BioGecko

Vol 12 Issue 02 2023
ISSN NO: 2230-5807

Forecasting Stock Market Prices Employing Opinion Miningand Deep Nets with Attention.

Amanpreet Singh[1], prof. (Dr.) R. K. Bathla[2]
[1]Department of Computer Science and Engineering, Desh Bhagat University, India, amanpreet.coe@pbi.ac.in
[2]Department of Computer Science and Engineering, Desh Bhagat University, India, dr.bathla@gmail.com

Abstract: The present volatility of the stock markets makes forecasting stock trends extremely challenging owing to several socio economic and political factors other than market trends. While machine learning models can be used to perform regression analysis based onhistorical numerical data trends, it becomes extremely challenging to incorporate the variabilities which are non-numeric in nature. Some of the factors which govern the rise andfall of stock prices are socio economic conditions, trade wars, current pandemic situation and global market slowdown, reliability of a company among others. Hence, one of the most effective ways to incorporate these trends is analyzing public trends pertaining to the same. While public sentiments may not always be coherent to prevailing market trends, yet they often portray the existential trends in the market and opinions of the public regarding potential purchases of stocks of a particular company in a given time period. This paper presents an approach which is an amalgamation of deep nets with attention and opinion mining for forecasting stock trends. The attention vector employed as an additional input computed on the moving average allows for current trend analysis along with opinion mining from public datasets encompassing both numeric data trends and non-numeric data parameters. The performance of the proposed system has been evaluated in terms of the error rates, regression and accuracy of forecasting for the system. Experimental analysis on benchmark S&P datasets show that the proposed approach outperforms baseline techniquesin terms of accuracy of forecasting and regression.

Keywords: Stock Market Forecasting, Deep Nets, Attention Vector, Opinion Mining, , Regression, Forecasting Accuracy

1. INTRODUCTION

Financial assessment and investing depend critically on stock market trend analysis. While stock market trend analysis is a time series regression problem, what makes it extremely complicated is the dependence on several non-numeric parameters such as socio economic conditions, political conditions of a country, political stability, financial crisis and trade wars, global slowdown and public sentiments pertaining to a company etc [1]. This leads to variabilities in the stock trends often exhibiting non-coherent patterns with respect to historical data [2].
Economic conditions can have a significant impact on stock trends. For example, during a recession, stock prices tend to decline as investors become more risk-averse. Conversely, during periods of economic growth, stock prices tend to rise as investors become more optimistic about the future [3]. Political factors can also impact stock trends. For example, changes in government policies, such as tax rates or regulations, can affect the performance of specific sectors of the economy and, in turn, impact stock prices [4]. The performance of individual companies can also impact stock trends. Positive news, such as strong earnings reports or new product launches, can cause a stock to rise, while negative news, such as a product recall or a decline in earnings, can cause a stock to fall [5]. Market sentiment, or the overall mood of investors, can also impact stocktrends. If investors are optimistic about the future, they may be more willing to invest in stocks, driving up prices [6]. Conversely, if investors are pessimistic, they may be more likely to sell theirstocks, leading to a decline in prices. Global events, such as natural

