International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 08 Issue: 03 | March - 2024 SJIF Rating: 8.448 ISSN: 2582-3930

Stock Market Price Prediction and Forecasting Using Stacked LSTM

Supriya Raut
Department of Computer Engineering
Vidyalankar Institute of Technology
Mumbai, India

Prof. Avinash Shrivas
Department of Computer Engineering
Vidyalankar Institute of Technology
Mumbai, India

---***---

Abstract - Stock price movement is non-linear and complex. Several research works have been carried out to predict stock prices. Traditional approaches such as Linear Regression and Support Vector Regression were used but accuracy was not adequate. Researchers have tried to improve stock price prediction using ARIMA. Due to very high variations in stock prices, deep learning techniques are applied due to its proven accuracy in various analytics fields. Artificial Neural Network was deployed to predict stock prices but as stock prices are time-series based, recurrent neural network was applied to further improve prediction accuracy. Therefore, data scientists and analysts found that Deep Learning outperformed Machine Learning which is also proofed by all the collected research papers, and it is the most suitable methodologies to apply to the stock market forecasting domain.

This paper explores different stacked LSTM Models for non-stationary financial time series in stock price prediction. This study is to predict stock market prices to make more acquaint and precise investment decisions. The experimental result will show that, this method can get quite accurate result, particularly effective in stock prediction. The proposed LSTM model will be designed to overcome gradient explosion, gradient vanishing, and save long-term memory. Firstly, The Model will have a comparative analysis between different LSTM models and integrating the model based on the result obtained from the analysis. The results will suggest that the developed stacked LSTM produces better predictive power and generalization.

***Key Words*:** *stock market prediction, stacked LSTM, Deep Learning, Data pre-processing techniques, Data normalization, Neural Networks, Data Visualization, Training and Testing Set, Time series analysis, Predictive models, Future Forecasting.*

1. INTRODUCTION

The stock market can be described as a highly volatile entity, with great sensitivity to various types of parameters. Much research has been conducted in the way of stock market prediction, with various models associated with time series analysis being adopted for this purpose. In economics, theories such as the efficient market hypothesis and the random walk theory challenge the thought that future stock price can be predicted using historic stock price data [1] [2]. With the availability of historic and current stock market data, researchers have attempted to predict the future state of the stock price through models that can analyze historic stock market data. Typically, this historic data consists of opening, closing, high, and low prices, and volume. Using various time series analysis models, daily data can be analyzed and the future state of the stock market can be predicted.

Machine Learning (ML) techniques have gained popularity more recently for time series analysis and stock market prediction, due to their ability to predict future states given large historical datasets [3]. Time series analysis is divided into two types:

• **Univariate Analysis -** The analysis of a time series which has a single input parameter, over equally spaced time intervals.

• **Multivariate Analysis -** The analysis of a time series with multiple input variables which vary over time.

Stock market analysis can be considered as Multivariate due to the multiple parameters, which vary over time, involved in its analysis. These parameters include, but are not limited to, time, opening price, closing price, among others. For an investor/trader these parameters have an impact on their decision making.

This study looks at various approaches to time series analysis that have gained popularity over the years, and reviews literature focusing on stock price behavior prediction. This study is to implement a model for Stock Price prediction by making use of different stacked LSTM (Long Short-Term Memory) networks and a comparative analysis between different LSTM networks such as LSTM and Stacked LSTM along with the evaluated result by using different performance measures proving which the best is and the most efficient network.

2. LITERATURE REVIEW

Time series forecasting is not a new area of research. A number of techniques have gained popularity within the sphere of time series analysis and forecasting, particularly for stock market prediction. Some of these include:

2.1. Support Vector Machines

The Support Vector Machine (SVM) algorithm has also had success in time series forecasting. It is a supervised learning technique which classifies training samples in a hyper plane by separating samples into one class or the other [6]. In a 2-D plane, this would be a line which assigns training samples to one of two classes, on either side of the line. This divider is known as a decision boundary.

The decision boundary can be depicted by the following equation [7]:

$$w.x + b = 0 \tag{1}$$

where x represents a point on the decision boundary, w is a n-dimensional vector at 90o to the decision boundary, and b is a constant. SVMs are also popular for their ability of avoid the problem of overfitting [8].

