Proceedings of the 8th International Conference on Communication and Electronics Systems (ICCES 2023)
IEEE Xplore Part Number: CFP23AWO-ART; ISBN: 979-8-3503-9663-8

2023 8th International Conference on Communication and Electronics Systems (ICCES) | 979-8-3503-9663-8/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICCES57224.2023.10192634

Innovative Deep Learning Model-based Stock Price Prediction using a Hybrid Approach of CNN and Gradient Recurrent Unit

1. Dr B. Srinivasa Rao, Professor, Department of Computer Science and Engineering, Gokaraju Rangaraju Institute of Engineering and Technology, Bachupally, Hyderabad, Telangana, India. bsrgriet2015@gmail.com

2. Dr Rajib Bhattacharya, Associate Professor, Department of PGDM, Globsyn Business School, Chandipur, Bishnupur JL No. 101, South 24 Paganas, West Bengal, India. rajib.learner@gmail.com

3. Manoj Kumar Tiwari, Assistant Professor, Department of Computer Application, JECRC University, Jaipur, Rajasthan, India. manojkumartiwari@rediffmail.com

4. K. Aruna Kumari, Assistant Professor, Mathematics. GSS, GITAM (Deemed to be) University, Visakhapatnam, India. akatraga@gitam.edu

5 Dr Mahavir A. Devmane, Professor, Department of Computer Engineering, VPPCOE & VA, Mumbai, Maharashtra, India. dmahavir@gmail.com

6. Dr. Kamlesh Singh, Professor, Department of Computer Science and Engineering, Graphic Era Hill University, Dehradun Campus, Dehradun, India. mailkamleshsingh@gmail.com

***Abstract* – The fluctuation in stock prices from industry to industry are a major source of concern in the market. As the market attracts more participants and stock prices play a larger role in more transactions, the ability to accurately predict stock price movements becomes more valuable. When making an investment, many people first look at the share price and then try to anticipate whether or not that price will go up or down in the future. The traditional problem of forecasting the stock market using Machine Learning tools and methodologies has been thoroughly studied. Time dependence, volatility, and similar complicated dependencies are interesting aspects that make this modeling non-trivial. To overcome this the proposed method in this work is a deep learning-based hybrid strategy for predicting stock prices. Preprocessing is performed after input is delivered to increase precision. Selecting MPSOA features is the next step. In the end, it's put to use in MC-GRU model training. The proposed method achieves better results than both the CNN and GRU models.**

***Keywords*—Convolutional Neural Network (CNN), Mass Particle Swarm Optimization Algorithm (MPSOA), Gradient Recurrent Unit (GRU).**

I. INTRODUCTION

The stock exchange is a marketplace for buying and selling stocks. Companies have been using this method of funding for almost 400 years. [1]. As shares are issued, a lot of new money enters the stock market. This leads to growth in the commodities economy, a higher concentration of capital, and an enhanced organic composition of enterprise capital. Hence, the stock market is a gauge of the health of the economy and the finances. Researchers are constantly seeking for ways to enhance their methods, and prediction is expected to remain a popular subject of study. Investment savvy, strategic planning, and the ability to effectively put plans into action are all essential. Due to the complicated nature of the stock market, predicting stock prices is one of the most challenging issues in financial forecasting. Those who invest in the stock market often wish they had access to a foolproof way of forecasting that would guarantee steady earnings with minimal risk. Scientists have used this as motivation to develop better and more accurate forecasting models. In recent years, the volume and complexity of financial transactions have risen alongside the rapid expansion of the global economy. Understanding the structure of financial operations and foreseeing their evolution and changes is a primary goal of research in academic and financial circles. Profit-maximizing investors and for-profit businesses can benefit from a better understanding of the evolution and changes in the financial market thanks to an approximate prediction of financial data using one or a set of approaches. The stock market looks to no longer be a good indication of the overall health of the economy. Partially dissolving the tight cause and time relationship between the stock exchange and the actual economic movements has made the already challenging process of predicting stock prices even more so. Stock market reactions to political and other relevant changes have become faster as data processing technology and communication systems have developed. The stock market provides a glimpse of investor confidence in the economy and the prospects for individual enterprises. Stock price fluctuations can be related to many different things, such as the state of the economy, market expectations, and investor faith in the company's leadership and operations. The public can now get their hands on more information, and it can get it faster, thanks to technological advancements. This has made stock analysis more challenging as more and more information needs to be digested in less time. Investors anticipate that deep learning and other developments in big data will improve their ability to interpret stock market data. Despite the attractive rate of return offered by equities investments, there is a considerable degree of risk involved due to market volatility. So, investors should get an understanding of the traits of individual stocks and the circumstances upon which their prices are dependent in order to increase their chances of making a profit. Yet, in order to make good investment decisions at the right time, investors require accurate and sufficient information, such as investor sentiment and interest rates. The ability of artificial neural network (ANN) technology to learn and identify link among nonlinear variables has contributed to its commercial success. It also enables more in-depth study of huge datasets, especially those that tend to vary rapidly over short time periods. Researchers have focused a lot of time and energy on improving both the accuracy of share value predictions and the efficiency of computing share values.

