Computing and Informatics, Vol. 33, 2014, 992–1024

A TECHNIQUE TO STOCK MARKET PREDICTION USING FUZZY CLUSTERING AND ARTIFICIAL NEURAL NETWORKS

Rajendran Sugumar

Department of Computer Science and Engineering
Velammal Institute of Technology
Chennai – Kolkatta High Way, Panchetti, Thiruvallur District
Chennai-601204, India
e-mail: dr.sugumar16@gmail.com

Alwar Rengarajan

Department of Computer Science and Engineering
Veltech Multitech SRS Engineering College
Chennai, India

Chinnappan Jayakumar

Department of Computer Science and Engineering
RMK Engineering College
Chennai, India

Abstract. Stock market prediction is essential and of great interest because successful prediction of stock prices may promise smart benefits. These tasks are highly complicated and very difficult. Many researchers have made valiant attempts in data mining to devise an efficient system for stock market movement analysis. In this paper, we have developed an efficient approach to stock market prediction by employing fuzzy C-means clustering and artificial neural network. This research has been encouraged by the need of predicting the stock market to facilitate the investors about buy and hold strategy and to make profit. Firstly, the original stock market data are converted into interpreted historical (financial) data i.e. via technical indicators. Based on these technical indicators, datasets that are required for analysis are created. Subsequently, fuzzy-clustering technique is used to generate different training subsets. Subsequently, based on different training subsets, different ANN models are trained to formulate different base models. Finally, a meta-learner, fuzzy system module, is employed to predict the stock price. The results for the stock market prediction are validated through evaluation metrics, namely mean absolute deviation, mean square error, root mean square error, mean absolute percentage error used to estimate the forecasting accuracy in the stock market. Comparative analysis is carried out for single Neural Network (NN) and existing technique neural. The obtained results show that the proposed approach produces better results than the other techniques in terms of accuracy.

Keywords: Stock market prediction, Rate of Change (ROC), Money Flow Index (MFI), Relative Strength Index (RSI), stochRSI, ultimate oscillator, MSE RMSE, MAPE, MAD

1 INTRODUCTION

From researchers and investors, financial prediction and trading presents a demanding task that attracts great interest because success may result in substantial rewards. On the other hand, since the financial market is a highly complex and dynamic system, predicting the financial market is not an easy task which involves the actual actions taken by the millions of investors and institutions [1]. By the predictability of the financial market, many investors are persuaded and they try to make profit through exploiting the analysis of financial data [1, 26]. Predicting current concern of factories and manufacturing companies is the desire of investors, auditors, financial analysts, governmental officials, employees and managers [11]. More prominently, within stock market research, it is believed that the information from periodical reports and annual reports can manipulate the price of a stock, especially for unexpected earnings or unexpected loss surprises [8]. The complexity of the task has raised questions on whether the stock market price can be predicted [9]. However, stock market investors believe stock prices can be expected, and profit can be made through exploiting assorted technical or fundamental analysis, in addition to momentum strategies (buy when market is bullish, sell when market is bearish) [9].

Stock prediction is one of the most challenging problems due to difficulty and uncertainty of stock market [6]. Obviously, stock markets are complex, nonlinear, and dynamic [7]. Therefore, stock market prediction is a dangerous venture. For researchers, stock market prediction has always had a certain appeal. No method has been discovered to accurately predict stock price movement, while various scientific attempts have been made. The complexity of prediction lies in the complexities of modeling human behavior [5]. By buying or selling their investments at an appropriate time, investors in the market want to make the most of their returns.

As stock market data are highly time-variant and are usually in a nonlinear pattern, predicting the future trend (i.e., rise, decrease, or remain steady) of a stock is a challenging problem [2]. Analysis and prediction of the stock market behavior have been accompanied by predictions of the behavior of the prices. To put forward future behavior some of the approaches rely on charts of the prices, volumes, and visual human analysis of these diagrammatic representations. Others manipulate the historical values of the time series to calculate technical indicators. The value, or values, of one or more of these are used to suggest good times for buying or selling stock. Both Chartist techniques and the use of indicators are technical models that use only information gained through the trading history of a stock. On the contrary, a basic model looks at the past financial performance of a company, the behavior of the economy as a whole, and the industry to which a company belongs. In predicting the future performance, some also use knowledge of the past performance of the directors. Other models mix both technical and fundamental aspects [10, 25]. The indecisive nature of the stock market requires the use of data mining techniques like clustering for stock market analysis and prediction [20].