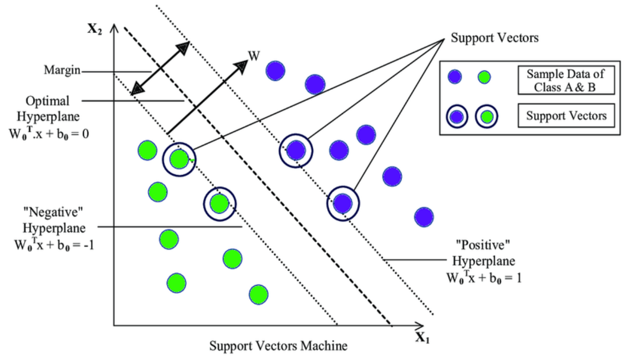

Figure 1. Support Vector Machine

2.2. Recurrent Neural Networks

A Recurrent Neural Networks (RNNs) is a type of ANN which make use of internal hidden neurons. A longstanding problem with RNNs is the vanishing gradient problem in which data is lost during the learning process as the internal state of an RNN is updated, recursively. The update of a RNN's state can be represented by the following equation:

$$h_t = \sigma(Wx_t + Uh_{t-1} + b) \tag{2}$$

where N represents the number of neurons in the RNN, *xt ∈ RM* is the input state and *ht ∈ RN* is the hidden state at a time t. *W ∈ R^NxM* is the weight of the current input, *U ∈ R^NxN* is the weight of the recurrent input, and *b ∈ R^N* is the bias. σ is an activation function.

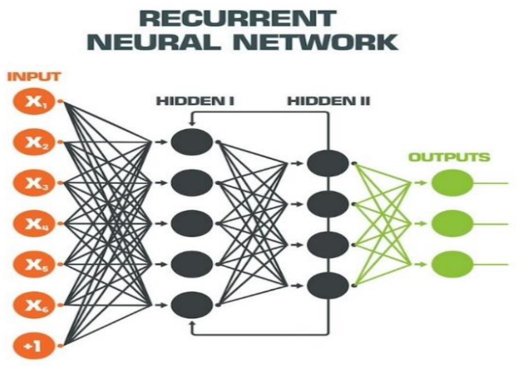

Figure 2. Recurrent Neural Network

2.3. Multilayer Perceptron

Multilayer Perceptron (MLP) is a type of Artificial Neural Network (ANN), made up of at least 3 layers namely, an input layer, a hidden layer, and an output layer. It makes use of supervised learning through back propagation for training [9]. Training of MLPs is achieved through a self-correcting approach in which the result of each hidden layer (known as a weight) is back propagated and these weights updated. Iteratively to improve the prediction performance of the network.

The most common means of evaluating the performance of an ANN is by calculating the Mean Squared Error (MSE) and Mean Absolute Deviation (MAD). Figure three illustrates a MLP with 4 Hidden layers.

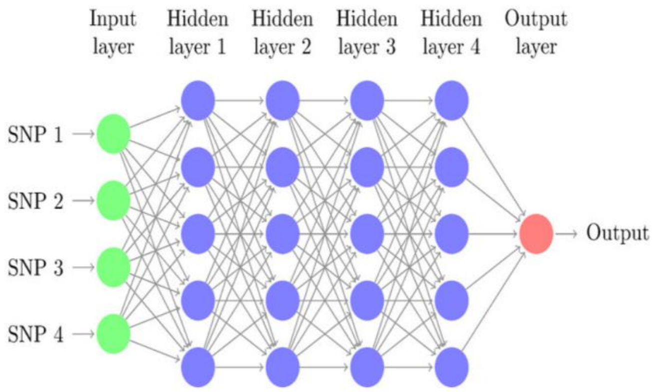

Figure 3. Recurrent Neural Network

2.4. Autoregressive Integrated Moving Average

Auto Regressive Integrated Moving Average (ARIMA) was developed from the Box Jenkins method and aims to predict the future value of a variable based on the previous values of the same variable. ARIMA has the best performance with linear and largely stationary time series data, and generally requires at least 50 historical data events to work effectively [4]. With the ARIMA model, the forecasted value of a variable is deduced through a combination of historic values and can be represented by the equation [5]:

$$yt = θ0 + ϕ1yt−1 + εt − θ1εt−1 \tag{3}$$

where *yt* is the future value, *εt* represents a random error at time *t*, *φt* and *θt* are coefficients, *p* represents an integer known as autoregressive polynomial, and *q* too is an integer known as a moving average polynomial [5].