disasters, geopolitical tensions, or pandemics, can also impact stock trends [7]. These events can create uncertainty in the markets and cause investors to become more risk-averse and hence it is essential to note that stock trends are inherently variable and can be influenced by a wide range of factors [8]. It's also worth mentioning that past performance does not guarantee future results, so investors should exercise caution when making investment decisions based on historical stock trends [9]. The variabilities in stock trends are influenced by a variety of factors, including economic conditions, political factors, company performance, market sentiment, and global events [10]. While several approaches have been developed to predict stock market trends, yet it is often challenging to incorporate the non- numeric global influencing factors as a feature for stock trend analysis. One of the approaches which has been proven to be effective in incorporating global influencing factors along with public sentiments is using opinion mining and sentiment analysis for garnering non-numeric data as an exogenous input in addition to historical numeric data trends [11]. Various trend analysis techniques try to estimate the movement of stock trends based on the variables of influence. Fundamental analysis methods involve studying the financial and economic factors that affect a company's stock price. This includes analyzing the company's financial statements, such as balance sheets, income statements, and cash flow statements [12]. Additionally, fundamental analysts consider macroeconomic factors such as interest rates, inflation, and GDP growth rates. The goal offundamental analysis is to identify companies that are undervalued or overvalued based on their financial and economic metrics [13]. Technical analysis, on the other hand, involves analyzing past market data, such as stock prices and trading volumes, to identify patterns that can be used to predict future stock price movements [14]. Technical analysts use various tools, such as charts and technical indicators, to analyze market data and identify trends and patterns. The goal of technical analysis is to identify trends in the market and use them to predict future price movements [15]. Some noteworthy contribution in the field has been presented subsequently to analyze latest trendsin data. Ren et al. [16] proposed a sentiment analysis based method along with support vector machine (SVM) for stock market forecasting. Polarities of sentiments along with historical data trends have been used to train the SVM regression model. Singh et al. in [17] presented two hybrid models comprising of the SVM-KNN (K-nearest neighbor) and Support Vector Regression (SVR), Artificial Neural Network (ANN) models for stock market trend analysis. The approach shows superior performance compared to the individual models. Gers et al. in [18] proposed a cascaded LSTM model for stock market forecasting, The two LSTM models were shown to have a domino effect with one of the modules predominantly avoiding over fitting and the other predominantly recognizing patterns and forecasting values.

2. DATA PRE-PROCESSING AND FEATURE EXTRACTION.

The objectives of the proposed methodology are addressing two fundamental challenges [19]:

1. Offsetting effects of noise and disturbances inherent to stock.
2. Employing a training algorithm which can render high prediction accuracy for multiple benchmark datasets.
3. Leveraging public sentiments [20].

2.1. Leveraging Sentiment Analysis

The stock markets seem to have a clear reliance on public sentiments. However, public sentiments are often extremely value and random in nature. Quantifying public sentiments is also challenging, in this approach, the opinions of public pertaining to the stock market are gathered from twitter and subsequent quantified as:

+1: Positive Sentiment
-1: Negative Sentiment
0: Neutral Sentiment

One of the major challenges in sentiment analysis is the contextual analysis of data. The different aspects are discussed subsequently [21].

Contextual Analysis

It is often difficult to estimate the context in which the statements are made. Words in textual data such as tweets can be used in different contexts leading to completely divergent meanings [22] Frequency Analysis

Often words in textual data (for example tweets) are repeated such as##I feel so happy today!!

In this case, the repetition of the word is used to emphasize upon the importance of the word. In other words, it increases to its weight. However, such rules are not explicit and do not follow any regular mathematical formulation because of which it is often difficult to get to the actuality of the tweet [23]

Converting textual data into numerically weighted data

The biggest challenge in using an ANN based classifier is the fact that the any ANN structure with a training algorithm doesn't work upon textual data directly to find some pattern. It needs to be fed with numerical substitutes. Hence it becomes mandatory to replace the textual information with numerical information so as to facilitate the learning process of the neural network [24].

the machine or artificial intelligence system requires training for the given categories [25]. Subsequently, the neural network model needs to act as an effective classifier. The major challenges here the fact that sentiment relevant data vary significantly in their parameter values due to the fact that the parameters for each building is different and hence it becomes extremely difficult for the designed neural network to find a relation among such highly fluctuating parameters. Generally, the Artificial Neural Networks model's accuracy depends on the training phase to solve new problems, since the Artificial Neural Networks is an information processing paradigm that learns from its environment to adjust its weights through an iterative process [26].