II. LITERATURE SURVEY

Stock forecasting using ANN and ARIMA models was compared by[2], who showed that the former yielded better results. Also, [3] compared ANNs and ARIMA for time series prediction and discovered the former to be superior at predicting the direction of stock movement due to its superiority in revealing latent patterns in the data. [4]compared the accuracy of forecasts made using an ANN and an ARIMA model for daily maximum ozone concentration. Empirical evidence also supported the ANN model's superiority over the ARIMA model. [5]examined the Indonesia stock market and found that ANN outperformed the ARIMA model in terms of accuracy. Predictions of stock prices using ANNs have been shown to be effective [6]. Thus, ANN is a preferred hybrid methodology or method for predicting shifts in time series. There is a plethora of literature on the subject of mathematical prediction because it has been such a popular topic for so long. To better predict stock prices, researchers at pioneered the use of neural network models[7]. The training forecasts made by his algorithm using IBM's daily common stock were extremely optimistic. Many studies then looked at whether or not neural networks could reliably predict the stock market[8] proposed a fusion model that combines the ANN, the HMM and the GA to anticipate market behavior in the financial industry. The performance of the fusion tool was evaluated against that of the basic model[9], which only used a single HMM, and was shown to be superior. The fusion model was shown to have similar results to the ARIMA model. The application of ANNs to financial market forecasting was the subject of a literature review by[10]. Traditional time series research failed to predict the stock market, but an artificial neural network showed promise. Neural Networks allow for the efficient mining of massive data stores for relevant insights. Artificial neural networks have already demonstrated potential in the literature for forecasting global stock markets. There is a lot of room for growth in the field of employing an ANN to predict stock market indices accurately, as they proposed. The use of data mining strategies to develop a market capital forecast system for brokerages is discussed, by [11]. Their work demonstrates how MATLAB's GUIDE graphical user interface may be used in tandem with neural networks to generate reliable forecasts. Current ideas for stock price movement and stock price projections can be broken down into three distinct camps, each with its own unique set of components and methods for calculating stock prices. The first applies a basic regression model to cross-sectional data [12].Due to the very non-linear nature of stock price movement, these models cannot reliably predict future outcomes. The second set of theories relies on time series models and methodologies to create projections about stock values[13]. The final set of suggestions [14] addresses the question of how natural language processing, deep learning, and machine learning natural language processing can be used to reliably predict future stock returns. Predictions of stock prices have often been made using quite simple mathematical models. Simple linear models, such as the basic auto regression model and the simple moving average model, were initially used by financial researchers to evaluate stock market data. The unit root test is used to determine if there is a stationary component to a time series. Non-stationary time series are used to "map" various procedures. However, the limitations of the linear model become more obvious as the prediction horizon lengthens. This is due to the abundance of noise and unknowns in stock data. Academics from all over the world have tried to predict stock values using a wide variety of methods, including Kalman filter models, vector auto-regressive models, error correction models and Bayesian vector auto-regressive models. Using a variety of LSTM designs, [15]found that LSTMs can learn patterns that are beneficial for predicting the stock market, and he attained reasonable RMSEs. This proposed approach to enlighten us to the fact that this is a time-series problem, and the merits of a sliding-window approach to fixing it. MLPs are superior to LSTMs in predicting stock prices, according to research by [16]The main focus of their study was on predicting prices between business days. This discovery gave us food for thought, and we almost tried using it to solve our problem of making short-term price predictions. In his research[17] constructed a variety of ensemble models to foresee the behavior of stock prices the following day .[18] To quantitatively examine the effects on stock prices, a fuzzy neural network based on a genetic algorithm can be used. The choice of variables has a direct impact on the productivity of a network and the bottom line of an organization. They used their methodology on the stock market in[19] use an FFNN trained with a genetic algorithm to forecast interest rates on three-month US Treasury Bills (GA). Using a NN, they found that these rates could be predicted with a high degree of accuracy. Current and historical pricing, along with profitability metrics, were used to calibrate the model. They [20]utilised online stock news articles and NASDAQ hourly stock prices for 30 distinct stocks. For a period of six months, tweets were gathered that made reference to one of the thirty stocks under investigation[21]collected information from the Hong Kong Stock Exchange over the course of five years. They gathered stories on the economy and the stock market at the same time so they could see if there was any connection between the two. The information covered the day's open, high, low, and close stock prices for each firm. The development of sophisticated prediction algorithms and the quantitative examination of high-frequency data are both made possible by recent developments in computing technology, as shown by. Conventional methods are inadequate for processing data at these higher frequencies. [22]provide recommendations for handling outliers and temporal anomalies in high-frequency data. High-frequency trading, also known as HFT, is a trading approach that makes rapid trades by analyzing and acting on processed high-frequency data (HFT). There are a number of authors that have investigated the effects of this type of transaction in modern financial markets, including[23]. If [24]argue that CNN models are comparable to the LSTM model and can achieve success in time series analysis. According to[25], the CNN model outperforms the ANN and SVM models when it comes to studying and predicting stock market time series. Beyond a single LSTM model, other studies have investigated the synergistic potential of merging LSTM and CNN models. The proposed approach uses CNN-GRU to train the model.