2.5. Long Short-Term Memory

Long Short-Term Memory (LSTM) network is a type of RNN which has gained popularity in time series / sequential analysis. They consist of an input layer, a hidden layer, and an output layer. The hidden layer of a LSTM network contains memory cells which in turn contains three gates which are responsible for update to their cell state. These three gates are*: an input gate, and output gate, and a forget gate*.

Unlike RNNs, LSTM networks do not suffer from the vanishing gradient problem which is an important factor in stock price prediction as previous information being processed in the neural network will affect the future/subsequent information [12]. Figure Four illustrates the structure of a LSTM network with its 3 gates.

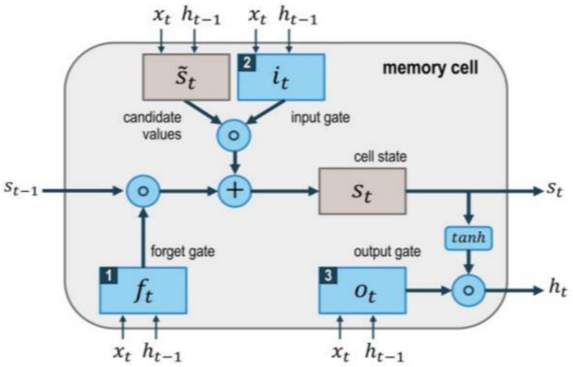

Figure 4. Long Short-Term Memory Network

The output of gate determines the values of the updated cell state. This is represented by the following equations [14]:

$$f_t = \sigma(W_f.[h_{t-1}, x_t] + b_f) \quad (4)$$

$$i_t = \sigma(W_i.[h_{t-1}, x_t] + b_i) \quad (5)$$

$$c_t = \tanh(W_c.[h_{t-1}, x_t] + b_c) \quad (6)$$

$$o_t = sigma(W_o[h_{t-1}, x_t] + b_o) \quad (7)$$

$$h_t = sigma * \tanh(c_t) \quad (8)$$

where *xt* and *ht* are input and output vectors respectively, *ft* is a vector representing the forget gate, *ct* represents the cell state vector, *it* is the vector of the input gate, *ot* is the vector of the output gate, and *W*, *b* represent the parameter matrix and vector.

3. RELATED WORK

In [4], the use of ARIMA for stock price prediction using historic data from the Pakistan company OGDCL was presented. The daily closing prices were used as observations from 23 January 2004 to 19 November 2018, amounting to 3632 data items. The results showed that for short term forecasting, ARIMA was effective, having an error estimate of 0.63 in monetary units but there were certain shortcomings present here.

In a study by [5], ARIMA and SVM are combined in a hybrid model for stock price forecasting. By combining ARIMA and SVM, the study aimed to combat the shortcomings of ARIMA in the inability to effectively analyze non-linear and non-stationary data. In analyzing the model, the stock data of 10 companies was examined and the model results were compared against other models which make use of either only ARIMA or SVM. The results show that the hybrid model outperformed the other discussed models in the paper for predicting future stock market data.

The authors in [12] adopted an ensemble model of LSTM networks in predicting stock market data and compares the results obtained to that of a multilayer LSTM network. The proposed model approaches the ensemble learning process by dividing the test data into a number of smaller data sets using the Bagging algorithm. The smaller datasets are then trained on a number of LSTM networks. The Bagging algorithm is then used to obtain a prediction result from the multiple test results. Data from the Shanghai Composite Index and Shenzhen Composite Index, among others, were used. Results of the study showed that the assemble LSTM network model had an increased accuracy of 11.7%, a precision increase of 8.5%, and a recall rate increase of 10%. The paper suggested that ensemble LSTM network model had a greater predictive accuracy than the multilayer LSTM network.

4. METHODOLOGY & PROPOSED MODEL

Stock market prediction seems a complex problem because there are many factors that have yet to be addressed and it doesn't seem statistical at first. But by proper use of machine learning techniques, one can relate previous data to the current data and train the machine to learn from it and make appropriate assumptions.

From Literature review and related-work we come to know that the The SVM and other model failed to consider factors that are used for testing the prediction model, this is including macro and micro factors. Additionally, implementing SVM methodologies can be very tedious due to the number of algorithms involved. Lastly, SVM models based on a small amount of training and testing sample, it is not suitable for financial time-series forecasting on a large scale.

