*Received Date:31st May, 2024*
*Revision Date:7th June, 2024*
*Accepted Date:15th July, 2024*

# Stock Market Prediction in Nepali Stock Market using Machine Learning Model

**Mausam Gurung[1*], Prabin Neupane[2], Sujan Shrestha[3]**
[1]*Dept of Electronics, Communication and Information Engineering, Kathmandu Engineering College,*
*Email:* mausaam.gurung593@gmail.com
[2] *Dept of Electronics, Communication and Information Engineering, Kathmandu Engineering College,*
*Email:* prabin.neupane699@gmail.com
[3] *Assoc. Professor, Dept of Electronics, Communication and Information Engineering, Kathmandu Engineering College,*
*Email:* sujan.shrestha@kecktm.edu.np

***Abstract*— The stock market, a key driver of the global economy, presents a formidable challenge for accurate prediction due to its intricate, chaotic, and dynamic nature. This study explores machine learning approaches, specifically comparing three prediction models—Back Propagation Neural Network (BPNN), Long Short-Term Memory (LSTM), and Gated Recurrent Unit (GRU). The analysis incorporates historical stock trading data, including open, high, low, and close prices, and technical analysis indicators such as moving average, Relative Strength Index (RSI), Moving Average Convergence Divergence (MACD), and Commodity Channel Index (CCI) so on. The evaluation focuses on the performance of these models in predicting future trends in stock prices, utilizing data from the Nepal Stock market spanning from 2013 to 2023. The selected stocks for analysis include two stock NIC Asia Bank Limited and, NMB Bank Limited, two index NEPSE and Hydropower index. Through rigorous assessment, it is determined that LSTM demonstrates superior overall performance compared to GRU and BPNN. This research contributes valuable insights into the effectiveness of machine learning models in forecasting stock market trends, with implications for investors and financial practitioners.**

***Keywords*— machine learning, stock market, BPNN, LSTM, GRU**

## Introduction

Predicting stock market trends has captivated the interest of not just traders, but also computer engineers. Stock market predictions typically leverage two primary methods: historical data analysis and the examination of social media information. Historical data analysis involves scrutinizing previous stock-related metrics, including opening and closing prices, high and low prices, adjusted closing prices, and trading volume.

The random walk theory [1] states that the price movement in the stock market are not predictable since they are determined by unexpected events with no correlation to the past. Furthermore, Efficient Market Hypothesis (EMH) [2] states that stock prices reflect all available information and cannot be beaten by analysis or trading strategies. The realm of stock marketplace prediction is difficult, given the multitude of complex financial indicators and the volatile nature of the market.

Despite the perceived intricacies and fluctuations within the market, technological progress is manifesting in tangible advancements, offering promising avenues for forecasting stock market behavior. This trend serves as a catalyst not only for investors and traders but also stimulates interest among computer engineers to explore and refine diverse machine learning models. [3]

Accurate market cost predictions are pivotal for maximizing the profitability of stock option investments whilst concurrently mitigating risks. It is evident from the above discussions that each of the algorithms in its own way can tackle this problem. It is also to be noticed that each of the algorithm has its own limitations. The final prediction outcome not only depends on the prediction algorithm used but is also influenced by the representation of the input. Identifying important features and using only them as the input rather than all the features may improve the prediction accuracy of the prediction.

This study focuses on comparing prediction performance of BPNN, LSTM, GRU the task of predicting stock and stock price index movement. LSTM have been doing research more than other model as this model is performing well due to its architecture. [4] [5] [6].

Ten technical parameters are used as the inputs to these models. The focus is to compare the performance of these prediction models when the inputs are represented in the form of real value. All the experiments are carried out using [10 years] of historical data of [two stocks NMB and NICA] and [two indices Nepse and Hydropower index]. Both stocks and indices are highly voluminous and vehemently traded in and so they reflect Nepal economy as a whole.

* *Corresponding Author*

The limitation is that according to Random Walk and EMH events are also the main reason but the study is to predict only through historical data and assuming that the events were reflected on the historical data.

## I. Literature Review

Researchers have explored an array of machine learning techniques like Support vector machines (SVM), LSTM, and regression to ever-evolving Artificial Neural Networks (ANN). ANN is one of the most widely used models reviewed by Mintarya et al. [3] and elaborated by Kim On et al. [7] to present a diverse range of approaches. Several researchers have explored various methods employing ANN models. For instance, Bing et al. employed Back-Propagation Neural Networks (BPNN) to predict the Shanghai Stock Exchange Composite Index [8]. Wensheng et al. compared Nonlinear Independent Component Analysis (NLICA) and BPNN for the Asian stock market [9]. Bailings et al. used random forest (RF), AdaBoost, kernel factory, NN, SVM, and k-nearest neighbors (KNN) to predict the stock market's direction for a year [10]. Patel et al. discussed several machine learning models, which are ANN, SVM, RF, and Naive Bayes, as well as made stock market index predictions using ANN, SVM, and RF [11]. Olivera et al. used a modified ANN to predict market behavior and stock market trends [12], and Li et al. compared Extreme Learning Machine (ELM) with SVM and BPNN, the result was that kernelized ELM and SVM had higher precision than BPNN and normal ELM [13].

Support Vector Machine (SVM) models have also found significant application in research. Ding et al. utilized SVM for forecasting stock market prices based on extensive public news data [14]. Meanwhile, Hegazy et al. conducted a comparative analysis between the least-squares SVM (LS-SVM) algorithm and particle swarm optimization (PSO) in the context of the financial sector [15]. Some researchers have also made modifications to the SVM model. For instance, Lin et al. assessed the performance of correlation-based SVM in comparison to quasi-linear SVM [16], and Ren et al. investigated the accuracy of SVM when integrated with sentiment analysis [17].

Another prevalent model, LSTM, has been extensively studied. Selvin et al. discussed their approaches involving Long Short-Term Memory (LSTM), Recurrent Neural Network (RNN), and Convolutional Neural Network with a sliding window (CNN-sliding window) [5]**.** Chen et al. focused on the utilization of the LSTM model [4]. Nelson et al. carried out a comparison between LSTM, Random Forest (RF), and Multilayer Perceptron (MLP) [6]. Hota et al. used American Airlines stock data on ANN, SVR, Random Forest and decision tree [18]. Additionally, G. et al. explored methods involving MLP, RNN, LSTM, and CNN [19] Moghar et al. used RNN-based LSTM [20]; Roondiwala et al. made a model using LSTM and RNN [21]; Kang et al. used a generative adversarial networks (GAN) model combined with MLP and LSTM [22]; Akita et al. used the LSTM approach with paragraph vector [23] ; Parmar et al. used Regression and LSTM on Indian stock exchange [24]. Jha et al used LSTM on Nepal stock data [25]

Several alternative models, including regression, can be employed for stock market prediction. Sharma et al. have explored various regression models in their work [26]. Furthermore, the utilization of support vector regression (SVR) optimized with a chaos-based firefly algorithm is examined by Kazem et al. [27]. Another approach involves employing the K-nearest neighbors (KNN) model for stock market forecasting, as demonstrated by Alkhatib et al., who applied the KNN algorithm and a non-linear regression approach to predict stock prices for six prominent companies listed on the Jordanian stock exchange [28]. Another approach using a random forest classifier and backtracking demonstrate by Bhamidipati et al. [29].