Deep learning models do have the capability to extract meaning form large and verbose datasets by finding patterns between the inputs and targets. Since neural nets directly process numeric data sets, the processing of data is done prior to training a neural network. The texts are first split into training and testing data samples in the ratio of 70:30 for training and testing. Further, a data vector containing known and commonly repeated spam and ham words is prepared. The SMS spam collection v.1 dataset is used as a dataset for the proposed work. Text normalization is followed by removal of special characters and punctuation marks. Subsequently the data set structuring and preparation is performed based on the feature selection. The proposed approach is mathematically modeled as:

The prepared data vector for training is used for training wherein the weights are initialized randomly. A stepwise implementation is done as:

1. Prepare two arrays, one is input and hidden unit and the second is output unit.

Here, a two dimensional array W_{ij} is used as the weight updating vector and output is a one dimensional array Y_i [27]

3. Original weights are random values put inside the arrays after that the output.

$$x_j = \sum_{i=0} y_i W_{ij} \qquad (1)$$

Where,

y_i is the activity level of the jth unit in the previous layer and

W_{ij} is the weight of the connection between the ith and the jth unit.

4. Next, activation is invoked by the sigmoid function applied to the total weighted input.

$$y_i = \left[\frac{e^x - e^{-x}}{e^x + e^{-x}}\right] \qquad (2)$$

Summing all the output units have been determined, the network calculates the error (E).

$$E = \frac{1}{2} \sum_i (y_i - d_i)^2 \quad (3)$$

Where, y_i is the event level of the j^{th} unit in the top layer and d_i is the preferred output of the j_i unit. Calculation of error for the back propagation algorithm is as follows [28]:

Error Derivative (EA_j) is the modification among the real and desired target:

$$EA_j = \frac{6E}{6y_j} = y_j - d_j \quad (3)$$

Here,
E represents the error
y represents the Target vector
d represents the predicted output
Error Variations is total input received by an output changed given by:

$$E_j = \frac{6E}{6K_j} = \frac{6E}{6y_j} X \frac{dy_j}{dx_j} = EA_j\, y_j (1 - y_i) \quad (4)$$

Here,
E is the error vector
X is the input vector for training the neural network
In Error Fluctuations calculation connection into output unit is computed as:

$$EW_{ij} = \frac{6E}{6W_{ij}} = \frac{6E}{6K_j} \frac{6K_j}{6W_{ij}} = EI_j\, y_i \quad (5)$$

Here,
W represents the weights
I represents the Identity matrix
I and j represent the two dimensional weight vector indicesOverall Influence of the error:

$$E_i = \frac{6E}{6y_i} = \sum_j \frac{6E}{6x_j} X \frac{6x_j}{6y_i} = \sum_j EI_j\, W_{ij} \quad (6)$$

The partial derivative of the Error with respect to the weight represents the error swing for the system while training. The gradient is considered as the objective function to be reduced in each iteration. A probabilistic classification using the Bayes theorem of conditional probability is given by:

$$P\left(\frac{H}{K}\right) = \frac{P\left(\frac{X}{H}\right)P(H)}{P(K)} \quad (7)$$

Here,

Posterior Probability [P (H/X)] is the probability of occurrence of event H when X has already occurred
Prior Probability [P (H)] is the individual probability of event HX is termed as the topple and H is termed as the hypothesis.
Here, [P (H/X)] denotes the probability of occurrence of event X when H has already occurred.

Subsequently, the wavelet transform is used on previous stock values to filter out local noise baseline. The wavelet transform can be thought of as a combination of high pass and low pass filtering techniques.

$$(n) \xrightarrow[DWT]{} Z_{LPF}, Z_{HPF} \quad (8)$$

Here,
Here,
DWTrepresents the discrete wavelet transform operator.
$Z_{LPF}$are the low pass filtered co-efficient values.
$Z_{HPF}$are the high pass filtered co-efficient values. [29].

Typically, the high pass co-efficient values contain the fluctuations and the low pass components contain the original information of the image [30]. The decomposition of the images using wavelet transform can be done as a decomposition tree in which each decomposition level would yield the approximate co-efficient values, the detailed co-efficient values, the horizontal co-efficient values and the vertical co-efficient values [31]. Thus the image in the spatial domain would be converted to the wavelet domain co-efficient as :

$$(x, y) \xrightarrow[DWT2]{} C_A, C_D, C_H, C_V \quad (9)$$

Here,
C_A represents the approximate co-efficient values.
C_D represents the detailed co-efficient values.
C_V represents the vertical co-efficient values.
C_H represents the horizontal co-efficient values.
$DWT2$represents the discrete wavelet transform on two dimensional image data.

