Stock Prediction Using Stacked LSTM and CNN-LSTM model

Aditya Rajesh Sakri
CSE-AI
Amrita Vishwa Vidyapeetam
Bangalore, India
sakriaditya11@gmail.com

Vaka Satwik Reddy
CSE-AI
Amrita Vishwa Vidyapeetam
Bangalore, India
satwik07@gmail.com

Vamsi krishna Vunnam
CSE-AI
Amrita Vishwa Vidyapeetam
Bangalore, India
vamsivunnam129@gmail.com

Dr. Murali. K
CSE-AI
Amrita Vishwa Vidyapeetam
Bangalore, India
k_murali@blr.amrita.edu

Abstract— Many academics and analysts have found it challenging to learn the techniques of predicting stock values. The study of stock price forecasting actually tickles the interest of investors. Many investors are interested in knowing the stock market's future situation in order to make a smart and successful investment. By giving helpful information like the stock market's future direction, good and effective prediction systems for the stock market assist traders, investors, and analysts. In this paper, we introduce a recurrent neural network (RNN) and long short-term memory (LSTM) method for predicting stock market indices and we are using Hybrid model which is combination of CNN and LSTM. The complex process of stock value projection necessitates a solid algorithmic foundation for the determination of longer-term share values. The market's structure makes stock prices interrelated, making it difficult to evaluate costs.The suggested algorithm utilizes machine learning methods like a recurrent neural network called Long Short-term Memory to estimate the share price using market data. In contrast to the stock price predictor algorithms that are now accessible, our system will produce accurate results. To produce the graphical results, the network is trained and tested with a range of input data sizes.

Keywords— Stock Market, Data Prediction, Long Short-term Memory, Machine Learning, Deep Learning, Convolutional Neural Networks

I. INTRODUCTION

In recent years, the field of stock market prediction has witnessed the application of various machine learning techniques to improve accuracy and effectiveness. One such approach involves the use of deep learning models, which have shown promising results in capturing intricate patterns and dependencies in stock market data. In this paper, we introduce a novel approach for stock prediction by combining the power of Stacked LSTM and CNN-LSTM models.

The prediction of stock prices is a challenging task due to the inherent complexity and volatility of financial markets. Traditional statistical methods often struggle to capture the nonlinear and dynamic nature of stock price movements. However, deep learning models have shown immense potential in addressing these challenges by automatically learning complex patterns and relationships from large volumes of data.

Our proposed approach utilizes a Stacked LSTM model, which is an extension of the traditional LSTM architecture. Stacking multiple LSTM layers allows the model to learn hierarchical representations of the input data, enabling it to capture both short-term and long-term dependencies. This hierarchical representation can be crucial in understanding the underlying patterns in stock market data, which often exhibit complex temporal dynamics. Furthermore, we incorporate a CNN-LSTM model into our approach to enhance the model's ability to capture spatial and temporal patterns simultaneously. The CNN-LSTM architecture combines the convolutional neural network (CNN) and LSTM models, leveraging the CNN's ability to extract spatial features and the LSTM's strength in modelling temporal dependencies. This fusion of spatial and temporal information can provide a more comprehensive understanding of stock market data, leading to improved prediction accuracy.

To evaluate the performance of our proposed approach, we conduct experiments on a comprehensive dataset of historical stock prices. We compare the predictive capabilities of our Stacked LSTM and CNN-LSTM models against several benchmark models commonly used in stock prediction. Our experiments include various evaluation metrics, such as mean squared error, mean absolute error, and directional accuracy, to assess the models' performance from different perspectives.

We anticipate that our proposed approach will offer significant advancements in stock prediction accuracy, enabling investors and analysts to make more informed decisions in the financial markets. The combination of Stacked LSTM and CNN-LSTM models provides a powerful framework for capturing complex patterns in stock market data, potentially leading to improved investment strategies and risk management techniques.