III. PROPOSED SYSTEM

The importance of stock price prediction to the fields of finance and economics has kept researchers engaged in the topic for decades. The suggested method employs preprocessing, MPSOA feature selection, and MC-GRU for model training.

A. Preprocessing:

Most heuristic methods benefit better from altered data, especially when applied to forecasting issues. A preprocessing technique should have the capacity to restore the original scale of the data once it has been transformed (called post-processing).

In this proposed approach employs Min-Max normalization, a standard method in this area. The accuracy of subsequent numerical computations is enhanced by applying the Min-Max normalization to a data set [26]. The following transformation function can be used to generate z^*_{new}, the normalized version of z_{old}, where z_{old}, z_{max}, and z_{min} are the original, maximum, and minimum values of the raw data, and z^*_{max} , z^*_{min} are the maximum and minimum of the normalized data, respectively.

$$z^*_{new} - \left(\frac{z_{old} - z_{min}}{z_{max} - z_{min}}\right)(z^*_{max} - z^*_{min}) + z^*_{min} \tag{1}$$

B. MPSOA Feature Selection

The PSO is an example of an optimization technique that employs a main population that is produced at random. The behavior of groups of animals, such as flocks of birds or schools of fish, served as inspiration for this algorithm. Each member of this set is defined by a unique combination of search space velocity and position vectors.[27]. Each new iteration begins with the particles' positions in the search space being reset according to the velocity and position vector. The new particle's position is updated at each time step using the previous particle's best position, the current velocity vector, and the position of the best particle in the group. Originally intended for use with continuous parameters, this method has found sufficient utility to warrant transfer to the discrete setting as well. Enhanced binary particle swarm optimization is introduced with (BPSO).