3. TRAINING ALGORITHM

The DWT is used to filter the raw data, subsequent to which the back propagation based GDM algorithm is used for pattern recognition and forecasting. The data features used in this study are date, previous day closing price, present day opening price, volume (swing), highest and lowest price of the day [32]. The training algorithm employed here is the back propagation-gradient descent. Following a standard convention, 70% of the data is utilized for training the neural network and 30% is used for testing.
In the proposed training algorithm, a gradient descent with mount factor has been employed to design an ensemble approach [33]. The momentum based approach take in into account the fact that often the condition to attain minima is not reached in optimal number of steps due to the fowlingreasons [34]:

1) Oscillations in the cost function leading to surpassing the convergence plane.
2) Lack of monotonicty in the cost function resulting to non-convergence.

The above constraints are addressed in the momentum based gradient descent where a momentum term plays the role of inertia factor reducing the acceleration along y-axis to impart higher acceleration along an orthogonal axis so as to reach convergence faster. This is depicted in figure 1.

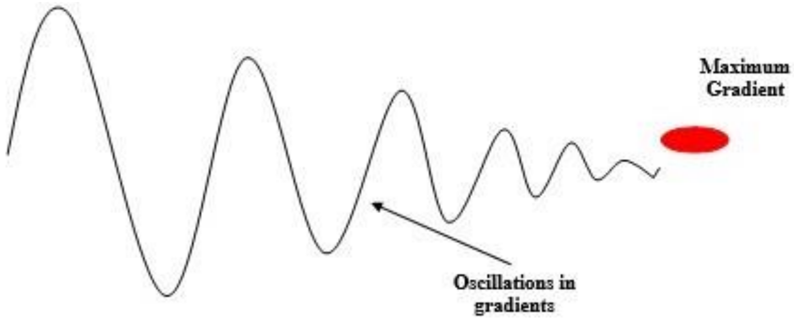

Figure 1. Depicting the oscillations based cost function The acceleration of the cost function can be expressed as:

$$cost_{acceleration} = x_a + x_b \quad (10)$$

Here,
$cost_{acceleration}$ is the overall cost function's acceleration.
x_a corresponds to the acceleration along one direction.
x_b corresponds to acceleration along the orthogonal direction.
By reducing the sudden acceleration along direction 'a', the acceleration along the orthogonal direction can be increased which would result in faster convergence. Thus for each iteration 'k', the weight vector would be updated as a scaled version of the weights as [35]:

$$v_a \partial w = m v_a \partial w + (1 - m) w \quad (11)$$

Here,
v_a represents the learning velocity along 'a'.
m represents the momentum factor
w represents the weights.
∂w represents the differential weights

The training algorithm can be expressed as:.
The training algorithm adopted in this work is given by:
Step.1: Initialize weights (w) randomly.
Step.2: Fix the maximum number of iterations (n)
Step.3: Update weights using gradient descent with an aim to minimize the objective function J given by:

$$J = \frac{1}{m} \sum_{i=1}^{m} (v_i - v_i')^2 \quad (12)$$

Step.4: Compute the Jacobian Matrix $\boldsymbol{J}$ given by:

$$J = \begin{bmatrix} \frac{\delta^2 e_1}{\delta w_1^2} & \cdots & \frac{\delta^2 e_1}{\delta w_m^2} \\ \vdots & \ddots & \vdots \\ \frac{\delta^2 e_n}{\delta w_1^2} & \cdots & \frac{\delta^2 e_n}{\delta w_m^2} \end{bmatrix} \quad (13)$$

Here,
The error e is computed as:

$$e = (v_i - v'_i) \quad (14)$$

Step.5: Iterate steps (1-4) till the cost function J stabilize or the maximum number of iterations setin step 2 is reached, whichever occurs earlier.