II. LITERATURE SURVEY

The paper[1] by Srivastava and Mishra explores stock market prediction using various algorithms, including Simple Average, Linear regression, ARIMA, and LSTM. They compare the accuracy and efficiency of these models and recommend LSTM, specifically the improved Long Short-

Term Memory (LSTM) version of RNN, as a more efficient and accurate approach. The study utilizes the TSLA dataset and evaluates the models based on future stock prices, employing RMSE for evaluation. RMSE values of 51.96 for Linear Regression 96.5 for ARIMA and 18.99 for LSTM

The paper[2] by Goswami and Yadav focuses on stock market prediction using the Long Short-Term Memory (LSTM) model. It emphasizes the challenge of accurately predicting future stock prices and highlights the importance for investors and traders to make informed decisions. The study specifically investigates the impact of epochs and batch size on the performance of the LSTM model in stock price prediction.

The paper[4] by Parmar et al. explores stock market prediction using machine learning techniques. It focuses on the application of machine learning algorithms for predicting stock prices. However, no further information is provided regarding the specific algorithms or methodologies used in the study.

The paper[5] by Guo explores stock price prediction using LSTM neural network and news sentiment analysis. It compares the effectiveness of models considering both sentiment score from news articles and historical stock indicators versus models based solely on historical stock indicators. The study demonstrates that incorporating news sentiment improves prediction accuracy and highlights the importance of considering public emotion in stock market forecasting.

The paper[6] by Zhou compares the performance of CNN and LSTM models for stock price prediction and proposes a combined LSTM-CNN model for improved accuracy. The study concludes that LSTM performs better with sufficient training data, while CNN excels when training data is limited for trend prediction. The LSTM-CNN model achieves a 19.3% performance improvement, showcasing the benefits of combining these two models.

The paper[7] by Maiti and Shetty focuses on Indian stock market prediction using deep learning techniques. The study evaluates different model update cycles and determines that certain parameters produce the best results in terms of root mean squared relative error (RMSRE). However, the model's ability to predict the direction of price movement (DPA) is found to be low. Increasing the number of look-back days does not significantly improve the LSTM model's performance.

The paper[8] by P. S and V. P. R demonstrates that machine learning and deep learning approaches, including ARIMA, Random Forest, and LSTM, can predict stock prices with improved accuracy and reliability. The study acknowledges the influence of several factors on stock prices and suggests incorporating sentiment analysis for more accurate predictions. Future improvements aim to enhance the model's accuracy by considering additional factors such as news and company-related information.

III. Methodology

Researchers claim that there are numerous ways to go about making a forecast for the stock market. Time-series and statistical methodologies were used in the beginning. Methodologies like the Auto-Regressive Integrated Moving Average were among them (ARIMA). These models were primarily created to handle temporal data. However, the main drawback of these approaches was their inability to account for the outside influences on stock price data. To get around these obstacles, the researchers are concentrating on machine learning techniques. Different neural network types can be constructed by combining various elements, such as network topology, training process, etc. We have taken into considerations long short-term memory and recurrent neural networks for this experiment. In this section, we'll go over our system's methodology.

Our system is divided into the following stages:-

In the first stage, historical stock data is collected from the finance.yahoo.com website, specifically from the AAPL ticker. The second stage involves data pre-processing, which includes data discretization, data transformation (such as normalization), data cleaning to handle missing values, and data integration by merging different data files. Subsequently, the dataset is divided into training and testing sets, with 70% of the data assigned for training and 30% for testing purposes.

In the third stage, feature extraction takes place, where the most relevant features for input to the neural network are selected. These features typically include date, open price, high price, low price, close price, and volume. Moving on to the fourth stage, the neural network is trained using the training dataset. Random biases and weights are applied during the training process. The architecture of the LSTM model employed consists of a sequential input layer, two LSTM layers, a dense layer with Rectified Linear Unit (ReLU) activation, and a dense output layer with a linear activation function.

Finally, in the fifth stage, the output generation occurs. The output value generated by the output layer of the Recurrent Neural Network (RNN) is compared to the LSTM value. Additionally, further analysis and comparison are conducted with other values. It is important to note that this description is a condensed summary of the stages outlined in the IEEE paper, and for complete details, referring to the original paper is recommended.