To address the limitation of a SVM Model in this work, we propose to develop a Deep Learning Model. A Model which shows the comparative analysis between different LSTM Networks such as LSTM, and stacked LSTM networks. It is a deep learning model whose architecture can be defined as an LSTM model comprised of multiple LSTM layers. Despite the potential of the existing SVM Model, it was time consuming and is not suitable for financial time-series forecasting on a large scale which is much needed for stock price prediction.

Here to know why we proposed this Model and to understand the Stacked LSTM we need to first know how the Basic LSTM Model works. Lately, LSTM network, which is suitable for

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 08 Issue: 03 | March - 2024 SJIF Rating: 8.448 ISSN: 2582-3930

learning temporal patterns, is extensively utilized for various tasks of time-series analyses. LSTM is preferred over the conventional RNN as it overcomes the problem of vanishing (or exploding) gradients and as it can effectively learn long term dependencies through memory cells and gates. Thus, many studies on financial time-series modelling are conducted using LSTMs.

LSTM can be trained to learn dependencies ranging over very long-term intervals of time. It beats the vanishing gradients problem faced by a general RNN by substituting the ordinary neuron by a complex architecture called the LSTM unit or block [6]. The working of LSTM can be summarised by the following set of equations:

$$\begin{aligned} z_t &= \tanh(W^z x_t + R^z h_{t-1} + b^z) \\ i_t &= \sigma(W^i x_t + R^i h_{t-1} + b^i) \\ f_t &= \sigma(W^f x_t + R^f h_{t-1} + b^f) \\ o_t &= \sigma(W^o x_t + R^o h_{t-1} + b^o) \\ s_t &= z_t \cdot i_t + s_{t-1} \cdot f_t \\ h_t &= \tanh(s_t) \cdot o_t \end{aligned} \quad \textbf{(9)}$$

(Ref: LSTM Related Work [6].)

where ‘it’ denotes the input gate and ‘ot’ denotes the output gate. The forget gate, memory cell, and hidden state are denoted by ‘ft’, ‘st’, and ‘ht’, respectively. The ‘σ’ and ‘tanh’ functions are defined respectively. The σ and tanh functions are defined in (10) respectively.

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

$$\tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}} \quad (10)$$

(Ref: LSTM Function Equation [6].)

Now, Stacked LSTMs are a stable technique for challenging sequence prediction problems. A Stacked LSTM architecture can be defined as an LSTM model comprised of multiple LSTM layers. An LSTM layer above provides a sequence output rather than a single value output to the LSTM layer below. Specifically, one output per input time step, rather than one output time step for all input time steps. The Input is only required for the first layer.

A stacked LSTM model is trained on the data, and evaluated by measuring the Mean Squared Error (MSE) and Root Mean Squared Error (RMSE) and R-Square Error of the model. 70% of the data will be used as training while the remaining 30% will be used as testing and validation. The one of the following equations are used to evaluate the model:

$$MSE = \frac{1}{n}\sum_k (r_k - y_k)^2 \quad (11)$$

n is the number of observations, *yk* is the output of the kth observation, and *rk* is the kth output of the model [19].

Stacked LSTM networks provide a deeper model for learning and are composed of multiple hidden layers of LSTMs. These multiple hidden layers act as a Deep Recurrent Neural Network (DRNN). Figure five shows the stacked LSTM model adopted by this study, for example with an input, 2 LSTM layers, and an output.

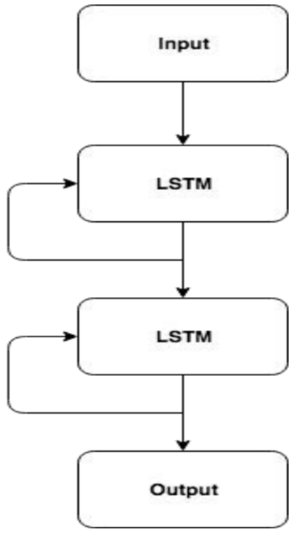

Figure 5. Representation of Stacked LSTM [20]

The main reason that we decided to use a stacked LSTM is to allow for greater model complexity. At every time step an LSTM, besides the recurrent input. If the input is already, the result from an LSTM layer (or a feedforward layer) then the current LSTM can create a more complex feature representation of the current input.

Having a stacked LSTM representation more complex input patterns can be described at every layer. Stacked LSTM hidden layers makes the model deeper, more accurately earning is the description as a deep learning technique [18]. The additional hidden layers are understood to recombine the learned representation from prior layers and create new representations at high levels of abstraction. For example, some complex problems like stock prediction may sometimes require several stacked hidden LSTM layers.