Step.6: The weight updating rule for the Bayesian Regularization algorithm is given by:

$$W_{k+1} = W_k - [J_K^T J_k + \mu I]^{-1} J_K^T e_k \quad (15)$$

Here,

I is an identity matrix,

W_k is weight for iteration k,

W_{k+1} is the weight for iteration k+1

e_k is the error for iteration k,

μ is the amount by which weight changes in each iteration

Generally, the gradient is the rate of change of error w.r.t. weights given by:

$$g = \frac{6e}{6w} \quad (16)$$

The second order gradients generally comprise the Jacobian matrix. The simple gradient is actually a function of time or iteration. As data is fed to a neural network for pattern recognition, the weights keep updating. However, it has been found that in case of time series problems, the latest data sample have the maximum impact on the latest output. Hence it is logical to calculate a moving average of latest (previous) data and apply it to the neural network [36]. This is also called a moving average. Mathematically,

$$I_k = X_{1,k}, Mean(X)_{k,k-n}, Y_k \quad (17)$$

Here,

$I_k$is the k[th] input sample to the neural network

$X_{1,k}$ are the data samples from the first to the k[th] sample

$Mea(X)_{k,k-n}$is the mean of the data samples from k-n to k, i.e. it is a moving average dependingon the value of k

$Y_k$is the target

Thus a moving average of the C_A and C_A values can be computed after the after the applicationof the PCA [37]. The next step would be creating a new training vector comprising of the followingvariables:

$$Tr = [X1_{CA,CD}, X2_{CA,CS} \ldots \ldots Xn - 1_{CA.CD} Xn_{CA,CD}, Avg_{n-k}, Y] \quad (18)$$

Here,

Tr is the training vector,

Y is the target vector.

$X1_{CA,CD}, X2_{CA,CS} \ldots \ldots Xn - 1_{CA.CD} Xn_{CA,CD}$ are the individual decomposed values of the features using the DWT iteratively.

Avg_{n-k} is the, owing average of the variables.

Attention Scores: For the ML model's states $\boldsymbol{h_i}$, the regression value targets denoted by the vector $\boldsymbol{s_{t-1}}$, the score value for the regression analysis, $\boldsymbol{e_t}$,, depicts the metric for regression or cosine alignment of the inputs with the targets. The dependence is thus given by [38]:

Vol 12 Issue 02 2023
ISSN NO: 2230-5807

$e_t \longrightarrow Reg\,(h, t)$ (19)
attention score

Weight Updating Rule: The weights $\boldsymbol{\alpha_t}$, can be estimated through the alignment factor for the weights corresponding to the input vectors. The softmax activation function may be used for the purpose given by:

$\alpha_{t,1} = softmax(e_{t,i})$ (20)

Computation of Context: The vector quantityc_t, for the hidden layer connections can becomputed as:

$c_t = \sum_{i=1}^{T} \alpha_{t,i} h_i$ (21)

The contextual vector, $V_{k,i}$, associated with weights $\alpha_{t,i}$ can be computed as:

$attention\ (V_{k,i}, \alpha_{t,i}) = \sum \alpha_{t,i} * V_{k,i}$ (22)

The results obtained through the application of the proposed approach are presents in the subsequent section.

4. RESULTS AND DISCUSSION

Three different datasets have been used in the proposed work which is that of SBI, Infosys and Reliance share prices, obtained from Yahoo Finance Repository (https://in.finance.yahoo.com/quote).The performance indices chosen are the accuracy, mean absolute percentage error, iterations to convergence and regression. The mean absolute percentage error of the system is found to be 7.6% for the SBI dataset. This yields and accuracy of 92.4% which is relatively high compared to the existing literature and more recent hybrid techniques. A similar analysis has been adopted for the Infosys and Reliance datasets.