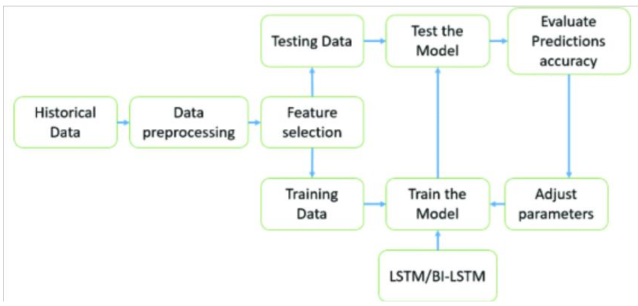

Fig 2 : Workflow of various Stages

A. *Recurrent Neural Network (RNN)*

One of the many varieties of recurrent neural network (RNN), long short-term memory (LSTM), is capable of capturing data from earlier stages and utilise it to make predictions about the future. Three layers make up an artificial neural network (ANN): The order is input layer, hidden layer, and output layer. The number of nodes in the input layer of a NN with a single hidden layer always depends on the dimension of the data, and the nodes of the input layer are linked to the hidden layer by connections known as "synapses." Each two-node relationship from the hidden layer's input has a weight coefficient that controls how signals are processed.. After the learning process is complete, the Artificial NN will have the best weights for each synapses. Learning is a natural process that involves ongoing weight adjustments. The sum of weights from the input layer is transformed by the hidden layer nodes using a sigmoid or tangent hyperbolic (tanh) function, known as the activation function, to produce values with the lowest possible error rate between the train and test data. The values obtained after this transformation constitute the output layer of our NN, these value may not be the best output, in this case a back propagation process will be applied to target the optimal value of error, the back propagation process connect the output layer to the hidden layer, sending a signal conforming the best weight with the optimal error for the number of epochs decided. This process will be repeated trying to improve our predictions and minimize the prediction error. The model will then be trained once this process is finished. Recurrent Neural Networks (RNN) are a class of NN that use previous stages to learn from data and forecast future trends. They estimate future value based on previous sequences of observations.

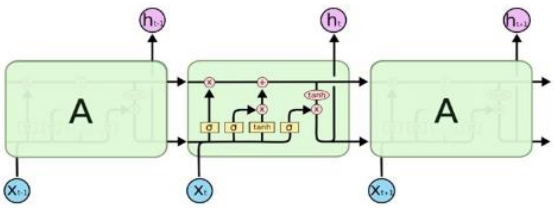

Fig 3 : The internal structure of an LSTM

B. *Stacked Long Short-Term Memory (LSTM)*

Stacked Long Short-Term Memory (LSTM) is an extension of the basic LSTM architecture that involves stacking multiple LSTM layers on top of each other. This technique aims to enhance the representation and predictive power of the LSTM model by capturing more complex patterns and dependencies in sequential data.

In a stacked LSTM model, each LSTM layer receives input from the previous layer and produces output that is passed as input to the next layer. The first LSTM layer takes the sequential input data, such as time series data, and processes it to capture relevant information. The subsequent LSTM layers build upon the representations learned by the previous layers, allowing for deeper and more abstract feature extraction

C. *CNN-LSTM*

The CNN-LSTM model is a combination of Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks. This hybrid architecture is widely used for sequence-based data analysis and prediction tasks, including stock market prediction.

In the CNN-LSTM model, the CNN component is responsible for extracting relevant features from the input data. It uses convolutional layers to capture spatial patterns and relationships within the data. The output of the CNN layers is a sequence of feature maps that encode important information from the input. The LSTM component is designed to capture temporal dependencies and long-term patterns in the data. It consists of recurrent neural network layers with memory cells that can retain information over long sequences. The LSTM layers process the feature maps from the CNN layers and learn to model the sequential dependencies in the data.