GRU model researched by Cho et al. 2014 [30] perform well same as LSTM reviewed by Bahadur Shahi et al. [31] using Nepali data NESPE and the news. Asiful Hossains et al. use the hybrid deep learning model of LSTM and GRU [32]. Chen et al. research to improve GRU-Based Stock price prediction [33]

## II. Research Data

*Data Description*

Ten years of data of total two stock indices (NEPSE Hydropower Index) and two stock (NMB, NICA) from 2013 to 2023 is used in this study. All the data is obtained from sharesansar.com/floorsheet websites. This data forms our entire dataset are shown in.

TABLE I
NICA

| | Symbol | Date | Open | High | Low | Close | % Change | Volume |
|---|---|---|---|---|---|---|---|---|
| 2380 | NICA | 2013-07-14 | 0 | 504 | 477 | 504 | 0 | 40 |
| 2379 | NICA | 2013-07-15 | 504 | 554 | 529 | 554 | 0 | 40 |
| 2378 | NICA | 2013-07-16 | 554 | 0 | 0 | 554 | 0 | 8,958.00 |
| 2377 | NICA | 2013-07-17 | 554 | 609 | 565 | 609 | 0 | 150 |
| 2376 | NICA | 2013-07-18 | 609 | 669 | 621 | 669 | 0 | 1,706.00 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 4 | NICA | 2023-12-12 | 509 | 517.8 | 504 | 510 | 0.02% | 62,527.00 |
| 3 | NICA | 2023-12-13 | 514 | 520 | 505.1 | 510 | 0.00% | 57,588.00 |
| 2 | NICA | 2023-12-14 | 508 | 511.8 | 501.6 | 506.9 | -0.61% | 40,832.00 |
| 1 | NICA | 2023-12-17 | 500 | 507 | 497 | 501 | -1.16% | 76,166.00 |
| 0 | NICA | 2023-12-18 | 504.5 | 519 | 501 | 515 | 2.79% | 139,478.00 |

2381 rows × 8 columns

TABLE II
NMB

| | Symbol | Date | Open | High | Low | Close | % Change | Volume |
|---|---|---|---|---|---|---|---|---|
| 2380 | NICA | 2013-07-14 | 0 | 504 | 477 | 504 | 0 | 40 |
| 2379 | NICA | 2013-07-15 | 504 | 554 | 529 | 554 | 0 | 40 |
| 2378 | NICA | 2013-07-16 | 554 | 0 | 0 | 554 | 0 | 8,958.00 |
| 2377 | NICA | 2013-07-17 | 554 | 609 | 565 | 609 | 0 | 150 |
| 2376 | NICA | 2013-07-18 | 609 | 669 | 621 | 669 | 0 | 1,706.00 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 4 | NICA | 2023-12-12 | 509 | 517.8 | 504 | 510 | 0.02% | 62,527.00 |
| 3 | NICA | 2023-12-13 | 514 | 520 | 505.1 | 510 | 0.00% | 57,588.00 |
| 2 | NICA | 2023-12-14 | 508 | 511.8 | 501.6 | 506.9 | -0.61% | 40,832.00 |
| 1 | NICA | 2023-12-17 | 500 | 507 | 497 | 501 | -1.16% | 76,166.00 |
| 0 | NICA | 2023-12-18 | 504.5 | 519 | 501 | 515 | 2.79% | 139,478.00 |

2381 rows × 8 columns

TABLE III
NEPSE

| | S.N. | Open | High | Low | Close | Change | Per Change (%) | Turnover | Date |
|---|---|---|---|---|---|---|---|---|---|
| **0** | 1 | 323 | 323 | 323 | 323 | -1.14 | -0.35 | 0 | 12/1/2011 |
| **1** | 2 | 322 | 322 | 322 | 322 | -0.46 | -0.14 | 0 | 12/4/2011 |
| **2** | 3 | 321 | 321 | 321 | 321 | -0.88 | -0.27 | 0 | 12/5/2011 |
| **3** | 4 | 321 | 321 | 321 | 321 | -0.32 | -0.10 | 0 | 12/6/2011 |
| **4** | 5 | 321 | 321 | 321 | 321 | -0.58 | -0.18 | 0 | 12/7/2011 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **2840** | 2841 | 1,971.36 | 1,974.58 | 1,960.92 | 1,965.93 | -3.24 | -0.16 | 2,366,975,594.11 | 4/25/2024 |
| **2841** | 2842 | 1,973.53 | 1,981.31 | 1,965.18 | 1,980.27 | 14.33 | 0.72 | 2,682,586,955.82 | 4/28/2024 |
| **2842** | 2843 | 1,990.31 | 1,990.31 | 1,974.93 | 1,980.34 | 0.07 | 0.00 | 2,959,458,501.43 | 4/29/2024 |
| **2843** | 2844 | 1,991.10 | 2,016.57 | 1,979.36 | 2,006.28 | 25.93 | 1.30 | 4,800,044,601.66 | 4/30/2024 |
| **2844** | 2845 | 2,025.78 | 2,025.78 | 1,995.84 | 1,998.96 | -7.31 | -0.36 | 4,133,362,726.23 | 5/2/2024 |

2845 rows × 9 columns

TABLE IV
Hydro

| | S.N. | Open | High | Low | Close | Change | Per Change (%) | Turnover | Date |
|---|---|---|---|---|---|---|---|---|---|
| **0** | 1 | 323 | 323 | 323 | 323 | -1.14 | -0.35 | 0 | 12/1/2011 |
| **1** | 2 | 322 | 322 | 322 | 322 | -0.46 | -0.14 | 0 | 12/4/2011 |
| **2** | 3 | 321 | 321 | 321 | 321 | -0.88 | -0.27 | 0 | 12/5/2011 |
| **3** | 4 | 321 | 321 | 321 | 321 | -0.32 | -0.10 | 0 | 12/6/2011 |
| **4** | 5 | 321 | 321 | 321 | 321 | -0.58 | -0.18 | 0 | 12/7/2011 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **2840** | 2841 | 1,971.36 | 1,974.58 | 1,960.92 | 1,965.93 | -3.24 | -0.16 | 2,366,975,594.11 | 4/25/2024 |
| **2841** | 2842 | 1,973.53 | 1,981.31 | 1,965.18 | 1,980.27 | 14.33 | 0.72 | 2,682,586,955.82 | 4/28/2024 |
| **2842** | 2843 | 1,990.31 | 1,990.31 | 1,974.93 | 1,980.34 | 0.07 | 0.00 | 2,959,458,501.43 | 4/29/2024 |
| **2843** | 2844 | 1,991.10 | 2,016.57 | 1,979.36 | 2,006.28 | 25.93 | 1.30 | 4,800,044,601.66 | 4/30/2024 |
| **2844** | 2845 | 2,025.78 | 2,025.78 | 1,995.84 | 1,998.96 | -7.31 | -0.36 | 4,133,362,726.23 | 5/2/2024 |

2845 rows × 9 columns

*Normalization*

The extracted data was then subjected to normalization to unify the data range within 0 and 1. Normalization of data is done to bring all stock data into a common range. Since we are using stock data from different market, we need the data to be under a common range. This process was done using the equation:

$$x_{norm} = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{1}$$

There are some technical indicators through which one can predict the future movement of stocks. Here in this study, total ten technical indicators as employed in Patel et al. (2014) [11] are used. These indicators are shown from table V to VIII shows summary statistics for the selected indicators of two indices and two stocks.