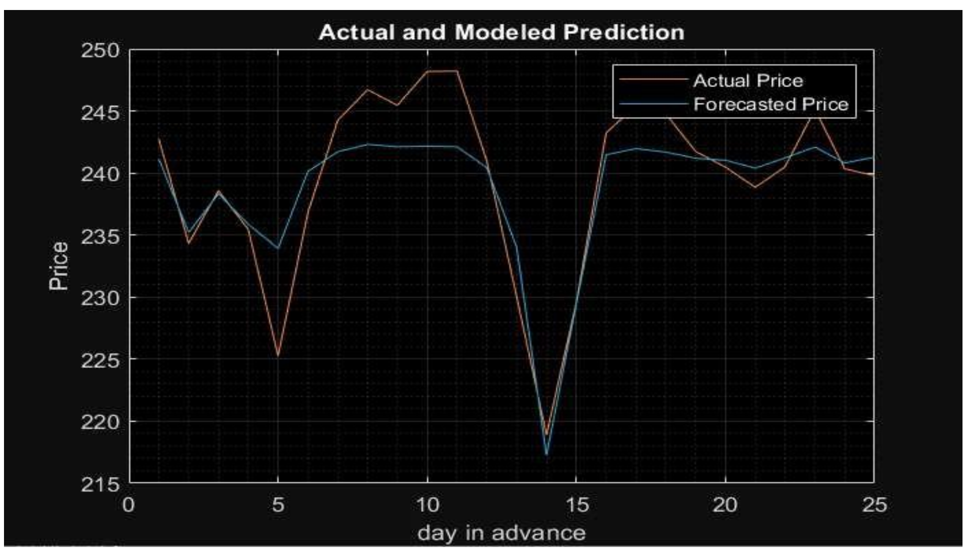

Figure 2. Predicted and Actual values

The MAPE and Regression Values for the Infosys and TCS datasets are 10.55, 0.915, and 11.3, .0.91 respectively. The value of the MAPE and accuracy are suggestive of the fact that the proposed system is capable to filter out the noisy values from the original noise floor and forecast the stock prices with relatively high accuracy. This statistical analysis of the approximate and detailed co-efficient values also indicate the same as the histogram of the detailed coefficient values are significantly w.r.t. to the original data while the detailed co-efficient values are convergent with the actual data. This indicates that the noise effects have been removed by iterative filtering employing the DWT. The regularization parameter avoids the chances of over fitting and facilitates pattern recognition.

Table 1.Summary of MAPE and Regression Values.

S. No.	Dataset	MAPE	Accuracy (%)	Iterations to Convergence	Regression (Overall)
1.	SBI	7.6	92.4	47	0.93
2.	Infosys	10.55	89.45	66	0.93
3.	Reliance	11.3	88.7	59	0.94

A comparative analysis with exiting work in the domain has presented in Table 2.

Table 2. Summary of Comparative Average Accuracy

S. No.	Technique	Accuracy
1.	Transfer Entropy and Machine Learning (Kim et al., 2020)	57%
2.	Augmented Textual Feature Based Learning (Bouktif at al., 2020)	60%
3.	LSTM with Sentiment Analysis (Li et al., 2020)	49.6%
4.	HFS based X-Boost (Pryima et al., 2019)	79%
5.	Hybrid Red Deer-Grey Algorithm (Xu et al., 2020).	85.2%
6.	Variation Auto encoders (VAE) (Liu et al., 2020)	67%
7.	Proposed Technique (Mean Accuracy)	90.183%

5. CONCLUSION

This paper presents a stock market forecasting model based on sentiment analysis and regression learning. The sentiment polarities of the stocks are obtained from twitter data available in the public domain. A DWT decomposition of the data is done and subsequent statistical analysis is performed to correlate the noise floor with the actual data to be analyzed. It has been successfully shown that discarding the detailed co-efficient values helps in data cleaning and retaining the approximate co-efficient values results in subsequent accurate pattern recognition in the data. The performance of the system has bene evaluated in terms of the regression, mean absolute percentage error and accuracy of the system. The system attains an average accuracy of 90.183% which is significantly higher compared to existing literature.