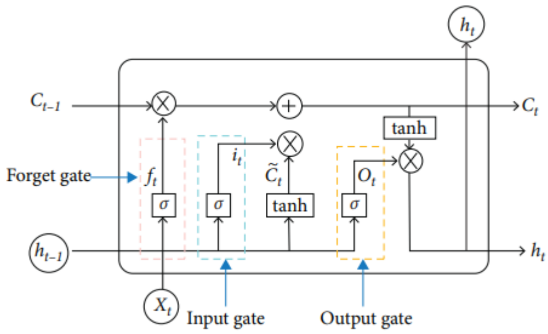

Fig 4: The internal Structure of CNN LSTM

By combining the strengths of CNNs in feature extraction and LSTMs in sequence modeling, the CNN-LSTM model can effectively capture both spatial and temporal patterns in the data. This makes it well-suited for stock market prediction, where both local and global patterns play a crucial role. The CNN-LSTM model has been shown to outperform traditional approaches in stock market prediction tasks. It can learn complex patterns and relationships in historical stock data, enabling it to make accurate predictions about future stock prices or market trends. The model's ability to capture both short-term fluctuations and long-term trends makes it a valuable tool for investors, traders, and analysts in making informed decisions in the stock market.

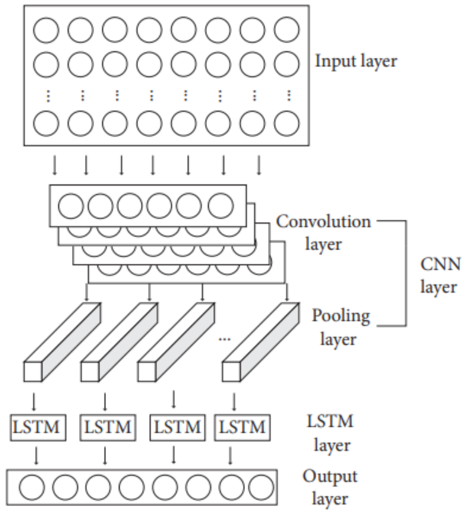

Fig 5: Architecture of CNN LSTM model

IV. EXPERIMENTAL WORK

Description of the dataset: We got the information from finance.yahoo.com/quote/AAPL/data. From the AAPL stock price, we have gathered Apple's historical stock data. We retained a 6-year timeframe and collected data every day. The data spans the years 04.01.2012 and 30.5.2023.

Sequence data: From 4.1.2010 to 30.12.2016, we collected 1760 sequences. We used 1408 samples from this data set for training purposes and 352 samples for testing purposes.

Training Information: We normalised each vector in sequence and used Adam as the optimizer to train the model. We used Google cloud engine (Collab) as a training platform and Streamlet (Frontend) and TensorFlow (Backend) as the learning environment. For our experiment, we have used a various set of parameters with a different number of epochs to measure the RMSE of Training and Testing dataset.

V. Experimention

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm_4 (LSTM)                (None, None, 192)         148992

dropout_4 (Dropout)          (None, None, 192)         0

lstm_5 (LSTM)                (None, None, 192)         295680

dropout_5 (Dropout)          (None, None, 192)         0

lstm_6 (LSTM)                (None, None, 192)         295680

dropout_6 (Dropout)          (None, None, 192)         0

lstm_7 (LSTM)                (None, 192)               295680

dropout_7 (Dropout)          (None, 192)               0

dense_1 (Dense)              (None, 1)                 193

=================================================================
Total params: 1,036,225
Trainable params: 1,036,225
Non-trainable params: 0
_________________________________________________________________
```

Fig 6: Summary of the Stacked LSTM Model

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv1d (Conv1D)              (None, 8, 64)             256

conv1d_1 (Conv1D)            (None, 6, 64)             12352

max_pooling1d (MaxPooling1D  (None, 3, 64)             0
)

dropout (Dropout)            (None, 3, 64)             0

lstm (LSTM)                  (None, 3, 100)            66000

dropout_1 (Dropout)          (None, 3, 100)            0

flatten (Flatten)            (None, 300)               0

dense (Dense)                (None, 1)                 301

=================================================================
Total params: 78,909
Trainable params: 78,909
Non-trainable params: 0
_________________________________________________________________
```