First two technical indicators are moving averages. The moving average (MA) is simple technical analyses tool that smoothies out price data by creating a constantly updated average price. In this paper, 14 day's Simple Moving Average (SMA) and Weighted Moving Average (WMA) are used as we are predicting short term future.

Using these indicator values, the input set is given to the predictor models. Performance of all the models under study is evaluated also for this representation of inputs.

*Technical parameters*

Simple n-day Moving Average (SMA): It is a calculation that averages the prices of a security over a specified period, providing a smoothed trend line to identify the overall direction of the price movement.

$$\text{Simple MA} = \frac{C_t + C_{t-1} + C_{t-2} + \cdots + C_t - C_{t-n+1}}{n} \tag{2}$$

Where,

$C_t$: Closing Price

n: number of days considered for the moving average

Weighted n-day Moving Average (WMA): Similar to SMA, but it assigns different weights to different data points, giving more significance to recent prices.

$$\text{Weighted MA} = \frac{(n-1)C_t + (n-2)C_{t-1} + (n-2)C_{t-2} + \cdots + (n+1)C_{t-n+1}}{\frac{n(n+1)}{n}} \tag{3}$$

*I)* *Momentum:* It measures the rate of change of a security's price and is used to identify the strength or weakness of a trend.

$$\text{Momentum} = C_t - C_{t-n} \tag{4}$$

Stochastic %K: A momentum indicator that compares the closing price of a security to its price range over a specific period, indicating the position of the current close relative to the high-low range.

$$K\% = \frac{H_h - L_l}{H_h - L_l} \times 100 \tag{5}$$

$H_h$: Highest high price in the specified period

$L_l$: Lowest low price in the specified period.

Stochastic %D: A smoothed version of the %K, providing a signal line to help identify potential buy or sell signals.

$$D\% = \frac{\sum_{k=0}^{n} K_{t-i}}{n}\% \tag{6}$$

Relative Strength Index (RSI): Measures the magnitude of recent price changes to evaluate overbought or oversold conditions, indicating potential reversal points.

$$\text{RSI} = 100 - \frac{100}{1 + \frac{\frac{\sum UP_{t-i}}{n}}{\frac{\sum DW_{t-i}}{n}}} \tag{7}$$

Up: Upward Price changes

DW: Downward Price changes

Moving Average Convergence Divergence (MACD): A trend-following momentum indicator that shows the relationship between two moving averages of a security's price.

$$MACD(n) = \frac{1}{n + \frac{1}{2\pi}} \times (DIFF_t - MACD(n)_{t-1}) \quad (8)$$

$DIFF_t$: Difference between two exponential moving averages (EMA)

Larry Williams R%: Also known as Williams Percent Range, it measures the level of the closing price relative to the high-low range over a specific period, helping identify overbought or oversold conditions.

$$R\% = \frac{H_h - C_t}{H_h - L_l} \times 100 \quad (9)$$

Accumulation/Distribution (A/D) Oscillator: It calculates the accumulation or distribution of a security by analyzing volume and price data, providing insights into buying or selling pressure.

$$\frac{A}{D} Oscillator = \frac{H_h - C_t}{H_h - L_l} \times V_t \quad (10)$$

Commodity Channel Index (CCI): A momentum oscillator that measures the current price level relative to its average price, helping identify overbought or oversold conditions and potential trend reversals.

$$CCI = \frac{M_t - SMA(n)}{0.015 \times SD(n)} \quad (11)$$

$M_t$: Midpoint price at time t (average of high, low, and closing prices)

SD: Standard Deviation over n days

TABLE V
NMB

| | Max | MIN | Mean | Std. |
|---|---|---|---|---|
| SMA | 897.2857143 | 150.5 | 369.0199 | 152.0668 |
| WMA | 898.9714286 | 149.8 | 368.9925 | 152.218 |
| Momentum | 204 | -194 | -0.18395 | 36.30405 |
| Stochastic %K | 100 | -3.78956E-14 | 46.26057 | 25.11447 |
| Stochastic %D | 98.94179894 | -1.77636E-14 | 46.24775 | 23.19943 |
| RSI | 90.46447843 | 14.43789405 | 48.91541 | 13.56527 |
| MACD | 45.58835175 | -51.81806312 | -0.03269 | 11.45501 |
| MACD Signal | 38.27747067 | -46.1487805 | -0.06057 | 10.91587 |
| Larry Williams R% | 0 | -100 | -54.9289 | 28.62548 |
| A/D Oscillator | 271311.0302 | -368553.583 | -14299.6 | 56267.27 |

TABLE VI
NICA technical Indicator

| | Max | MIN | Mean | Std. |
|---|---|---|---|---|
| SMA | 1040.635714 | 310.8571 | 642.529 | 190.7714 |
| WMA | 1038.717143 | 309.0952 | 642.3691 | 191.1305 |
| Momentum | 254 | -352 | -0.9815 | 65.34859 |
| Stochastic %K | 100 | -6.9E-14 | 44.90907 | 24.45112 |
| Stochastic %D | 97.89377704 | 0.871866 | 44.93349 | 22.61596 |
| RSI | 91.89942325 | 8.300548 | 49.93532 | 14.32121 |
| MACD | 71.96286564 | -72.5961 | -0.40825 | 19.19698 |
| MACD Signal | 62.3548561 | -62.478 | -0.38772 | 17.98109 |
| Larry Williams R% | 0 | -100 | -55.7716 | 28.00935 |
| A/D Oscillator | 442296.0328 | -424208 | -15907.8 | 62027.17 |

TABLE VII
NEPSE

| | Max | MIN | Mean | Std. |
|---|---|---|---|---|
| SMA | 3145.802 | 301.5 | 1425.235 | 668.3299 |
| WMA | 3158.899 | 301.1619 | 1426.517 | 668.007 |
| Momentum | 426.14 | -451.45 | 8.329812 | 97.9739 |
| Stochastic %K | 100 | 0 | 49.92066 | 31.90748 |
| Stochastic %D | 100 | -1.06E-13 | 49.91415 | 29.88661 |
| RSI | 95.9052 | 11.32907 | 53.48499 | 16.33661 |
| MACD | 111.0821 | -104.787 | 4.226957 | 30.90907 |
| MACD Signal | 100.5305 | -90.6886 | 4.248659 | 29.31716 |
| Larry Williams R% | 2.970989 | -101.162 | -48.1012 | 34.93224 |
| A/D Oscillator | 317.094 | -331.303 | 6.577553 | 111.4875 |