REFERENCES

1. X Zhang, Y Zhang, S Wang, Y Yao, B Fang, "Improving stock market prediction via heterogeneous information fusion", Knowledge-Based Systems, Elsevier 2018, vol. 143, pp.236-247.
2. M Yousfi, YB Zaied, NB Cheikh, BB Lahouel, "Effects of the COVID-19 pandemic on the US stock market and uncertainty: A comparative assessment between the first and second waves", Technological Forecasting and Social Change, Elsevier, 2021, vol. 167, 120710.
3. X. Zhang, S. Qu, J. Huang, B. Fang and P. Yu, "Stock Market Prediction via Multi-Source Multiple Instance Learning," in IEEE Access 2018, vol. 6, pp. 50720-50728.
4. M. Nabipour, P. Nayyeri, H. Jabani, S. S. and A. Mosavi, "Predicting Stock Market Trends Using Machine Learning and Deep Learning Algorithms Via Continuous and Binary Data; a Comparative Analysis," in IEEE Access 2020, vol. 8, pp. 150199-150212
5. S. M. Idrees, M. A. Alam and P. Agarwal, "A Prediction Approach for Stock Market Volatility Based on Time Series Data," in IEEE Access 2019, vol. 7, pp. 17287-17298
6. S Sohangir, D Wang, A Pomeranets, "Big Data: Deep Learning for financial sentiment analysis", Journal of Big Data, Springer 2018, vol. 5, Article number. 3.
7. M Sadiq, CC Hsu, YQ Zhang, F Chien, "COVID-19 fear and volatility index movements: empirical insights from ASEAN stock markets", Environmental Sciences ad Pollution Research, Springer 2021, vol. 28, pp.67167–67184.
8. M Balcilar, M Bonato, R Demirer, R Gupta, "Geopolitical risks and stock market dynamics of the BRICS", Economic Systems, Elsevier 2018, vol.42, no.2, pp. 295-306.
9. D. Shah, H. Isah and F. Zulkernine, "Predicting the Effects of News Sentiments on the Stock Market," 2018 IEEE International Conference on Big Data (Big Data), Seattle, WA, USA, 2018, pp. 4705-4708.
10. M. S. Hegde, G. Krishna and R. Srinath, "An Ensemble Stock Predictor and Recommender System," 2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Bangalore, India, 2018, pp. 1981-1985.
11. Y Cui, Y Jiang, H Gu, "Novel Sentiment Analysis from Twitter for Stock Change Prediction", International Conference on Data Mining and Big Data (DMBD), Springer 2022, pp 160–172.
12. MM Kumbure, C Lohrmann, P Luukka, "Machine learning techniques and data for stock market forecasting: A literature review", Expert Systems with Applications, Elsevier 2022, vol. 197, 116659.
13. X Li, P Wu, "Stock price prediction incorporating market style clustering", Cognitive Computation, Springer 2022, vol. 14, pp.149–166.
14. G. Li, A. Zhang, Q. Zhang, D. Wu and C. Zhan, "Pearson Correlation Coefficient-Based Performance Enhancement of Broad Learning System for Stock Price Prediction," in IEEE Transactions on Circuits and Systems II: Express Briefs, 2022, vol. 69, no. 5, pp. 2413-2417.
15. G. Duan, M. Lin, H. Wang and Z. Xu, "Deep Neural Networks for Stock Price Prediction," 2022 14th International Conference on Computer Research and Development (ICCRD), Shenzhen, China, 2022, pp. 65-68.
16. R. Ren, D. D. Wu and T. Liu, "Forecasting Stock Market Movement Direction Using Sentiment Analysis and Support Vector Machine," in IEEE Systems Journal, vol. 13, no. 1, pp. 760-770.

