Estimating the Movement of Stock Prices Based Using Artificial Neural Networks

Swarna Gupta[1], Prof. Sachin Mahajan[2]
[1, 2] JIT, Borawan

Abstract- *Predicting stock market movements using neural networks is a popular application of machine learning. Neural networks can analyze historical stock data and attempt to identify patterns and trends that can be used to predict future price movements. While neural networks can be effective tools for stock market prediction, it's important to note that the stock market is highly complex and influenced by a multitude of factors, including economic indicators, geopolitical events, investor sentiment, and more. Therefore, predicting stock market movements with high accuracy is a challenging task, and no method can guarantee consistent success. This paper presents an artificial neural network based method for stock market prediction and it has been shown that the proposed approach attains higher accuracy compared to existing approaches.[1].*

Keywords- Stock Market Prediction, Artiticial Neural Network, Mean Absolute Percentage Error (MAPE), Accuracy.

I. INTRODUCTION

While stock market prediction models can be valuable tools, it's important to acknowledge their limitations. Financial markets are complex and subject to a wide range of unpredictable factors. Models are based on historical data and assumptions, and unexpected events or market shifts can challenge their accuracy. Therefore, stock market predictions should be used as one source of information among many, and investors should exercise caution and diversify their strategies. The main applications of stock market prediction are:

Investment Decision Making: Investors and traders use stock market prediction models to make informed investment decisions. By predicting stock price movements, these models can help identify potential buying or selling opportunities, optimize portfolio allocation, and manage risk. Risk Management: Stock market prediction models assist in risk management by providing insights into potential market volatility and downside risks. By anticipating market downturns or identifying periods of increased volatility, investors can take appropriate measures to protect their portfolios. Algorithmic Trading: With the rise of algorithmic trading, stock market prediction models are employed to automate trading decisions. These models use real-time market data and predictive analytics to execute trades automatically based on predefined rules, optimizing trading strategies and minimizing human biases. Financial Planning: Stock market prediction models can be used in financial planning to estimate future investment returns and assess the viability of long-term financial goals. They help individuals and financial advisors evaluate different investment scenarios and make informed decisions about asset allocation and retirement planning. Market Analysis and Research: Researchers, analysts, and financial institutions employ stock market prediction models to gain insights into market dynamics and behavior. These models can uncover hidden patterns, detect market inefficiencies, and contribute to the development of new investment strategies and theories. Economic Forecasting: Stock market performance is often considered a barometer of economic health. Stock market prediction models can aid in economic forecasting by providing insights into the overall direction of the market and its potential impact on the broader economy. Risk Assessment and Regulation: Stock market prediction models are used by regulatory bodies and risk management departments of financial institutions to assess market risks and implement appropriate regulations and safeguards. These models help monitor market stability, detect abnormal trading patterns, and prevent fraudulent activities.

The design of machine learning based approaches are fundamental in stock market prediction.

II. MACHINE LEARING FOR STOCK MARKET PREDICITON

There are several approaches for stock market prediction using machine learning. The scientific report provides a thorough examination of machine learning approaches for stock market prediction. It covers various techniques, including neural networks, support vector machines, random forests, and ensemble methods, and discusses their applications, strengths, and limitations. The report also There are several approaches for stock market prediction using machine learning. The scientific report provides a thorough examination of machine learning approaches for stock market prediction. It covers various techniques, including neural networks, support vector machines, random forests, and ensemble methods, and discusses their applications, strengths, and limitations. The report also addresses important aspects of data preprocessing, feature selection, and evaluation metrics specific to stock market prediction. Furthermore, it explores the challenges and limitations associated with the use of machine learning models in this domain and highlights recent advancements and future directions for research. Overall, this report serves as a comprehensive resource for researchers, practitioners, and investors interested in leveraging machine learning for stock market prediction. There are several machine learning approaches commonly used for stock market prediction. Here are some of the prominent techniques:

Neural Networks:

Feedforward Neural Networks: These networks consist of an input layer, hidden layers, and an output layer. They are trained using backpropagation and can capture complex relationships in stock market data.

Recurrent Neural Networks (RNNs): RNNs are effective for sequential data analysis, making them suitable for time series prediction tasks. They have memory cells that retain information about past inputs, allowing them to capture temporal dependencies.