Uncovering market trends, planning investment strategies, identifying the best time to purchase the stocks and what stocks to purchase are included in stock market. For approaching these enormously complex and dynamic problems with data mining tools, financial institutions produce huge data sets that build a foundation [3, 4]. To predict stock market along with the development of artificial intelligence, especially machine learning and data mining, ever more researchers try to build automatic decision-making systems [6]. Among these approaches, soft computing techniques such as fuzzy logic, neural networks, and probabilistic reasoning (which includes genetic algorithms, chaos theory, etc.) draw most consideration due to their abilities to handle ambiguity and noise in stock market [12, 6]. In both trend analysis and forecasting stock data mining has given encouraging results by using these certain artificial intelligence techniques [21, 27]. In addition, to accomplish better prediction results for forecasting incorporating emerging Artificial Intelligence techniques such as neural networks and/or fuzzy logic with the data mining methods are of wide interest.

In this article, we devise an efficient approach to stock market prediction by employing fuzzy C-means clustering and artificial neural network. Here, we have based our work on the interpreted historical (financial) data, i.e. via technical indicators. Based on these technical indicators, datasets that are required for analysis are created. By using fuzzy clustering technique, the whole training set can be divided into subsets which have less size and lower complexity. Therefore, based on these subsets, the stability of individual ANN can be improved, and the detection precision, especially for low-frequent attacks, can also be enhanced.

The rest of the paper is organized as follows: A brief review of researches related to the proposed technique is presented in Section 2. Section 3 describes contribution of the research paper. The technical indicator description is presented in Section 4. The proposed technique is presented in Section 5. The detailed experimental results and discussions are given in Section 6. The conclusions are summed up in Section 7.

2 REVIEW OF RELATED RESEARCH

Prediction of stock price variation is the most difficult task and the price movement behaves more like a random walk and time varying. Now, researchers have started using different types of AI techniques to make trading decisions. Here, we present a brief review of some of the significant researches. Sheta [13] has used Takagi-Sugeno (TS) technique. This technique is used to develop fuzzy models for two nonlinear processes. For a NASA software project and the prediction of the next week S & P 500 for stock market, they were used for the software effort estimation. Determination of the membership functions in the rule antecedents using the model input data and the estimation of the consequence parameters are the two steps through which the development of the TS fuzzy model can be achieved. To estimate these parameters they used least-square estimation.

Su et al. [14] have developed a self-organized, five-layer neuro-fuzzy model. This model is designed to model the dynamics of stock market by using technical indicators. In prediction and forecasting, the model effectiveness was validated by a set of data containing four indicators and they are the stochastic oscillator (% K and % D), volume adjusted moving average (VAMA) and ease of movement (EMV) from TAIEX (Taiwan Stock Exchange Capitalization Weighted Stock Index). To predict the input set for the neuro-fuzzy model in forecasting stock price, a modified moving average method can be proposed. The model was effective in prediction and accurate in forecasting which was proved from the simulation results. From the prediction of the modified moving average method, the input error attenuated significantly by the neuro-fuzzy model to yield better forecasting results.

Fazel Zarandi et al. [15] have developed a type-2 fuzzy rule based expert system. For stock price analysis the proposed system was developed. Interval type-2 fuzzy logic system permitted to model rule uncertainties and every membership value of an element was interval itself. As the input variables, the proposed type-2 fuzzy model applied the technical and fundamental indexes. In Asia the model can be tested on stock price prediction of an automotive manufactory. The model had successfully forecasted the price variation for stocks from different sectors, through the intensive experimental tests. During the trading period, the results were very encouraging and implemented in a real-time trading system for stock price prediction.

Lai et al. [16] have established a financial time series-forecasting model. The established model was designed by evolving and clustering fuzzy decision tree for stocks in Taiwan Stock Exchange Corporation (TSEC). To construct a decision-making system based on historical data and technical indexes, the forecasting model integrated a data clustering technique, a fuzzy decision tree (FDT), and genetic algorithms (GA). By adopting K-means algorithm the set of historical data can be divided into k sub-clusters. For each input index in FDT, GA was then applied to evolve the number of fuzzy terms so the forecasting accuracy of the model can be further improved. A different forecasting model was generated for each sub-cluster. In other words, in each sub-cluster the number of fuzzy terms was different. When compared with other approaches on various stocks in TSEC, hit rate applied as a performance measure and the proposed GAFDT model had the best performance.