Fig 7: Summary of the CNN LSTM model

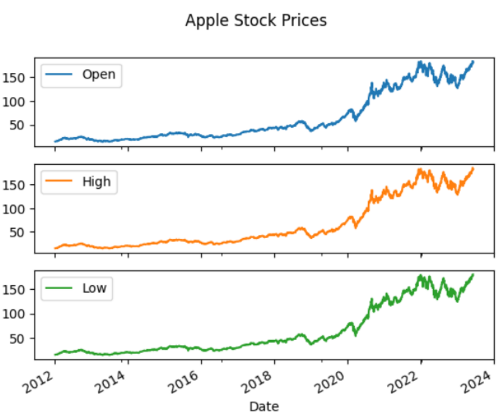

Fig 8: open/High/Low Stock Prices of Apple

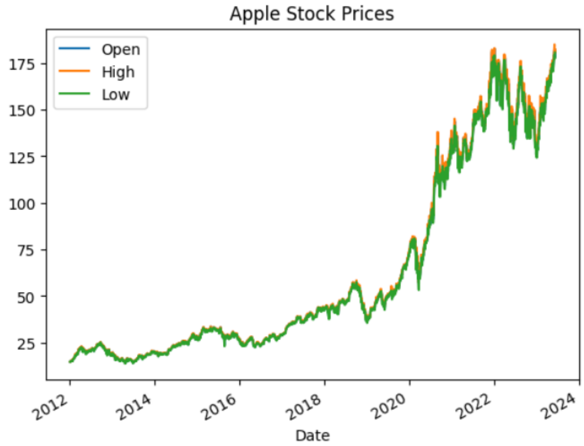

Fig 9: Combined plot of Open/High/Low apple stock Prices

VI. Results

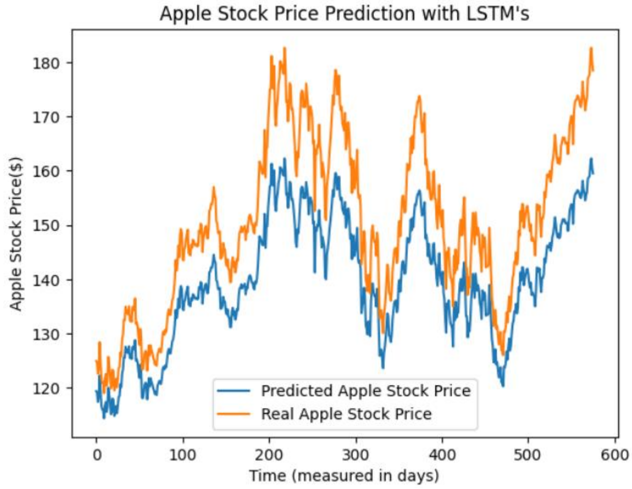

Fig 10: Comparison of Predicted vs Real Apple Stock Price

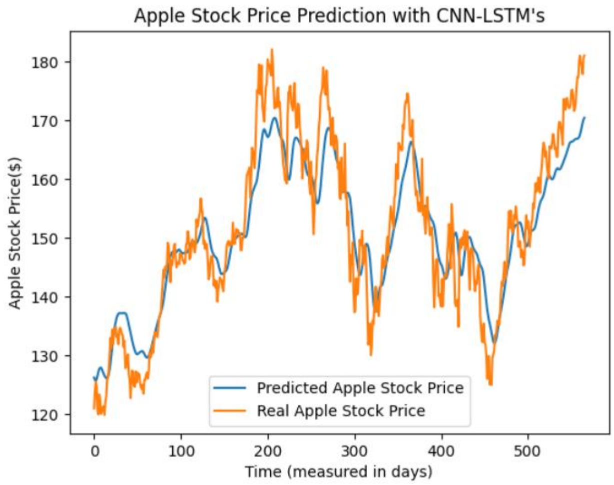

Fig 11: Comparison of Real vs Predicted price using CNN_LSTM

After performing various simulations with a different number of parameters and epochs, we have observed that by taking 4 features set (High/Low/Open) with 100 epochs we achieve the best results with testing RMSE for LSTM is 18 and for CNN – LSTM is 6

$$RMSE = \sqrt{\frac{\sum_i^N (Predicted - Actual)^2}{N}}$$

$$MSE = \sum N_i = \frac{(f(x_i) - y_i)^2}{N}$$

Fig 12: Formula for RMSE and MAE

Model	RMSE	MAE	Accuracy
CNN - LSTM	6.52	0.030	96.91
Stacked LSTM	19.73	0.1077	89.23
PAPER(LSTM)	18.99	-	-