Long Short-Term Memory (LSTM) Networks: LSTMs are a type of RNN that address the vanishing gradient problem and can learn long-term dependencies. They have proven successful in capturing patterns in financial time series data.
Support Vector Machines (SVM):

SVMs are widely used for classification and regression tasks. They map input data into a higher-dimensional feature space and find an optimal hyperplane to separate different classes or predict continuous values.
Random Forests:

Random Forests are ensemble learning methods that combine multiple decision trees. They generate predictions based on the majority vote or average of predictions from individual trees. Random Forests can handle high-dimensional data, capture nonlinear relationships, and reduce overfitting.

Gradient Boosting Methods:

Gradient Boosting methods such as Gradient Boosting Machines (GBM), XGBoost, and LightGBM create an ensemble of weak prediction models and iteratively optimize them to minimize the prediction error. These algorithms are known for their excellent performance in various machine learning tasks, including stock market prediction.

Deep Learning Architectures:

Convolutional Neural Networks (CNNs) are primarily used for image analysis but can also be applied to financial data by treating it as an image-like input.

Generative Adversarial Networks (GANs) can generate synthetic financial data, which can be used for training or augmenting the available dataset for stock market prediction.
Reinforcement Learning:

Reinforcement Learning (RL) algorithms, such as Q-Learning and Deep Q-Networks (DQN), can learn optimal trading policies by interacting with the market environment. These approaches can capture complex trading strategies and adapt to changing market conditions.

It's important to note that the effectiveness of these approaches can vary depending on the quality and characteristics of the data, feature engineering techniques, model architecture, and the specific characteristics of the stock market being analyzed. The selection of an appropriate machine learning approach should consider these factors and be validated through rigorous testing and evaluation.

III. PROPOSED APPROACH

The proposed approach uses a neural network based approach for stock market prediction

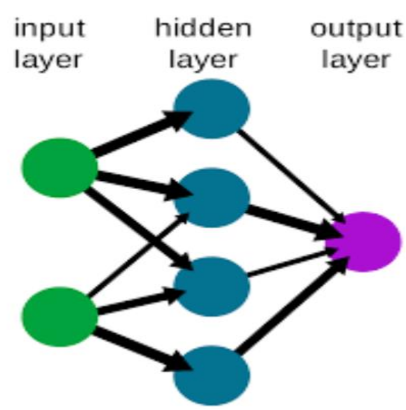

Fig.1 The Neural Network Architecture

The figure above depicts the deep neural network architecture with multiple hidden layers. The output of the neural network however follows the following ANN rule:

$$Y = \sum_{i=1}^{n} X_i \cdot W_i + \theta_i \qquad (2)$$

Where,

X are the inputs
Y is the output
W are the weights
Θ is the bias.
Training of ANN is of major importance before it can be used to predict the outcome of the data inputs.

Back propagation is one of the most effective ways to implement the deep neural networks with the following conditions:

1) Time series behavior of the data
2) Multi-variate data sets
3) Highly uncorrelated nature of input vectors

We proposed an error feedback mechanism generally is well suited to time series problems in which the dependent variable is primarily a function of time along with associated variables. Mathematically,

$$Y = f(t, V_1 \ldots . V_n) \quad (2)$$

Here,
Y is the dependent variable
f stands for a function of
t is the time metric
V are the associated variables
n is the number of variables

In case of back propagation, the weights of a subsequent iteration doesn't only depend on the conditions of that iteration but also on the weights and errors of the previous iteration mathematically given by:

$$W_{k+1} = f(W_k, e_k, V) \quad (3)$$

Here,
W_{k+1} are the weights of a subsequent iteration
W_k are the weights of the present iteration
e_k is the present iteration error
V is the set of associated variables

Mathematically, let the gradient be represented by g and the descent search vector by p, then

$$p_0 = -g_0 \quad (4)$$

Where,
g_0 denotes the gradient given by $\frac{\partial e}{\partial w}$
The sub-script 0 represents the starting iteration
The negative sign indicates a reduction in the errors w.r.t. weights
The tradeoff between the speed and accuracy is clearly given by the following relations:

$$W_{k+1} = W_k - \alpha g_x, \quad \alpha = \frac{1}{\mu} \quad (5)$$

Here,
w_{k+1} is the weight of the next iteration
w_k is the weight of the present iteration
g_x is the gradient vector
μ is the step size for weight adjustment in each iteration.