Table VIII Hydro

| | Max | MIN | Mean | Std. |
|---|---|---|---|---|
| SMA | 3145.802 | 301.5 | 1425.235 | 668.3299 |
| WMA | 3158.899 | 301.1619 | 1426.517 | 668.007 |
| Momentum | 426.14 | -451.45 | 8.329812 | 97.9739 |
| Stochastic %K | 100 | 0 | 49.92066 | 31.90748 |
| Stochastic %D | 100 | -1.06E-13 | 49.91415 | 29.88661 |
| RSI | 95.9052 | 11.32907 | 53.48499 | 16.33661 |
| MACD | 111.0821 | -104.787 | 4.226957 | 30.90907 |
| MACD Signal | 100.5305 | -90.6886 | 4.248659 | 29.31716 |
| Larry Williams R% | 2.970989 | -101.162 | -48.1012 | 34.93224 |
| A/D Oscillator | 317.094 | -331.303 | 6.577553 | 111.4875 |

**Machine Learning Model for prediction:**

*BPNN*

The BP Neural Network is a kind of one-way transmission of multilayer feed forward neural network, with one or more layers of hidden nodes, besides input and output nodes in its structure. There is no connection between nodes on the same level. We can treat it as a highly nonlinear mapping from input to output. Our model uses learning algorithm, gradient search techniques in the learning process and the error back propagation to modify weights, to achieve the minimum of the output error. The following diagram shows a usual BP neural network model with one hidden layer.

Rectified Linear Unit ReLU activation function is used in the hidden layer on this study

$$ReLU(x) = max(0, x) \quad (12)$$

Algorithm for back Propagation neural network:

Step (i): Initialize all weights ((W)) and biases ((b)) randomly.

Step (ii): Consider the training dataset and corresponding input as ($I_i = O_i$).

Step (iii): Perform a forward pass to calculate hidden layer values:

$$I_h = \sum (W_{ij} \times O_i) + b_j \quad (13)$$

$$O_j = \frac{1}{1 + e^{-I_j}} \quad (14)$$

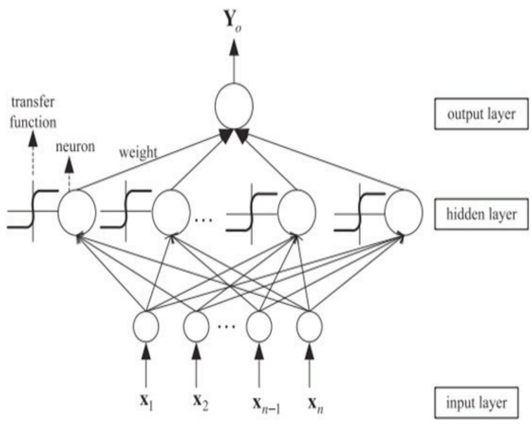


Figure 1 Simply BPNN topology [9].

Step (iv): Calculate output layer values:

$$I_k = \sum\left(W_{jk} \times O_j\right) + b_k \tag{15}$$

$$O_k = \frac{1}{1 + e^{-I_k}} \tag{16}$$

Step (v): Calculate error at each layer:

Output layer error:

$$E_k = O_k(1 - O_k)(T - O_k)\sum\left(W_{jk} \times O_j\right) + b_k \tag{17}$$

Hidden layer error:

$$E_j = O_j\left(1 - O_j\right)\sum\left(E_k \times W_{jk}\right) \tag{18}$$

Step (vi): Update weights and biases as,

- Change in hidden layer weight and bias

$$\Delta W_{ij} = \alpha \times E_j \times O_i \tag{19}$$

$$W_{ij} = W_{ij} + \Delta W_{ij} \tag{20}$$

$$\Delta b_j = \alpha \times E_j \tag{21}$$

$$b_j = b_j + \Delta b_j \tag{22}$$

- Change in output layer weight.

$$\Delta W_{jk} = \alpha \times E_k \times O_j \tag{23}$$

$$W_{jk} = W_{jk} + \Delta W_{jk} \tag{24}$$

$$\Delta b_k = \alpha \times E_k \tag{25}$$

$$b_k = b_k + \Delta b_k \tag{26}$$

Step (vii): Repeat steps (ii) to (vi) until all training set values are satisfied.

*LSTM:*

LSTM networks, which stand for Long Short-Term Memory networks were created as an improvement, over neural networks (RNNs) to tackle the issues of long-term dependency and vanishing gradient. They excel in processing data like time series, text, and speech by utilizing valuable information from previous data points.

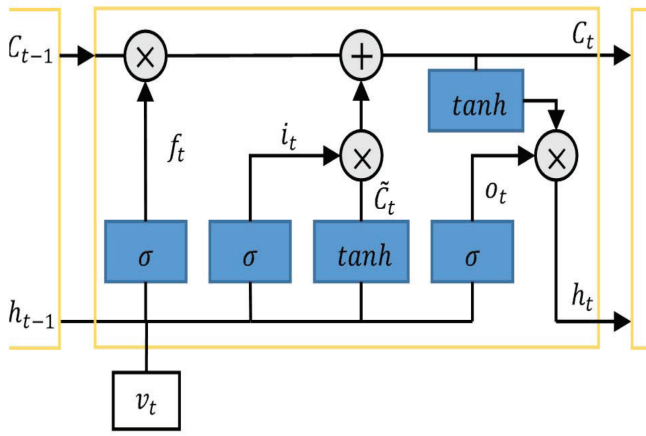


Figure 2 The internal structure of an LSTM [34]

**Forget Gate (Ft):** Determines which information to discard from the cell state.

$$F_t = \sigma\left(W_f\left[h_{t-1}, x_t\right] + b_f\right) \tag{27}$$

**Input Gate (It):** Decides which new information to store in the cell state1.

$$i_t = \sigma\left(W_i\left[h_{t-1}, x_t\right] + b_i\right) \tag{28}$$

**Candidate Memory ($\tilde{C}_t\tilde{C}_t$):** Creates a vector of new candidate values that could be added to the state2.

$$\tilde{C}_t = \tanh\left(W_c\left[h_{t-1}, x_t\right] + b_c\right) \tag{29}$$

**Cell State Update ($C_t$):** Updates the old cell state to the new cell state3.

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \tag{30}$$

**Output Gate ($O_t$):** Determines which parts of the cell state to output4.

$$O_t = \sigma\left(W_o\left[h_{t-1}, x_t\right] + b_o\right) \tag{31}$$

**Hidden State ($h_t$):** The filtered output based on the cell state and output gate5.

$$h_t = O_t * \tanh\left(C_t\right) \tag{32}$$

These equations collectively describe the LSTM's mechanism for processing data through time while maintaining long-term dependencies.