Hence, we have proposed this model which analyses LSTM and Stacked LSTM Deep Neural networks to predict stock prices and visualize the stock data and give use the best working network based on the performance Metrix accordingly.

In developing the model, Python 3 will be used as the programming language of choice with libraries that include pandas, Numpy, and scikit-learn, keras-tensorflow, matplotlib for data analysis, pre-processing and data visualization. The Integrated Development Environment which will be used is Spyder/ Jupyter or Google-Colab depending on the need.

5. PERFORMANCE MEASURES

The different performance measures which are used for evaluation are as follows: the mean squared error (MSE), the Root mean square error (RMSE), mean absolute percentage error (MAPE) and the forecast accuracy [19]. The described variables have the following significance: 'yt' is the original value, 'ft' is the predicted value, 'et=yt – ft' is the predict error and 'n' is the extent of the testing dataset. The details of the forecasting evaluation measures are as follows:

1. **Mean square error (MSE):** is a network performance function. MSE measures network performance based on the average error of the forecasting error [19]. This function is used in prediction and regression analysis to verify the results of the experimental. The Value of MSE should be less or nearer to zero. The formula of the MSE is given by:

$$MSE = \frac{1}{n}\sum_{t=1}^{n} e_t^{\,2}$$

2. **Root mean square error (RMSE**): is one of the most common measures used in regression error metrics. It equals the square root of the MSE. RMSE is an error measure that can see how spread out these residuals are. This error measurement can tell how the concentration of data around the most suitable line [19]. By basic principles, the best value of RMSE should also be close to zero. The equation to calculate the RMSE is given by:

$$RMSE = \sqrt{\frac{1}{n}\sum_{t=1}^{n} e_t^{\,2}}$$

3. **Mean absolute percentage error (MAPE**): is a statistical measure to compute the mean absolute percentage error function for the forecast. The concept of MAPE is separated from the measurement level by data conversion. The value of MAPE has very little deviation. It is unable to determine the direction of the error. The best value of MAPE is close to zero.

$$MAPE = \frac{1}{n}\sum_{t=1}^{n} \left|\frac{e_t}{y_t}\right| \times 100$$

4. **Accuracy:** is typically measured using MAPE. The equation to calculate the accuracy forecast performance is given by:

$$ACCURACY = \left(1 - \frac{1}{n}\sum_{t=1}^{n} \left|\frac{e_t}{y_t}\right| \times 100\right) \times 100$$

5. **R-Squared**: R-squared is a statistical measure that indicates how much of the variation of a dependent variable is explained by an independent variable in a regression model. In investing, R-squared is generally interpreted as the percentage of a funds or security's price movements that can be explained by movements in a benchmark index. R-squared values range from 0 to 1 and are commonly stated as percentages from 0% to 100%.

$$R^2 = 1 - \frac{\text{Unexplained Variation}}{\text{Total Variation}}$$

In our proposed Model, we will be using three matrices from above and they are as follows: Mean Square Error (MSE), Root Mean Square Error (RMSE), and R-Squared Error (R^2).

6.CONCLUSION

This paper study the most attractive topic in the financial market, which is the prediction of the stock price. Due to many external factors ranging from predictable to unpredictable, it is nearly impossible to perform prediction of the future price movements. Fortunately, in recent years, the field of predictive analytic, more specifically, the field of Machine Learning and Deep Learning is exploding with many new models and techniques that can be used in predicting the stock market trend.

Hence, after weighing the benefit and drawbacks of all methodologies, a conclusion can be drawn that Deep Learning has more advantages in comparison to Machine Learning models. Moreover, until this point in time, the development of methodologies to achieve accuracy of 80% or 90% is still a myth, this is not only due to the non-stationary, non-linear data set with chaotic and complex nature but also due to the influence of external factors like the national monetary, fiscal policies and many unpredictable and unforeseeable external factors.

Therefore, it is found from this project that Stacked LSTM Deep Learning Model out-performed Machine Learning Model in all the collected research papers, and it is the most suitable methodologies to apply to the stock market forecasting domain. This can help the investors to gain much financial benefit while retaining a sustainable environment in stock market. In future, we plan to analyse the data from more stock markets of different categories to investigate the performance of our approach. Moreover, with the increase in the advancement of technologies, Deep Learning might become the most favourable method to use when it comes to this domain.