The wavelet transform is used to filter the data. The wavelet transform is an effective tool for removal of local disturbances. Stock prices show extremely random behavior and local disturbances. The major advantage of the wavelet transform is the fact that it is capable of handling fluctuating natured data and also local disturbances. The DWT can be defined as:

$$W\Phi\,(Jo, k) = \frac{1}{\sqrt{M}} \sum_n S(n).\Phi(n)_{jo',k} \quad (10)$$

The entire methodology can be understood using the system flowchart depicted below.

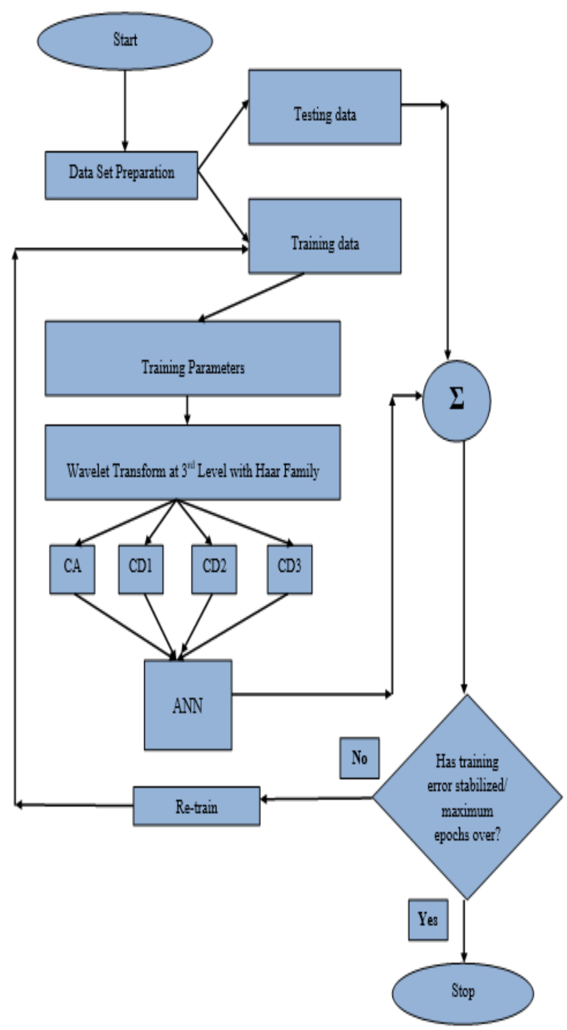

Fig.2 Flowchart of Proposed System

The data is divided in the ration of 70:30 for training and testing data set bifurcation.

The final performance metrics computed for system evaluation are:

1) Mean Absolute Percentage Error (MAPE)

$$MAPE = \frac{100}{M}\sum_{t=1}^{N}\frac{E-E_t|}{E_t} \qquad \textbf{(11)}$$

Here E_t and $E_t^{\sim}$ stand for the predicted and actual values respectively.

The number of predicted samples is indicated by M.

2) Regression

The extent of similarity between two variables is given by the regression where the maximum value is 1 and the minimum is 0.

IV. SYSTEM MODEL AND RESULTS

The proposed work is implemented on matlab for the following reasons:

MATLAB is a powerful programming language and software environment that can be utilized for stock market prediction. Here's a general outline of how you can use MATLAB for this purpose:

Data Collection and Preprocessing:

Use MATLAB's data import capabilities to retrieve historical stock market data from various sources or APIs.

Perform data preprocessing tasks such as handling missing values, removing outliers, and normalizing or scaling the data to ensure consistency.

Feature Selection and Engineering:

Utilize MATLAB's data manipulation and analysis functions to select relevant features for stock market prediction, such as technical indicators (moving averages, oscillators, etc.) and fundamental data (earnings per share, price-to-earnings ratio, etc.).

Create custom functions or scripts to engineer new features that capture specific patterns or relationships in the data.