*GRU:*

Introduced by Cho et al. in 2014 [30] GRU (Gated Recurrent Unit) aims to solve the vanishing gradient problem which comes with a standard recurrent neural network. GRU is a simplified version of LSTM, reducing the three gates in LSTM to two where GRU combines the forget and the input gates into a single update gate. It also merges the cell state and the hidden state and makes some other changes. [35] GRU model is simpler yet faster network than the standard LSTM models although the basic purpose of using GRU is similar as LSTM.

Consequently, the GRU exhibits enhanced proficiency in capturing and learning long-term dependencies in time-series data while also reducing model complexity and computational costs, thus providing superior training efficiency. The improved ability of the GRU to handle long-term dependencies in time-series data makes it the preferred choice. Additionally, the GRU has lower storage requirements, rendering it suitable for processing large-scale datasets. Therefore, the basic GRU model was selected as the primary model in this study. The architecture of the GRU model is illustrated in figure 3.

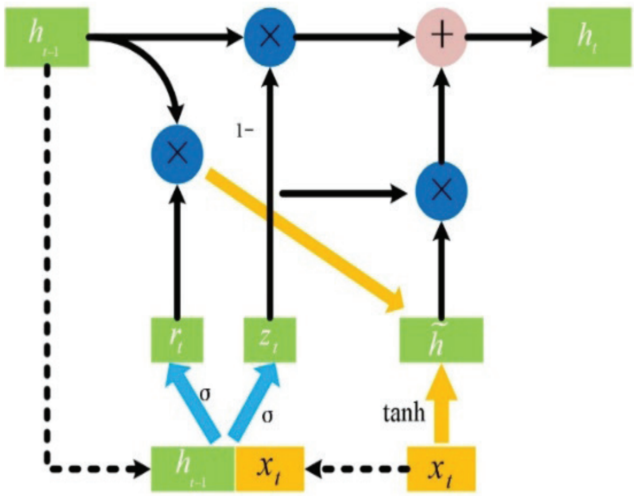


Figure 3 Internal structure of GRU [33]

Equation of GRU:

$$r_t = \sigma(W_r x_t + U_r h_{t-1} + b_r) \tag{33}$$

$$z_t = \sigma(W_z x_t + U_z h_{t-1} + b_z) \tag{34}$$

$$\tilde{h}_t = \tanh(W x_t + U(r_t * h_{t-1}) + b_h) \tag{35}$$

$$h_t = z_t * \tilde{h}_t + (1 - z_t) * h_{t-1} \tag{36}$$

Here, * represents the element-wise product formula; $W_r$ and $W_z$ are the weight matrices of the $r_t$ gate and the $z_t$ gate, respectively; $U_h$ represents the weight matrix for the output; $x_t$ represents the input data at time t; ~ $\tilde{h}_t$ and $h_t$ represent the candidate state and output state at time t; $b_r$ , $b_z$, and $b_h$ are constants; and σ and tanh are the sigmoid and tanh activation functions, respectively, used to activate the control gates and candidate states.

**Methodology**

A. *Architecture of Methodology*

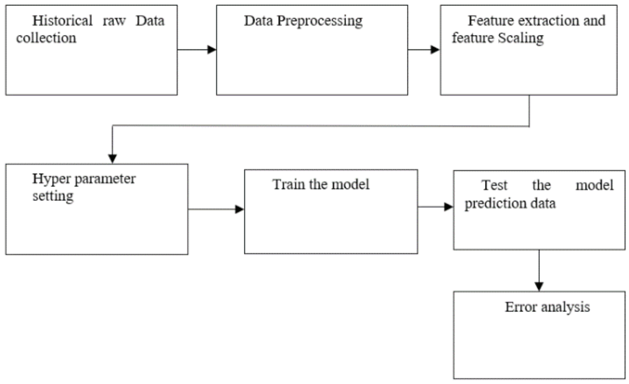


Figure 4 Architecture of Methodology

Historical raw data is collected from ShareSansar.com, a platform providing stock market information. This data

serves as the foundation for subsequent analysis and prediction.

Normalization technique is applied to preprocess the collected data. This step ensures that all features have the same scale, which is essential for effective model training and prediction. Features such as Last Traded Price (LTP), Open, High, and Low are extracted from the raw data. Additionally, technical parameters are calculated based on this data to create further features. These technical parameters might include indicators like Moving Averages, Relative Strength Index (RSI), or MACD (Moving Average Convergence Divergence).

Three chosen models—BPNN, LSTM, and GRU—were trained on the preprocessed data. Each model underwent training with varying epochs and parameters specific to its architecture. BPNN leverages backpropagation for optimizing weights, LSTM networks are adept at capturing long-term dependencies, and GRU utilizes gating mechanisms to efficiently capture sequential dependencies for robust predictions.

The preprocessed data is divided into training and testing sets using an 80/20 split. 80% of the data is allocated for training the models, while the remaining 20% is set aside for testing the trained models.

To provide a good balance between having enough data to train the model effectively and a sufficient amount to test its performance, the research implements an 80-20 train-test split. To ensures the model captures patterns and trends from historical data (80%) and evaluates its generalization on future data (20%), maintaining the temporal order, which is crucial for time series analysis. This standard practice helps make a reliable model performance assessment while avoiding underfitting or unreliable metrics.

In hyperparameter tuning, the research leverages the Adam optimizer, Mean Squared Error (MSE), and varying epochs for their efficacy in improving convergence and performance, especially in addressing complex tasks. Adam's adaptive learning rates enable rapid adaptation to complex patterns, which is vital for dynamic environments like stock market prediction. Meanwhile, MSE acts as a reliable metric for evaluating predictive accuracy, aiding in informed decision-making within trading strategies.

The performance of each trained model is evaluated using the testing data by calculating metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE). These error metrics assess the accuracy of the models in predicting stock prices, providing insights into how well the models generalize to unseen data. This evaluation process aids in selecting the best-performing model for further analysis and deployment.

**Experiment results**

Error analysis is used to evaluate the performance of proposed models. Prediction value and actual values helps to calculate the following error analysis.

*A Error analysis parameter: [11]*

Mean Absolute Percentage Error (MAPE):

$$MAPE = \frac{1}{n}\sum_{i=1}^{n} \left| \frac{A_i - P_i}{A_i} \right| \times 100 \tag{37}$$

Mean Absolute Error (MAE):

$$MAE = \frac{1}{n}\sum_{i=1}^{n} |A_i - P_i| \tag{38}$$

Mean Squared Error (MSE):

$$MSE = \frac{1}{n}\sum_{i=1}^{n} (A_i - P_i)^2 \tag{39}$$

Root Mean Squared Error (rRMSE):

$$rRMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n} \left(\frac{A_i - P_i}{A_i}\right)^2} \tag{40}$$

To evaluate the model, we have four different types of datasets. We use two approaches

1. Different features for the same model with the same dataset
2. We approach different hyperparameter settings to see the performance of the model