Let's pretend our search space has d dimensions. The position vector Z_k of the kth particle in this d-dimensional space looks like this:

$$Z_k = \left(z_{k_1}, z_{k_2}, \ldots., z_{k_d}\right) \tag{2}$$

Likewise, the velocity vector of the kth particle is calculated as follows, where $w_k$the velocity vector is.

$$w_k = (w_{k_1}, w_{k_2}, \ldots., w_{k_e} \tag{3}$$

$Q_{K.best}$ Denotes the optimal location for the kth particle.

$$Q_{K.best} = (Q_{k_1}, Q_{k_2}, \ldots. Q_{k_e}) \tag{4}$$

$Q_{h.best}$ is a particle's optimal placement, as determined by the largest particle within the entire particle, is defined as follows:

$$Q_{h.best} = \left(Q_{h_1}, Q_{h_2}, \ldots.. Q_{h_e}\right) \tag{5}$$

To keep track of where each particle is at all times, it utilizes:

$$w_k(s) = v * w_k(s-1) + d_1 * rand_2\left(Q_{K.best} - z_k(s-1)\right) + d_2 * rand_2\left(Q_{h.best} - z_k(s-1)\right) \tag{6}$$

At each repetition, the velocity vector $w_k(s)$ is affected by the velocity vector ($w_k(s-1)$) from the preceding repetition, where V is the inertial mass index. $w_k(s-1)$ Represents the velocity vector $(s-1)$th, and $z_k(s-1)$ is represents the position vector $(s-1)$ th. d_1 and d_2 are constant training coefficients (heading in the direction of the best value of the particle being studied and the best-found particle in the entire population, respectively, rand 1 and rand 2 are two uniformly distributed random values in the region [0,1]. $(s-1)$.

It restrict velocity changes to the range w_{min} to w_{max}; $w_{min} \leq w \leq w_{max}$ are the upper and lower limit set by the type of problem, so that a particle's velocity does not exponentially rise as it travels from one location to another (the velocity divergence).

C. Training the Model:

1) MC-GRU

There are several MC layers and one GRU layer in the model. In addition, there is a batch normalizing layer and a pooling layer (BN). The MC extracts deep features from the data stream using a three-layer convolutional neural network with multiple convolution kernels. By pooling computing on the data from the convolutional layer, the processing efficiency of the lower network is enhanced, and the model converges more quickly. The BN helps the network model display nonlinear relationships more accurately. The model's processing performance can be improved by adding a long short-term memory (LSTM) based augmented GRU layer after the batch normalization layer so that it can learn the context and time series features of the input. To simplify the intricate coadaptive interaction between neurons, the dropout layer selectively eliminates part of them. Classification results are then generated from the SoftMax logistic regression layer, which is fed with the processed vector features for feature fusion.

2) MC LAYER

The CNN model takes the sample data as its input, and can apply any of their filters to it to conduct convolutional transformations. Three layers of convolution can be found in MC-GRU in a row [28]. There are 132, 96, and 64 convolution kernels, respectively. Using multiple convolutional networks in an iterative fashion allows for the collection of increasingly complex and insightful features of the global data flow. The linear rectification function Relu is used as the activation function in all three-layer convolutional networks. The output of the signature function is a linear combination of the input, but the sample data used to build stock price projections is not always linear. As a result, the sigmoid activation function is abandoned in favor of the rectified linear unit, or Relu. The time required to calculate the model could be decreased by employing the rule activation function. Proof that it enhances the effectiveness of training and detection is provided.

$$Relu(z) = \max(z, 0) \tag{7}$$

After the first layer of processing in a convolutional network is complete, it looks like this:

$$z_1 = \begin{cases} c_1 + v_1 * Z, & c_1 + v_1 * z > 0, \\ 0, & c_1 + v_1 * z \leq 0 \end{cases} \tag{8}$$

Following is the output of a convolutional network with two more layers:

$$z_n = \begin{cases} c_n + v_n * Z_{n-1}, & c_n + v_n * z_{n-1} > 0, \\ 0, & c_n + v_n * z_{n-1} \leq 0 \end{cases} \tag{9}$$