17. S. Singh, T. K. Madan, J. Kumar and A. K. Singh, "Stock Market Forecasting using Machine Learning: Today and Tomorrow," 2019 2nd International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT), Kannur, India, 2019, pp. 738-745.
18. Gers F, Eck D, Schmidhuber J.(2020) Applying LSTM to time series predictable through time-window approaches, Neural Nets WIRN Vietri-01. Perspectives in Neural Computing. Springer, 193-200.
19. Gu Q, Chang Y, Xiong N, Chen. (2021), Forecasting Nickel futures price based on the empirical wavelet transform and gradient boosting decision trees, Applied Soft Computing, Elsevier, 109, 107472.
20. Li X, Wu P, Wang W. (2020) Incorporating stock prices and news sentiments for stock market prediction: A case of Hong Kong", Information Processing & Management, Elsevier 57(5), 1-19.
21. Guo Y, Han S, Shen C, Li Y, Yin X, Bai Y. (2018), An adaptive SVR for high-frequency stock price forecasting, IEEE Access, 6, 11397 – 11404.
22. Kim S, Ku S, Chang W, Song J.W. (2020) Predicting the Direction of US Stock Prices Using Effective Transfer Entropy and Machine Learning Techniques, IEEE Access 2020, 8, 111660 – 111682.
23. Lincy G, Joh Cn (2016). A multiple fuzzy inference systems framework for daily stock trading with application to NASDAQ stock exchange, Journal of Expert Systems with Applications, 44, 1-18.
24. Liu H, Long Z (2020). An improved deep learning model for predicting stock market price time series, Digital Signal Processing, Elsevier, 102, 102741.
25. Liu W, Chen L, Chen Y, Zhang W (2020), Accelerating Federated Learning via Momentum Gradient Descent, IEEE Transactions on Parallel and Distributed Systems, 31(8), 1754-1766.
26. Luo X, Qin W, Dong A, Sedraoui K, Zhou M (2021). Efficient and High-quality Recommendations via Momentum-incorporated Parallel Stochastic Gradient Descent-Based Learning, IEEE/CAA Journal of Automatica Sinica, 8(2), 402-411.
27. Majumdar S, Laha A. (2020), Clustering and classification of time series using topological data analysis with applications to finance, Expert Systems with Applications, Elsevier, 162, 113868.
28. Moghar A, Hamiche M. (2020), Stock market prediction using LSTM recurrent neural network, Procedia in Computer Science, Elsevier, 170, 1168-1173.
29. Nabipour M, Nayyeri P, Jabani H, Mosavi A. (2020) Predicting Stock Market Trends Using Machine Learning and Deep Learning Algorithms Via Continuous and Binary Data; a Comparative Analysis, IEEE Access, 8, 150199-150212.
30. Naeini M, Taremian H, Hashemi H (2010), Stock market value prediction using neural networks, 2010 International Conference on Computer Information Systems and Industrial Management Applications (CISIM), IEEE, 132-13.
31. Pryima S, Vovk R, Vovk V (2019) Using Artificial Neural Networks to Forecast Stock Market Indices, 2019 XIth International Scientific and Practical Conference on Electronics and Information Technologies (ELIT), 108-112.
32. Quilty J, Adamowski J.(2017) Addressing the incorrect usage of wavelet-based hydrological and water resources forecasting models for real-world applications with best practices and a new forecasting framework, Journal of Hydrology, Elsevier, 563, 336-353.
33. Raimundo M, Okamoto J (2018). SVR-wavelet adaptive model for forecasting financial time series, 2018 International Conference on Information and Computer Technologies (ICICT), 111-114.

34. Selvin S, Vinayakumar R, Gopalakrishnan E, Menon V, Soman K (2017), Stock price prediction using LSTM, RNN and CNN-sliding window model, 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI), 1643- 1647.
35. Shah D, Isah H, Zulkernine F. (2019), Stock market analysis: A review and taxonomy of prediction techniques, International Journal of Financial Studies, MDPI, 7(2), 1-26.
36. Vargas M, dos Anjos C, Bichara G, Evsukoff A (2018), Deep Leaming for Stock Market Prediction Using Technical Indicators and Financial News Articles, 2018 International Joint Conference on Neural Networks (IJCNN), 1-8.
37. F. Nie, Z. Wang, R. Wang and X. Li, "Submanifold-Preserving Discriminant Analysis With anAuto-Optimized Graph," in IEEE Transactions on Cybernetics. 2020, vol. 50, no. 8, pp. 3682-3695.
38. D Soydaner,” Attention mechanism in neural networks: where it comes and where it goes”, Neural Computing and Applications, Springer 2022, vol.34, pp. 13371–13385.