Model Development and Training:

Implement machine learning algorithms or build custom predictive models using MATLAB's built-in functions and libraries, such as Neural Network Toolbox, Statistics and Machine Learning Toolbox, or Econometrics Toolbox.

Define the appropriate model architecture, set hyperparameters, and split the data into training and testing sets.

Train the model using the training data and evaluate its performance using appropriate metrics.

Model Evaluation and Optimization:

Use MATLAB's statistical analysis functions to assess the performance of the trained model, such as computing accuracy, precision, recall, or mean squared error.

Perform model optimization by tuning hyperparameters, exploring different algorithms, or employing ensemble methods to improve prediction accuracy.

Prediction and Visualization:

Use the trained model to make predictions on new, unseen data.

Visualize the predicted stock market movements and compare them with the actual prices using MATLAB's plotting and visualization capabilities.

Analyze the results and iterate on the model or feature selection process as needed.

The results have been evaluated based on the following parameters:

1. (MAPE)
2. Regression
3. MSE w.r.t. the number of epochs

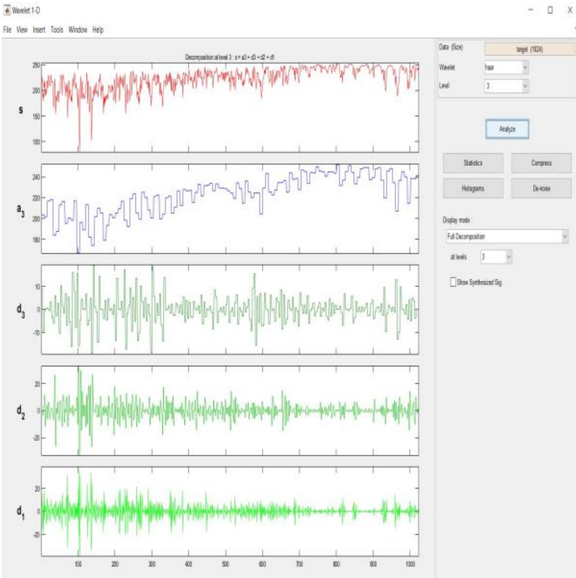

Fig.3 Decomposition of Data using DWT

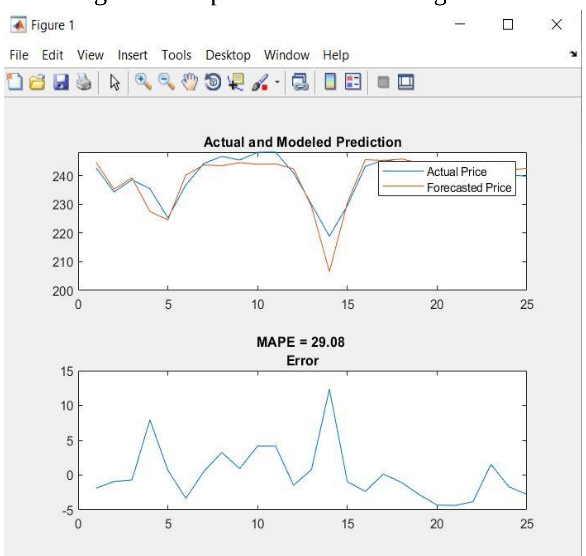

Fig.4 Predicted and Actual Values

The figure above depicts the predicted and actual stock behavior.

From the above figures, it can be concluded that the proposed system attains the following results:
1) MAPE of Proposed work=29.08%
2) Accuracy of Proposed work=70.92
3) Accuracy of Previous work [1]=57%

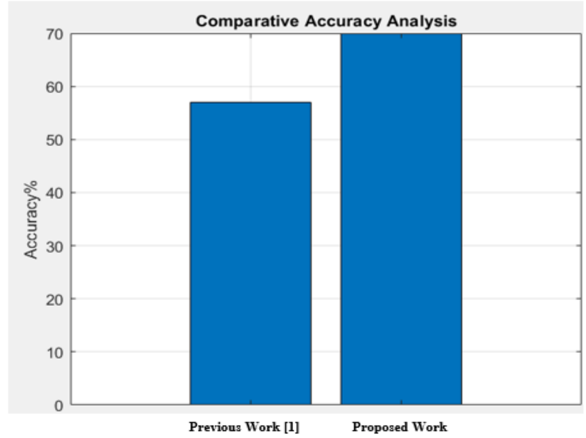

Fig.5 Comparative Accuracy Analysis w.r.t. Previous Work [1]

The proposed approach attains higher accuracy compared to existing work.