The Formula (8) stands for the eigenvalue of the detected and preprocessed traffic data; the matrices; c_1 and v_1 are the bias and weight matrices of the first layer of the convolutional network; the formula (9) stands for the output of the previous layer of the convolutional network;; c_n and v_n , are the bias and weight matrices of each layer of the convolutional network $m = 3, 4,$*

3) Pooling Layer:

All of the attributes from the preceding layer are combined in the pooling layer. As a result, after the pooling layer, less processing power and effort are required to make stock price predictions. The pooling filter retains the highest-scoring instance of each feature component from the input feature vector. How exactly they deduce it:

$$\begin{cases} G_{out} = \dfrac{g_{in} - g_{filter}}{S+1} \\ V_{out} = \dfrac{v_{in} - v_{filter}}{S+1} \end{cases} \tag{10}$$

T represents the step size of the pooling filter scan, g_{in} in represents the feature vector output from the layer before, g_{filter} represents the feature vector output from the pooling layer's height, v_{in} in represents the feature vector output from the layer before, and V_{out} represents the feature vector output from the pooling layer's width.

4) GRU:

A GRU network based on long short-term memory (LSTM) is developed to learn the contextual elements of data flow and timing information while also meeting the detection accuracy, processing power and real-time needs of wireless sensor nodes. Only two gates the update gate x_s and the reset gate q_s. The hidden layer unit's current state (h_{s-1}) is cleared by the reset gate.

$$q_s = \mu\left(V_q . [h_{s-1}, z_s], d_q\right) \tag{11}$$

Information retained after forgetting is denoted by $h_{s-1} \cdot q_s$ h_{s-1}: The update gate's job is to regulate the weight given to either the current input information or the previously stored state of the hidden layer (h_{s-1}):

$$x_s = \mu(V_s \,.\, [g_{s-1}, z_s], d_x) \tag{12}$$

After forgetting the $h_{s-1} \cdot q_s$, here it is:

$$\overline{g_s} = tang\left(V_{\widetilde{gs}} \cdot [Q_s * G_{s-1}, Z_s], d_g\right) \tag{13}$$

g_s after equilibrium:

$$g_s = (1 - x_s) * g_{s-1} + x_s * \overline{g_s} \tag{14}$$

When sigmoidal activation function and hyperbolic tangent function tanh are defined:

$$sigmoid(z) = \frac{1}{1 + e^{-z}} \tag{15}$$

$$tang(z) = \frac{1 - e^{-2z}}{1 + e^{2z}} \tag{16}$$

5) Dropout layer

To simplify the intricate coadaptation interaction between neurons, a certain number of cells must be removed in a predetermined ratio. Too many parameters and too few training data will cause over fitting in a neural network model. Using dropout in the stock price prediction model can improve the overall performance of the model by reducing the effects of overfitting. Here is the mathematical expression in full:

$$softmax(z_k) = \frac{e^{z_q}}{\sum_{j=1}^{m} e^{z_i}} \tag{17}$$

The suggested MC-GRU network model employs a $softmax$ function as the classifier in the last layer, with traffic classification determined by the computed probability value.

IV. RESULT AND DISCUSSION

In order to determine which algorithm is best at the challenging task of detecting fraudulent financial transactions, a number of metrics have been studied and contrasted. Three metrics accuracy, recall, and precision are used to evaluate the effectiveness of machine learning systems. A Confusion matrix can be used to calculate any of these quantities. All models were evaluated according to these standards. To emphasize the significance of sampling, the models were evaluated using both the original data and the oversampled data.