Below are the two types of result we find out

*B Features selection [FS1]:*

Ltp, High, Open, Low

TABLE IX

NICA

| Model | Hyper Parameter | | RMSE | MAE | MAPE |
|---|---|---|---|---|---|
| | Batch | Epoch | | | |
| BPNN | 10 | 10 | 22.925 | 16.287 | 2.329 |
| | 10 | 100 | 32.199 | 28.132 | 3.917 |
| | 100 | 250 | 25.45 | 17.57 | 2.474 |
| LSTM | 10 | 10 | 19.276 | 13.874 | 1.912 |
| | 10 | 100 | 14.998 | 10.048 | 1.412 |
| | 100 | 250 | 14.319 | 8.872 | 1.254 |
| GRU | 10 | 10 | 48.919 | 46.473 | 6.527 |
| | 10 | 100 | 19.092 | 14.804 | 2.072 |
| | 100 | 250 | 15.790 | 10.391 | 1.436 |

Table X

NMB

| Model | Hyper Parameter | | RMSE | MAE | MAPE |
|---|---|---|---|---|---|
| | Batch | Epoch | | | |
| BPNN | 10 | 10 | 15.2116 | 13.242 | 4.944 |
| | 10 | 100 | 7.35 | 4.89 | 1.737 |
| | 100 | 250 | 7.626 | 5.639 | 2.077 |
| LSTM | 10 | 10 | 5.471 | 3.938 | 1.48 |
| | 10 | 100 | 4.849 | 3.264 | 1.228 |
| | 100 | 250 | 4.809 | 3.212 | 1.198 |
| GRU | 10 | 10 | 4.995 | 3.343 | 1.256 |
| | 10 | 100 | 9.312 | 8.547 | 3.370 |
| | 100 | 250 | 4.801 | 3.156 | 1.181 |

Table XI

NEPSE

| Model | Hyper Parameter | | RMSE | MAE | MAPE |
|---|---|---|---|---|---|
| | Batch | Epoch | | | |
| BPNN | 10 | 10 | 15.211 | 13.242 | 4.944 |
| | 10 | 100 | 7.35 | 4.89 | 1.737 |
| | 100 | 250 | 7.626 | 5.639 | 2.077 |
| LSTM | 10 | 10 | 36.568 | 29.956 | 1.416 |
| | 10 | 100 | 36.568 | 29.956 | 1.416 |
| | 100 | 250 | 45.261 | 35.371 | 1.652 |

| | | | | | |
|---|---|---|---|---|---|
| GRU | Batch | Epoch | | | |
| | 10 | 10 | 69.286 | 60.914 | 2.886 |
| | 10 | 100 | 33.932 | 26.926 | 1.263 |
| | 100 | 250 | 34.549 | 26.873 | 1.259 |

TABLE XII

HYDRO

| Model | Hyper Parameter | | RMSE | MAE | MAPE |
|---|---|---|---|---|---|
| BPNN | Batch | Epoch | | | |
| | 10 | 10 | 56.76 | 43.29 | 2.012 |
| | 10 | 100 | 96.318 | 85.01 | 4.075 |
| | 100 | 250 | 76.302 | 56.859 | 2.641 |
| LSTM | Batch | Epoch | | | |
| | 10 | 10 | 30.772 | 23.024 | 1.076 |
| | 10 | 100 | 30.894 | 23.284 | 1.090 |
| | 100 | 250 | 41.404 | 31.464 | 1.469 |
| GRU | Batch | Epoch | | | |
| | 10 | 10 | 90.689 | 72.093 | 2.797 |
| | 10 | 100 | 65.345 | 46.436 | 1.759 |
| | 100 | 250 | 72.062 | 57.303 | 2.22 |

### *C Features selection [FS2]:*

Open, High, Low, SMA, WMA, Momentum, Stochastic %K, Stochastic %D, RSI, MACD, MACD signal, Larry Williams R%, A/D Oscillator, CCI: LTP

| Model | Hyper Parameter | | RMSE | MAE | MAPE |
|---|---|---|---|---|---|
| BPNN | Batch | Epoch | | | |
| | 10 | 10 | 24.207 | 16.420 | 2.374 |
| | 10 | 100 | 31.108 | 25.260 | 3.597 |
| | 100 | 250 | 27.866 | 20.550 | 2.950 |
| LSTM | Batch | Epoch | | | |
| | 10 | 10 | 17.942 | 12.515 | 1.782 |
| | 10 | 100 | 15.680 | 10.158 | 1.422 |
| | 100 | 250 | 20.735 | 15.781 | 2.222 |
| GRU | Batch | Epoch | | | |
| | 10 | 10 | 19.339 | 14.416 | 2.029 |
| | 10 | 100 | 20.959 | 16.138 | 2.294 |
| | 100 | 250 | 15.943 | 10.827 | 1.521 |

Table XIII

NMB

| Model | Hyper Parameter | | RMSE | MAE | MAPE |
|---|---|---|---|---|---|
| BPNN | Batch | Epoch | | | |
| | 10 | 10 | 11.718 | 9.255 | 3.583 |
| | 10 | 100 | 10.211 | 7.796 | 3.051 |
| | 100 | 250 | 26.412 | 22.872 | 8.891 |
| LSTM | Batch | Epoch | | | |
| | 10 | 10 | 5.726 | 4.193 | 1.616 |
| | 10 | 100 | 6.079 | 4.739 | 1.838 |
| | 100 | 250 | 9.909 | 8.433 | 3.165 |
| GRU | Batch | Epoch | | | |
| | 10 | 10 | 6.3465 | 6.3465 | 1.850 |
| | 10 | 100 | 6.063 | 6.063 | 1.755 |
| | 100 | 250 | 9.211 | 9.211 | 3.021 |

Table XIV

NEPSE

| Model | Hyper Parameter | | RMSE | MAE | MAPE |
|---|---|---|---|---|---|
| BPNN | Batch | Epoch | | | |
| | 10 | 10 | 67.625 | 58.816 | 2.834 |
| | 10 | 100 | 43.577 | 32.992 | 1.531 |
| | 100 | 250 | 83.094 | 64.348 | 3.017 |
| LSTM | Batch | Epoch | | | |
| | 10 | 10 | 34.743 | 25.432 | 1.179 |
| | 10 | 100 | 38.661 | 28.821 | 1.350 |
| | 100 | 250 | 35.600 | 27.226 | 1.275 |
| GRU | Batch | Epoch | | | |
| | 10 | 10 | 109.261 | 95.712 | 4.456 |
| | 10 | 100 | 44.391 | 34.590 | 1.634 |
| | 100 | 250 | 43.933 | 36.182 | 1.714 |

Table XV

Hydro

| Model | Hyper Parameter | | RMSE | MAE | MAPE |
|---|---|---|---|---|---|
| BPNN | Batch | Epoch | | | |
| | 10 | 10 | 67.625 | 58.816 | 2.834 |
| | 10 | 100 | 43.577 | 32.992 | 1.530 |
| | 100 | 250 | 83.094 | 64.348 | 3.017 |
| LSTM | Batch | Epoch | | | |
| | 10 | 10 | 30.772 | 25.432 | 1.179 |
| | 10 | 100 | 30.894 | 28.821 | 1.350 |
| | 100 | 250 | 41.404 | 27.226 | 1.275 |

| | | | | | |
|---|---|---|---|---|---|
| GRU | Batch | Epoch | | | |
| | 10 | 10 | 82.888 | 61.420 | 2.378 |
| | 10 | 100 | 81.170 | 57.335 | 2.189 |
| | 100 | 250 | 72.205 | 51.066 | 1.937 |