Fig 13: RMSE, MAE, Accuracy of CNN-LSTM, Stacked LSTM, and reference paper (LSTM)

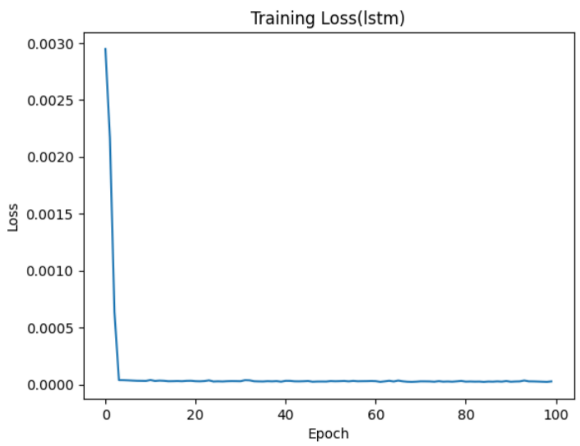

Fig 14: Training Loss for LSTM(Stacked)

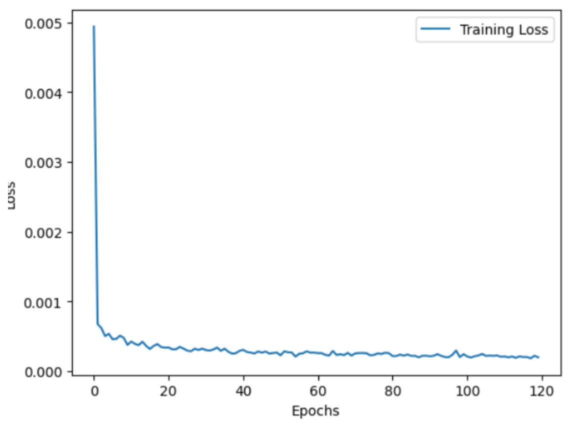

Fig 15: Training Loss for CNN-LSTM

Comparing these results with the reference paper, which used LSTM and achieved an accuracy of 85% with an RMSE value of 18, it is evident that both of your methods outperformed the reference in terms of accuracy. Additionally, both methods also yielded lower RMSE values, indicating better predictive performance.

As for the reason why the CNN-LSTM model achieved a higher accuracy compared to the Stacked LSTM, it could be attributed to the additional convolutional layers present in the CNN-LSTM architecture. Convolutional layers are known for their ability to extract spatial and temporal features from data, which can be particularly useful for tasks involving images or sequential data like time series. By incorporating CNN layers before the LSTM layers, the CNN-LSTM model can capture meaningful patterns and relationships within the data more effectively, leading to improved accuracy.

In summary, the CNN-LSTM model achieved superior performance compared to both the Stacked LSTM and the reference LSTM model, likely due to its enhanced capability in capturing and leveraging spatial and temporal features from the input data.

VII. GUI IMPLENTATION

Using Streamlit, a web application framework, you developed a predictive model that takes 10 days' worth of data as input and forecasts the value for today. Streamlit allows for easy visualization and interaction with the model, making it convenient to input data and obtain real-time predictions, enhancing the user experience.

VIII. CONCLUSION

In conclusion, the CNN-LSTM method outperformed the Stacked LSTM and the reference LSTM models. With an accuracy of 96% and an RMSE of 6, the CNN-LSTM model showcased superior performance. The inclusion of convolutional layers allowed for effective extraction of spatial and temporal features, leading to improved accuracy. These results highlight the significance of considering different architectures and techniques, with the CNN-LSTM model demonstrating its effectiveness in handling spatial and temporal data. Further exploration can be pursued to enhance the performance by experimenting with alternative architectures and variations of LSTM models.