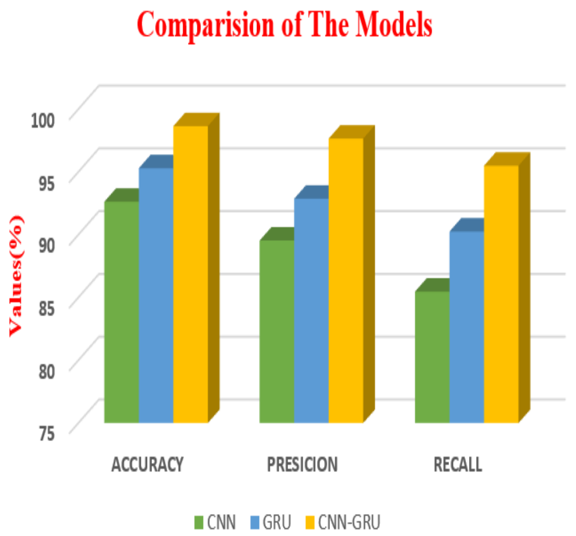

Fig. 1. Accuracy Comparision of the Models

In this proposed approach to tested the ability to predict the stock price using three distinct methods (CNN, GRU, and CNN-GRU). The results of the proposed model are shown in Figure 1; Results analysis reveals extremely high accuracy, however this does not equate to faultless performance. Ideally, other criteria should be considered in tandem with accuracy. It is shown that conventional algorithms, such CNN-GRU, may compete with even the most fundamental neural network in terms of accuracy and speed.

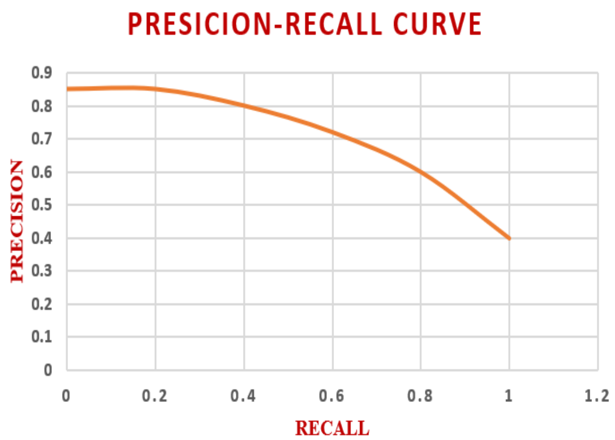

Fig. 2. Presicion-Recall Evaluation of CNN-GRU Model

It's important to keep in mind that it's not easy to discover a good solution when both precision and recall are high. Very high recall tends to come at the expense of low precision, and vice versa. Because of this, we also offer the F1-score, which allows for a quick and easy evaluation of which algorithm is best for the data. Test set precision-recall images are shown in Figure 2. Precision is represented by the X axis and recall by the Y axis.

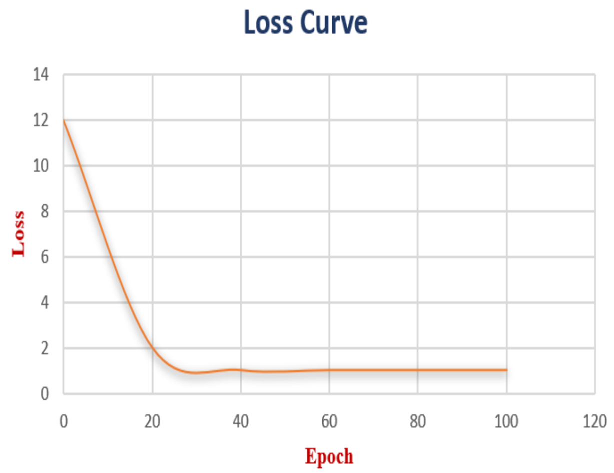

Fig. 3. Loss Value Curve of GRU-CNN

Optimize your training with a rate of 0.0001 and a batch size of 32. Figure 3 shows that after 100 epochs of training, the model had reached convergence. Three distinct models CNN, GRU, and CNN-GRU emerged from the model selection process. The experimental results strongly favor the CNN-GRU algorithm model over the other three alternatives.

V. CONCLUSION:

The growth of modern society is in large part due to the work of the financial markets. They make it possible to put out monetary assets. Stock price movements are a reflection of market conditions. One of the most common ways to put money to work is through stock investments, which offer the highest return of any major category of assets. Investors can make a lot of money off of stock price fluctuations. As stock market forecasting is a highly noisy problem, it is essential to make advantage of any available data that can improve accuracy. In this study, we extend the collection of characteristics for the daily return prediction issue in stocks to include indicators not only for the stock being predicted but also for a range of other stocks and currencies. After that, it's employ techniques for prediction like as preprocessing, feature selection, and classification. The suggested method employs MC-GRU for categorization. MC-GRU boosts the learning of spatial and time series features of traffic data and improves the detection performance of the model by using the high-level features output after convolution operations as input parameters of the GRU network. With an impressive 98.6 percent precision, the proposed technique performs wonderfully in practice.

