterns using S-ARIMA and two deep learning models such as CNN, LSTM. These models have been exploited on the NIFTY-500 dataset. It is a daunting challenge to be competitive in the stock market. Based on historical stock data, future stock price prediction mainly falls into the technical domain whereas the evaluation of the securities evaluations is derived from the statistics produced by market activities.

Through the results, we observe that Deep Learning Models outperform Machine Learning Model. The performance of the LSTM-based deep learning regression models was found to be much too superior to that of the machine-learning-based predictive models across both the machine learning and deep learning-based regression models.

The research has conclusively proven our assumption that deep learning-based models are far more capable of extracting and learning the characteristics of time series data than their corresponding counterparts in machine learning.

Generative adversarial networks (GAN's) may be investigated as future scope for time series analysis and forecasting of stock prediction.

REFERENCES

[1] Xinxiang, Z., Bao, Z., & Huijuan, F. (2017). A comparison study of outpatient visits forecasting effect between ARIMA with seasonal index and SARIMA. 2017 International Conference on Progress in Informatics and Computing (PIC).

[2] Yang, Y., Zheng, H., & Zhang, R. (2017). Prediction and analysis of aircraft failure rate based on the SARIMA model. 2017 2nd IEEE International Conference on Computational Intelligence and Applications (ICCIA).

[3] Bang, S., Bishnoi, R., Chauhan, A. S., Dixit, A. K., & Chawla, I. (2019). Fuzzy Logic-based Crop Yield Prediction using Temperature and Rainfall parameters predicted through ARMA, SARIMA, and ARMAX models. 2019 Twelfth International Conference on Contemporary Computing (IC3).

[4] Chanpanit, T., Arkamanont, N., & Pranootnarapran, N. (2019). Predicting the Number of People for Road Traffic Accident on Highways by Hour of Day. 2019 8th International Conference on Industrial Technology and Management (ICITM).

[5]Wang, Z., & Lou, Y. (2019). Hydrological time series forecast model based on wavelet de-noising and ARIMA-LSTM. 2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC).

[6] A. Tsantekidis, N. Passalis, A. Texas, J. Kanniainen, M. Gabbouj, and A. Iosifidis, "Forecasting Stock Prices from the Limit Order Book Using Convolutional Neural Networks," 2017 IEEE 19th Conference on Business Informatics (CBI), Thessaloniki, 2017, pp. 7-12, DOI: 10.1109/CBI.2017.23.

[7] J. Wang, T. Sun, B. Liu, Y. Cao, and D. Wang, "Financial Markets Prediction with Deep Learning," 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA), Orlando, FL, 2018, pp. 97-104, DOI: 10.1109/ICMLA.2018.00022.

[8]R. Zhang, Z. Yuan, and X. Shao, "A New Combined CNN-RNN Model for Sector Stock Price Analysis," 2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC), Tokyo, 2018, pp. 546-551, DOI: 10.1109/COMPSAC.2018.10292

[9] S. Selvin, R. Vinayakumar, E. A. Gopalakrishnan, V. K. Menon, and K. P. Soman, "Stock price prediction using LSTM, RNN and CNN-sliding window model," 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Udupi, 2017, pp. 1643-1647, DOI: 10.1109/ICACCI.2017.8126078.

[10] Mehtab, S., & Sen, J. (2019). A robust predictive model for stock price prediction using deep learning and natural language processing. Available at SSRN 3502624.

[11] Hegde, M. S., Krishna, G., & Srinath, R. (2018). An Ensemble Stock Predictor and Recommender System. 2018 International Conference on Advances in Computing, Communications, and Informatics (ICACCI).

[12] Preeti, R. Bala, and R. P. Singh, "Financial and Non-Stationary Time Series Forecasting using LSTM Recurrent Neural Network for Short and Long Horizon," 2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Kanpur, India, 2019, pp. 1-7, DOI: 10.1109/ICCCNT45670.2019.8944624.

[13] Y. Liu, Z. Su, H. Li, and Y. Zhang, "An LSTM based classification method for time series trend forecasting," 2019 14th IEEE Conference on Industrial Electronics and Applications (ICIEA), Xi'an, China, 2019, pp. 402-406, DOI: 10.1109/ICIEA.2019.8833725.

[14] M. R. Vargas, C. E. M. dos Anjos, G. L. G. Bichara and A. G. Evsukoff, "Deep Leaming for Stock Market Prediction Using Technical Indicators and Financial News Articles," 2018 International Joint Conference on Neural Networks (IJCNN), Rio de Janeiro, 2018, pp. 1-8, DOI: 10.1109/IJCNN.2018.8489208.

[15] Ojo, S. O., Owolawi, P. A., Mphahlele, M., & Adisa, J. A. (2019). Stock Market Behaviour Prediction using Stacked LSTM Networks*. 2019 International Multidisciplinary Information Technology and Engineering Conference (LIMITED).

[16] Gold Price Forecast based on LSTM-CNN Model Zhanhong He∗, Junhao Zhou∗, Hong-Ning Dai∗, Hao Wang†,2019 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress.

[17] C. Y. Lai, R. Chen and R. E. Caraka, "Prediction Stock Price Based on Different Index Factors Using LSTM," 2019 International Conference on Machine Learning and Cybernetics (ICMLC), Kobe, Japan, 2019, pp. 1-6, doi: 10.1109/ICMLC48188.2019.8949162.

[18] Sen, J. and Datta Chaudhuri, T.: A Predictive Analysis of the Indian FMCG Sector Using
Time Series Decomposition-Based Approach. Journal of Economics Library, 4(2), 206 –226 (2017)

[19] S. Alhazbi, A. B. Said and A. Al-Maadid, "Using Deep Learning to Predict Stock Movements Direction in Emerging Markets: The Case of Qatar Stock Exchange," 2020 IEEE International Conference on Informatics, IoT, and Enabling Technologies (ICIoT), Doha, Qatar, 2020, pp. 440-444, doi: 10.1109/ICIoT48696.2020.9089616..

[20] L. Owen and F. Oktariani, "SENN: Stock Ensemble-based Neural Network for Stock Market Prediction using Historical Stock Data and Sentiment Analysis," 2020 International Conference on Data Science and Its Applications (ICoDSA), Bandung, Indonesia, 2020, pp. 1-7, doi: 10.1109/ICoDSA50139.2020.9212982.