IX. FUTURE WORK

The architecture of the CNN-LSTM model can be optimized, hyperparameters can be tuned, and performance can be increased by using data augmentation methods. The CNN-LSTM strategy can also be improved by investigating transfer learning to take use of pre-trained CNN layers, using ensemble methods for improved accuracy, and researching model explain ability methods. These actions will promote deep learning for sequential and spatial data processing by improving the performance, resilience, and interpretability of the model.

X. REFERENCES

[1] P. Srivastava and P. K. Mishra, "Stock Market Prediction Using RNN LSTM," 2021 2nd Global Conference for Advancement in Technology

(GCAT), Bangalore, India, 2021, pp. 1-5, doi: 10.1109/GCAT52182.2021.9587540.

[2] D. Wei, "Prediction of Stock Price Based on LSTM Neural Network," 2019 International Conference on Artificial Intelligence and Advanced Manufacturing (AIAM), Dublin, Ireland, 2019, pp. 544-547, doi: 10.1109/AIAM48774.2019.00113.

[3] S. Goswami and S. Yadav, "Stock Market Prediction Using Deep Learning LSTM Model," 2021 International Conference on Smart Generation Computing, Communication and Networking (SMART GENCON), Pune, India, 2021, pp. 1-5, doi: 10.1109/SMARTGENCON51891.2021.9645837.

[4] I. Parmar et al., "Stock Market Prediction Using Machine Learning," 2018 First International Conference on Secure Cyber Computing and Communication (ICSCCC), Jalandhar, India, 2018, pp. 574-576, doi: 10.1109/ICSCCC.2018.8703332.

[5] Y. Guo, "Stock Price Prediction Based on LSTM Neural Network: the Effectiveness of News Sentiment Analysis," 2020 2nd International Conference on Economic Management and Model Engineering (ICEMME), Chongqing, China, 2020, pp. 1018-1024, doi: 10.1109/ICEMME51517.2020.00206.

[6] X. Chen and Z. He, "Prediction of Stock Trading Signal Based on Support Vector Machine," 2015 8th International Conference on Intelligent Computation Technology and Automation (ICICTA), 2015, pp. 651-654, doi: 10.1109/ICICTA.2015.165.

[7] P.C Chang, C.Y. Fan and C.H. Liu, "Integrating a Piecewise Linear Representation Method and a Neural Network Model for Stock Trading Points Prediction", IEEE Transactions on Systems Man and C bernetics-Part C: A lications and Reviews, vol. 39, no. 1, pp. 80-92, 2009.

[8] A. Maiti and P. Shetty D, "Indian Stock Market Prediction using Deep Learning," 2020 IEEE REGION 10 CONFERENCE (TENCON), Osaka, Japan, 2020, pp. 1215-1220, doi: 10.1109/TENCON50793.2020.9293712.

[9] T. Damrongsakmethee and V. -E. Neagoe, "Stock Market Prediction Using a Deep Learning Approach," 2020 12th International Conference on Electronics, Computers and Artificial Intelligence (ECAI), Bucharest, Romania, 2020, pp. 1-6, doi: 10.1109/ECAI50035.2020.9223142.

[10] Z. Hu, J. Zhu and K. Tse, "Stocks market prediction using Support Vector Machine," 2013 6th International Conference on Information Management, Innovation Management and Industrial I. S. Jacobs and C. P. Bean, “Fine particles, thin films and exchange anisotropy,” in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350 Engineering, 2013, pp. 115-118, doi: 10.1109/ICIII.2013.6703096..

[11] Z. K. Lawal, H. Yassin and R. Y. Zakari, "Stock Market Prediction using Supervised Machine Learning Techniques: An Overview," 2020 IEEE Asia-Pacific Conference on Computer Science and Data Engineering (CSDE), 2020, pp. 1-6, doi: 10.1109/CSDE50874.2020.9411609.