REFERENCES

[1] Z. Jin, Y. Yang, and Y. Liu, "Stock closing price prediction based on sentiment analysis and LSTM," *Neural Comput. Appl.*, vol. 32, no. 13, pp. 9713–9729, 2020, doi: 10.1007/s00521-019-04504-2.

[2] J. YAO, C. L. TAN, and H.-L. POH, "Neural Networks for Technical Analysis: a Study on Klci," *Int. J. Theor. Appl. Financ.*, vol. 02, no. 02, pp. 221–241, 1999, doi: 10.1142/s0219024999000145.

[3] J. V. Hansen, J. B. Mcdonald, and R. D. Nelson, "Time series prediction with genetic-algorithm designed neural networks: an empirical comparison with modern statistical models," *Comput.*

Intell., vol. 15, no. 3, pp. 171–184, 1999, doi: 10.1111/0824-7935.00090.

[4] V. R. Prybutok, J. Yi, and D. Mitchell, "Comparison of neural network models with ARIMA and regression models for prediction of Houston's daily maximum ozone concentrations," *Eur. J. Oper. Res.*, vol. 122, no. 1, pp. 31–40, 2000, doi: 10.1016/S0377-2217(99)00069-7.

[5] Y. B. Wijaya, S. Kom, and T. A. Napitupulu, "Stock price prediction: Comparison of Arima and artificial neural network methods - An Indonesia stock's case," *Proc. - 2010 2nd Int. Conf. Adv. Comput. Control Telecommun. Technol. ACT 2010*, pp. 176–179, 2010, doi: 10.1109/ACT.2010.45.

[6] T. Hyup Roh, "Forecasting the volatility of stock price index," *Expert Syst. Appl.*, vol. 33, no. 4, pp. 916–922, 2007, doi: 10.1016/j.eswa.2006.08.001.

[7] H. White, "Economic prediction using neural networks: The case of IBM daily stock returns," pp. 451–458, 1988, doi: 10.1109/icnn.1988.23959.

[8] M. R. Hassan, B. Nath, and M. Kirley, "A fusion model of HMM, ANN and GA for stock market forecasting," *Expert Syst. Appl.*, vol. 33, no. 1, pp. 171–180, 2007, doi: 10.1016/j.eswa.2006.04.007.

[9] M. R. Hassan and B. Nath, "Stock market forecasting using Hidden Markov Model: A new approach," *Proc. - 5th Int. Conf. Intell. Syst. Des. Appl. 2005, ISDA '05*, vol. 2005, pp. 192–196, 2005, doi: 10.1109/ISDA.2005.85.

[10] D. RK and P. DD, "Application of Artificial Neural Network for Stock Market Predictions: a Review of Literature," *Int. J. Mach. Intell.*, vol. 2, no. 2, pp. 14–17, 2010, doi: 10.9735/0975-2927.2.2.14-17.

[11] A. Nawani, H. Gupta, and N. Thakur, "Prediction of Market Capital for Trading Firms through Data Mining Techniques," *Int. J. Comput. Appl.*, vol. 70, no. 18, pp. 7–12, 2013, doi: 10.5120/12165-8005.

[12] K. R. Fama, E. F., & French, "Size and book-to-market factors in earnings and returns," *J. Finance*, vol. 50, no. 1, pp. 131–155, 1995.

[13] S. Mondal, P., Shit, L., & Goswami, "Study of effectiveness of time series modeling (ARIMA) in forecasting stock prices," *Int. J. Comput. Sci. Eng. Appl.*, p. 13, 2014, doi: 10.1016/S0140-6736(01)13488-4.