V. CONCLUSION

Based on the preceding discussions, stock market prediction is a challenging task due to its time series nature and sensitivity to external factors. Achieving high levels of accuracy in prediction is often difficult. In this proposed approach, a deep learning model based on backpropagation is suggested, employing a 1-10-1 configuration. The neural network is trained using the adaptive gradient descent algorithm (GDA). Data preprocessing is performed using the discrete wavelet transform. The results demonstrate that the proposed approach achieves a mean absolute percentage error of 29.08%, outperforming previous work with a mean absolute percentage error of 57% [1]. Additionally, the regression value at 27 epochs is 0.93, indicating low errors, high accuracy, and a relatively small number of iterations. Moreover, the proposed system exhibits superior accuracy compared to existing systems when evaluated on benchmark datasets.

REFERENCES

[1] Xiongwen Pang, Yanqiang Zhou, Pan Wang, Weiwei Lin, Victor Chang., “An innovative neural network approach for stock market prediction”, Springer 2018.
[2] John M. Kirimi, Christopher A. Moturi, “Application of Data Mining Classification in Employee Performance Prediction”, International Journal of Computer Applications (0975 – 8887) Volume 146 – No.7, July 2016
[3] Thomas Fischer, Christopher Krauss, “Deep Learning with long short term memory networks for financial market predictions”, Elsevier 2017

[4] Daša Dragnić, "Impact of Internal and External Factors on The Performance of Fast-Growing Small And Medium Business", Preliminary communication, Accepted: 26. 05. 2014 UDC: 005.71-022.51/.55(497.5)
[5] Amin Hedayati Moghaddam, Moein Hedayati Moghaddam, Morteza Esfandyari, "Stock market index prediction using artificial neural network", Journal of Economics, Finance and Administrative Science 21 (2016) 89–93,
[6] Aditya Bhardwaj, Yogendra Narayan, Vanraj, Pawan, Maitreyee Dutta, "Sentiment Analysis for Indian Stock Market Prediction Using Sensex and Nifty", Procedia Computer Science 70 (2015) 85 – 91
[7] Mehak Usmani, Syed Hasan Adil, Kamran Raza, Syed Saad Azhar Ali, "Stock Market Prediction using Machine Learning Techniques", DOI: 10.1109/ICCOINS.2016.7783235
[8] Reza Hafezi, Jamal Shahrabi, Esmaeil Hadavandi, "A bat-neural network multi-agent system (BNNMAS) for stock price prediction: Case study of DAX stock price", Applied Soft Computing 29 (2015) 196–210
[9] Xiongwen Pang, Yanqiang Zhou, Pan Wang, Weiwei Lin, Victor Chang, "An innovative neural network approach for stock market prediction", Springer Science+Business Media, LLC, part of Springer Nature 2018
[10] Lili Wang, Zitian Wang, Shuai Zhao, Shaohua Tan, "Stock market trend prediction using dynamical Bayesian factor graph", Expert Systems with Applications, No. of Pages 9, 24 March 2015
[11] Adam Atkins, Mahesan Niranjan, Enrico Gerding, "Financial news predicts stock market volatility better than close price", The Journal of Finance and Data Science 4 (2018) 120-137
[12] Aparna Nayak, M. M. Manohara Pai and Radhika M. Pai, "Prediction Models for Indian Stock Market", Procedia Computer Science 89 (2016) 441 – 449
[13] Xingyu Zhou, Zhisong Pan, Guyu Hu, Siqi Tang, and Cheng Zhao, "Stock Market Prediction on High-Frequency Data Using Generative Adversarial Nets", Hindawi Mathematical Problems in Engineering Volume 2018, Article ID 4907423, 11 pages
[14] Sahar Sohangir, Dingding Wang, Anna Pomeranets and Taghi M. Khoshgoftaar, "Big Data: Deep Learning for financial sentiment analysis", J Big Data (2018) 5:3