D. Result Graph: Feature selection 1 [FS1]

NICA

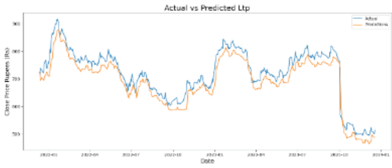


Figure 5: BPNN 100 epoch

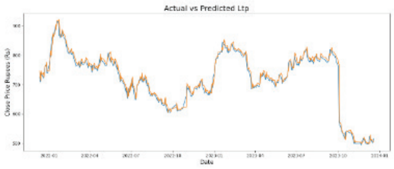


Figure 6: LSTM 100 epochs

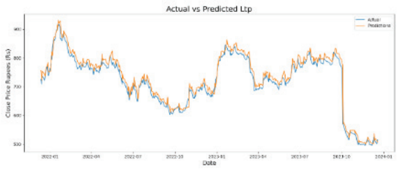


Figure 7: GRU 100 epochs

**NMB**

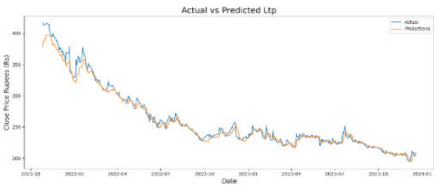


Figure 8: BPNN 100 epochs

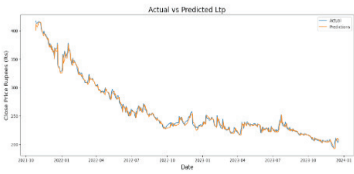


Figure 9: LSTM 100 epochs

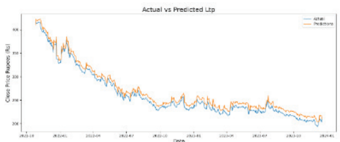


Figure 10: GRU 100 epochs

**NEPSE**

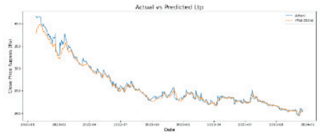


Figure 11: BPNN 100 epochs


Figure 12: LSTM 100 epochs

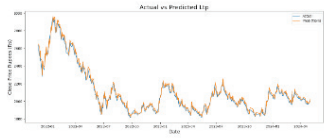


Figure 13: GRU 100 epochs

**Hydro**

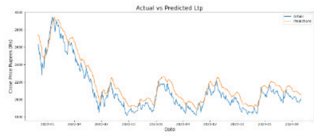


Figure 14: BPNN 100 epochs

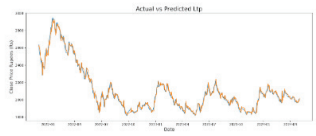


Figure 15: LSTM 100 epochs

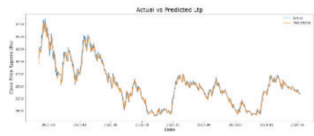


Figure 16: GRU 100 epochs

E. Feature selection 2 [ FS2]

NICA


Figure 17: BPNN 100 epochs


Figure 18: LSTM 100

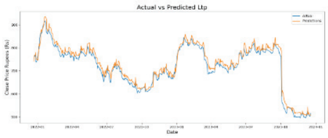

Figure 19: GRU 100 epochs

**NMB**

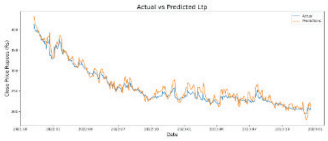

Figure 20: BPNN 100 epochs

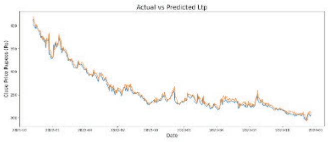

Figure 21: LSTM 100 epochs

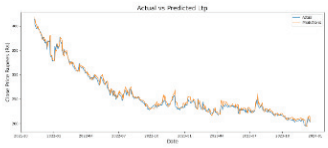

Figure 22: GRU 100 epochs

**NEPSE**

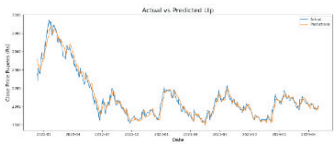

Figure 23: BPNN 100 epochs

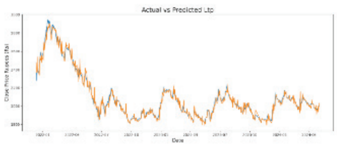

Figure 24: LSTM 100 epochs

Figure 25: GRU 100 epochs

**Hydro**

Figure 26: BPNN 100 epochs

Figure 27: LSTM 100

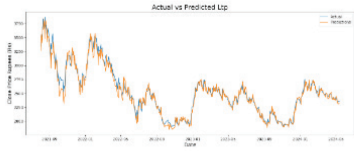

Figure 28: GRU 100 epochs

## Discussion

Stock market data is an example of non-stationary data. At particular time there can be trends, cycles, random walks or combinations of the three. It is desired that if a particular year is part of a cycle say a bullish one then our model should follow this pattern for trend prediction. Same can be considered for a trending year. However, usually stock values of a particular year are not isolated and there are days with random walks. Stock values are also affected by external factors creating trends and state of the country' s economy. Political scenarios are also the influencing factors which may result in cycles.

During our experimentation with BPNN, LSTM and GRU models, we tested varying epochs ranging from 10 to 250, observing improved performance with increasing epochs, particularly in capturing market trends. We utilized the Adam optimizer and mean square loss function for both models.

Our findings highlight the variability in model performance across different datasets, emphasizing that there isn't a one-size-fits-all model. Notably, while GRU achieved a low 2% error rate on some datasets, its performance on others, like the hydro dataset, exhibited higher error rates above 2%. Additionally, we observed that smaller batch sizes, particularly with 100 epochs, performed better than larger batch sizes of 250 epochs, underscoring the significance of batch size in model training.

The fundamental differences in model architectures were also evident. BPNN, reliant on backpropagation for weight and bias updates, showed competitive performance but was surpassed by LSTM's ability to retain and utilize past information effectively as well as by GRU a light weight version of LSTM.

Furthermore, as we increased the number of features, we noted varying impacts on model performance. GRU and BPNN generally improved with more features, while LSTM's performance remained stable. Notably, the choice of feature selection methods influenced model performance, with NMB performing well and FS1 demonstrating lower efficacy compared to FS2.

## Conclusion

The LSTM model's superior performance, attributed to its memory system, stands out among the BPNN and GRU models in forecasting stock market trends. However, it's crucial to note the inherent variability across datasets, emphasizing the absence of a universally perfect prediction model. Despite this, our LSTM model consistently achieved an impressive average accuracy of 98%, with a mere 2%

error rate, showcasing its reliability. The importance of optimizing batch size and epoch selection during training cannot be understated, as these parameters significantly impact model efficiency and convergence. Notably, the LSTM model outperformed the BPNN, while GRU model also demonstrated promising results, highlighting its ability to handle non-linear relationships and feature importance in stock market forecasting. Overall, while each model has strengths and weaknesses, the LSTM and GRU models emerge as robust choices for accurate stock market predictions

## References


[1] B. G. Malkiel, A Random Walk Down Wall Street, 1973.