REFERENCES

1. V. Atanasov, C. Pirinsky, and Q. Wang, "The efficient market hypothesis and investor behavior," 2018.

2. S. Rehman, I. U. Chhapra, M. Kashif, and R. Rehan, "Are stock prices a random walk. An Empirical evidence of asian stock markets," ETIKONOMI, vol. 17, no. 2, pp. 237–252, 2018.

3. S. Agrawal, D. Thakkar, D. Soni, K. Bhimani, and C. Patel, "Stock market prediction using machine learning techniques," International Journal of Scientific Research in Computer Science, Engineering and Information Technology, pp. 1099–1103, 04 2019.

4. M. Afeef, A. Ihsan, and H. Zada, "Forecasting stock prices through univariate arima modeling," 2018.

5. P.-F. Pai and C.-S. Lin, "A hybrid arima and support vector machines model in stock price forecasting," Omega, vol. 33, no. 6, pp. 497–505, 2005.

6. D. Karmiani, R. Kazi, A. Nambisan, A. Shah, and V. Kamble, "Comparison of predictive algorithms: Backpropagation, svm, lstm and kalman filter for stock market," in 2019 Amity International Conference on Artificial Intelligence (AICAI). IEEE, 2019, pp. 228–234.

7. E. Ahmadi, M. Jasemi, L. Monplaisir, M. A. Nabavi, A. Mahmoodi, and P. A. Jam, "New efficient hybrid candlestick technical analysis model for stock market timing on the basis of the support vector machine and heuristic algorithms of imperialist competition and genetic," Expert Systems with Applications, vol. 94, pp. 21–31, 2018.

8. S. Sharma and B. Kaushik, "Quantitative analysis of stock market prediction for accurate investment decisions in future," Journal of Artificial Intelligence, vol. 11, pp. 48–54, 2018.

9. S. Jain, M. Kain, and N. Singh, "Prediction for stock marketing using machine learning," Bhagwan Parshuram Institute of Technology, p. 1, 2018.

10. A. A. Heidari, H. Faris, I. Aljarah, and S. Mirjalili, "An efficient hybrid multilayer perceptron neural network with grasshopper optimization," Soft Computing, pp. 1–18, 2018.

11. W. De Mulder, S. Bethard, and M.-F. Moens, "A survey on the application of recurrent neural networks to statistical language modeling," Computer Speech & Language, vol. 30, no. 1, pp. 61–98, 2015.

12. Q. Xie, G. Cheng, X. Xu, and Z. Zhao, "Research based on stock predicting model of neural networks ensemble learning," in MATEC Web of Conferences, vol. 232. EDP Sciences, 2018, p. 02029.

13. T. Fischer and C. Krauss, "Deep learning with long short-term memory networks for financial market predictions," European Journal of Operational Research, vol. 270, no. 2, pp. 654–669, 2018.

14. S. Selvin, R. Vinayakumar, E. Gopalakrishnan, V. K. Menon, and K. Soman, "Stock price prediction using lstm, rnn and cnn-sliding window model," in 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI). IEEE, 2017, pp. 1643–1647.

15. J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, T. Liu, X. Wang, G. Wang, J. Cai et al., "Recent advances in convolutional neural networks," Pattern Recognition, vol. 77, pp. 354–377, 2018.

16. T. Kim and H. Y. Kim, "Forecasting stock prices with a feature fusion lstm-cnn model using different representations of the same data," PloS one, vol. 14, no. 2, p. e0212320, 2019.

17. R. M. I. Kusuma, T.-T. Ho, W.-C. Kao, Y.-Y. Ou, and K.-L. Hua, "Using deep learning neural networks and candlestick chart representation to predict stock market," arXiv preprint arXiv:1903.12258, 2019.

18. M. Peng, C. Wang, T. Chen, and G. Liu, "Nirfacenet: A convolutional neural network for near-infrared face identification," Information, vol. 7, no. 4, p. 61, 2016.

19. E. Guresen, G. Kayakutlu, and T. U. Daim, "Using artificial neural network models in stock market index prediction," Expert Systems with Applications, vol. 38, no. 8, pp. 10 389–10 397, 2011.

20. U. Singh, S. Chauhan, A. Krishnamachari, and L. Vig, "Ensemble of deep long short term memory networks for labelling origin of replication sequences," in 2015 IEEE International Conference on Data Science and Advanced Analytics (DSAA). IEEE, 2015, pp. 1–7.

--------------------------------***-----------------------------