[14] Q. Wu, Y. Chen, and Z. Liu, "Ensemble model of intelligent paradigms for stock market forecasting," *Proc. - 1st Int. Work. Knowl. Discov. Data Mining, WKDD*, pp. 205–208, 2008, doi: 10.1109/WKDD.2008.54.

[15] H. Jia, "Investigation Into The Effectiveness Of Long Short Term Memory Networks For Stock Price Prediction," pp. 1–6, 2016.

[16] M. P. Naeini, H. Taremian, and H. B. Hashemi, "Stock market value prediction using neural networks," *2010 Int. Conf. Comput. Inf. Syst. Ind. Manag. Appl. CISIM 2010*, pp. 132–136, 2010, doi: 10.1109/CISIM.2010.5643675.

[17] P. K. Mahato and V. Attar, "Prediction of gold and silver stock price using ensemble models," *2014 Int. Conf. Adv. Eng. Technol. Res. ICAETR 2014*, pp. 5–8, 2014, doi: 10.1109/ICAETR.2014.7012821.

[18] R. J. Kuo, C. H. Chen, and Y. C. Hwang, "An intelligent stock trading decision support system through integration of genetic algorithm based fuzzy neural network and artificial neural network," *Fuzzy Sets Syst.*, vol. 118, no. 1, pp. 21–45, 2001, doi: 10.1016/S0165-0114(98)00399-6.

[19] M. Aiken and M. Bsat, "Forecasting market trends with neural networks," *Inf. Syst. Manag.*, vol. 16, no. 4, pp. 42–48, 1999, doi: 10.1201/1078/43189.16.4.19990901/31202.6.

[20] H. Alostad, H., & Davulcu, "Directional prediction of stock prices using breaking news on twitter," *2015 IEEE/WIC/ACM Int. Conf. Web Intell. Intell. Agent Technol.*, vol. 1, pp. 523–530, 2015.

[21] X. Li, H. Xie, Y. Song, S. Zhu, Q. Li, and F. L. Wang, "Does Summarization Help Stock Prediction? A News Impact Analysis," *IEEE Intell. Syst.*, vol. 30, no. 3, pp. 26–34, 2015, doi: 10.1109/MIS.2015.1.

[22] C. T. Brownlees and G. M. Gallo, "Financial econometric analysis at ultra-high frequency: Data handling concerns," *Comput. Stat. Data Anal.*, vol. 51, no. 4, pp. 2232–2245, 2006, doi: 10.1016/j.csda.2006.09.030.

[23] E. J. Lee, "High Frequency Trading in the Korean Index Futures Market," *J. Futur. Mark.*, vol. 35, no. 1, pp. 31–51, 2015, doi: 10.1002/fut.21640.

[24] S. Chen and H. He, "Stock Prediction Using Convolutional Neural Network," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 435, no. 1, 2018, doi: 10.1088/1757-899X/435/1/012026.

[25] H. S. Sim, H. I. Kim, and J. J. Ahn, "Is Deep Learning for Image Recognition Applicable to Stock Market Prediction?," *Complexity*, vol. 2019, 2019, doi: 10.1155/2019/4324878.

[26] S. Asadi, E. Hadavandi, F. Mehmanpazir, and M. M. Nakhostin, "Hybridization of evolutionary Levenberg-Marquardt neural networks and data pre-processing for stock market prediction," *Knowledge-Based Syst.*, vol. 35, pp. 245–258, 2012, doi: 10.1016/j.knosys.2012.05.003.

[27] A. D. Kasbi, I. Dadashi, and K. Azinfar, "Stock price prediction using the Chaid rule-based algorithm and particle swarm optimization (pso)," *Adv. Math. Financ. Appl.*, vol. 5, no. 2, pp. 197–213, 2020, doi: 10.22034/amfa.2019.585043.1184.

[28] Z. Jingjing, Y. Tongyu, Z. Jilin, Z. Guohao, L. Xuefeng, and P. Xiang, "Intrusion Detection Model for Wireless Sensor Networks Based on MC-GRU," *Wirel. Commun. Mob. Comput.*, vol. 2022, 2022, doi: 10.1155/2022/2448010.