[2] E. Fama, "Efficient capital markets: a review of theory and empirical work," *The Journal of Finance,* vol. 25 pp. 383–417, 1970.

[3] J. N. H. A. S. A. A. K. Latrisha N. Mintaryaa, "Machine learning approaches in stock market prediction: A systematic literature review," in *International Conference on Computer Science and Computational Intelligence*, 2023.

[4] A. L.-b. m. f. s. r. p. :. A. c. s. o. C. s. market, "A LSTM-based method for stock returns prediction :," in *2015 IEEE International Conference on Big Data (Big Data)*, Santa Clara, CA, USA, 2015.

[5] V. R. E. G. V. K. M. Sreelekshmy Selvin, "Stock prediction using LSTM, RNN and CNN-sliding window model," in *International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, 2017.

[6] A. C. M. P. R. A. d. O. David M. Q. Nelson, "Stock Market's Price Movement Prediction With LSTM Neural Networks," in *2017 International Joint Conference on Neural Networks (IJCNN)*, Anchorage, AK, USA, 2017.

[7] R. A. ,. P. A. Chin Kim On, "A review of stock market prediction with Artificial neural network (ANN)," in *IEEE International Conference on Control System, Computing and Engineering,*, 2013.

[8] H. J. Z. S. Bing Yang, "Stock Market Prediction Using Artificial Neural Networks," in *Trans Tech Publications Ltd*, Switzerland, 2012.

[9] J.-Y. W. C.-J. L. Wensheng Dai, "Combining nonlinear independent component analysis and neural network for," *Expert Systems with Applications,* 2012.

[10] D. V. d. P. N. H. R. G. Michel Ballings, "Evaluating multiple classifiers for stock price direction prediction," *Expert Systems with Applications,* 2015.

[11] S. S. P. T. K. K. Jigar Patel, "Predicting stock market index using fusion of machine learning," *Expert Systems with Applications,* 31 October 2014.

[12] C. N. N. L. E. Z. Fagner A. de Oliveira, "Applying Artificial Neural Networks to prediction of stock price and improvement of the directional prediction index – Case study of PETR4, Petrobras, Brazil," *Expert Systems with Applications,* 2013.

[13] H. X. ,. R. W. ,. C. ,. C. ,. F. W. ,. M. ,. D. Xiaodong Li, "Empirical analysis: stock market prediction via extreme learning," *Extreme Learning Machine and Application,* 2014.

[14] Y. Z. T. L. J. D. Xiao Ding, "Using Structured Events to Predict Stock Price Movement:," *Conference on Empirical Methods in Natural Language Processing (EMNLP),* p. 1415–1425, 2014.

[15] O. S. S. a. M. A. S. Osman Hegazy, "A Machine Learning Model for Stock Market," *International Journal of Computer Science and Telecommunications,* vol. 4, no. 12, 2013.

[16] H. G. a. J. H. Yuling LIN, "An SVM-based Approach for Stock Market Trend Prediction," in *International Joint Conference on Neural Networks (IJCNN)*, 2013.

[17] M. D. D. W. a. T. L. Rui Ren, "Forecasting Stock Market Movement Direction Using," *IEEE SYSTEMS JOURNAL,* vol. 13, no. 1, 2019.

[18] S. C. ,. B. K. P. ,. a. H. B. Jagruti Hota, "Stock Market Prediction Using Machine Learning Techniques," in *Advances in Computation Intelligence, its Concepts & Applications at ISIC*, 2022.

[19] G. E. V. K. M. S. K. Hiransha M, "NSE Stock Market Prediction Using Deep-Learning Models," *International Conference on Computational Intelligence and Data Science(ICCIDS),* 2018.

[20] M. H. Adil MOGHAR, "Stock Market Prediction Using LSTM Recurrent Neural Network," *Procedia Computer Science,* vol. 170, pp. 1168-1173, 2020.

[21] H. P. S. V. Murtaza Roondiwala, "Predicting Stock Prices Using LSTM," *International Journal of Science and Research (IJSR),* vol. 6, no. 4, 2017.

[22] G. Z. J. D. S. W. Y. W. Kang Zhanga, "Stock Market Prediction Based on Generative Adversarial Network," in *International Conference on Identification, Information and Knowledge in the Internet of Things*, 2018.

[23] A. Y. T. M. K. U. Ryo Akita, "Deep Learning for Stock Prediction Using Numerical and Textual Information," in *2016 IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS)*, 2016.

[24] G. E. V. K. M. S. K. Hiransha M, "NSE Stock Market Prediction Using Deep-Learning Models," *Procedia Computer Science,* vol. 132, pp. 1351-1362, 2018.

[25] P. B. S. K. S. S. Neha Jha, "Subarna Shakya," in *KEC Conference* , Stock Price Prediciton of NEpal using LSTM.

[26] A. Sharma, D. Bhuriya and U. Singh, "Survey of stock market prediction using machine learning approach," in *2017 International conference of Electronics, Communication and Aerospace Technology (ICECA)*, Coimbatore, India, 2017.

[27] E. S. F. K. H. M. S. Ahmad Kazema, "Support vector regression with chaos-based firefly algorithm for stock market," *Applied Soft Computing,* vol. 13, no. 2, pp. 947-958, 2013.

[28] N. H. I. H. M. K. A. S. Khalid Alkhatib, "Stock Price Prediction Using K-Nearest Neighbor (kNN) Algorithm," *International Journal of Business, Humanities and Technology ,* vol. 3, 2013.

[29] S. D. Venkata Sai P Bhamidipati, "Stock Price Prediction using Random Forest Classifier and Backtesting," in *2023 International Conference on Computer Communication and Informatics (ICCCI)*, 2023.

[30] D. B. F. B. H. S. Y. B. Kyunghyun Cho, "Learning Phrase Representations using RNN Encoder–Decoder," in *Yoshua Bengio*, 2014.

[31] A. S. A. N. a. W. G. Tej Bahadur Shahi, "Stock Price Forecasting with Deep Learning: A Comparative Study," in *Mathematics 2020,*, 2020.

[32] R. K. R. T. R. T. Y. W. Mohammad Asiful Hossain, "the Eighth International Symposium," in *Rezaul Karim*, 2018.

[33] L. X. a. W. X. Chi Chen, "Research on Improved GRU-Based Stock Price," *Applied Science,* vol. 13, 2023.

[34] B. S. L. P. d. L. a. A. G. E. Manuel R. Vargas, "Deep learning for stock market prediction from," in *IEEE*, 2017.

[35] L. M. D. L. M. D. Huy D. Huynh, "A New Model for Stock Price Movements Prediction Using Deep Neural Network," in *The Eighth International Symposium*, 2017.