Shyi-Ming Chen and Yu-Chuan Chang [17] have presented a method for multivariable fuzzy forecasting. Based on fuzzy clustering and fuzzy rule interpolation techniques the proposed method was developed. Initially, the proposed method constructed training samples to construct fuzzy rules based on the variation rates of the training data set and then used the training samples by making use of the fuzzy C-means clustering algorithm, where each fuzzy rule corresponded to a given cluster. Then, with respect to the input observations, they determined the weight of each fuzzy rule and to determine the predicted output such weights were used, based on the multiple fuzzy rules interpolation scheme. For the temperature prediction problem and the Taiwan Stock Exchange Capitalization Weighted Stock Index (TAIEX) data the proposed method was applied. The experimental results have shown that the proposed method can produce better forecasting results, than several existing methods.

Ho et al. [18] have proposed an interday financial trading system. This system is proposed with a predictive model empowered by a novel brain-inspired evolving Mamdani-Takagi-Sugeno Neural-Fuzzy Inference System (eMTSFIS). As compared to existing econometric and neural-fuzzy techniques, the eMTSFIS predictive model possessed synaptic mechanisms and information processing capabilities of the human hippocampus resulted in a more robust and adaptive forecasting model. Based on the moving-averages-convergence/divergence (MACD) principle, the trading strategy of the proposed system was developed to generate buy-sell trading signals. The lagging nature of the MACD trading rule can be addressed by introducing forecasting capabilities to the computation of the MACD trend signals. Experimental results based on the S & P 500 Index established that eMTSFIS was able to present highly accurate predictions and the resultant system was able to recognize timely trading opportunities while avoiding unnecessary trading transactions. To yield higher multiplicative returns for an investor, these attributes enabled the eMTSFIS-based trading system.

Hadavandi et al. [19] have presented an integrated approach. This approach is based on genetic fuzzy systems (GFS) and artificial neural networks (ANN) for constructing a stock price forecasting expert system. Initially, to determine factors which have most influence on stock prices, they used stepwise regression analysis (SRA). At the next stage, by means of self-organizing map (SOM) neural networks, they divided their raw data into k clusters. Finally, with the ability of rule base extraction and data base tuning, all clusters were fed into independent GFS models. By applying it on stock price data gathered from IT and airlines sectors, and compared to the outcomes with previous stock price forecasting methods using mean absolute percentage error (MAPE), they evaluated the capability of the proposed approach. The results have shown that the proposed approach outperformed all previous methods, so it can be considered as a suitable tool for stock price forecasting problems.

3 CONTRIBUTION OF THE RESEARCH

The main contributions of our proposed technique are as follows:

- We have taken five technical indicators such as Rate of Change (ROC), Money Flow Index (MFI), Relative Strength Index (RSI), StochRSI, Ultimate Oscillator. Based on these technical indicators, datasets that are required for analysis are created.
- Prediction of stock market price is done with the help of fuzzy C-means clustering and artificial neural network
- Evaluation matrices parameters such as mean absolute deviation, mean square error, root mean square error and mean absolute percentage error are used to estimate the forecasting accuracy in the stock market prediction.
- We analyze the performance of our proposed approach with the neural network and existing technique [24] in terms of MAPE, MSE, RMSE and MAD.

4 TECHNICAL INDICATORS

There are numerous technical indicators available to support stock market prediction, namely moving average, exponential moving average, weighted moving average, moving average difference oscillator, relative strength index, volume, volume change, moving average convergence-divergence and more. In our work, we make use of the following technical indexes in order to achieve effective trading rules for real time stock trading:

Money Flow Index (MFI): The Money Flow Index (MFI) is an oscillator that uses both price and volume to measure buying and selling pressure. Created by Gene Quong and Avrum Soudack, MFI is also known as volume-weighted RSI. MFI starts with the typical price for each period. Money flow is positive when the typical price rises (buying pressure) and negative when the typical price declines (selling pressure). A ratio of positive and negative money flow is then plugged into an RSI formula to create an oscillator that moves between zero and one hundred. As a momentum oscillator tied to volume, the Money Flow Index (MFI) is best suited to identify reversals and price extremes with a variety of signals.

Relative Strength Index (RSI): An oscillator, which was introduced by J. Welles Wilder, Jr., is on the basis of the difference between average gains versus average loss over a given period. The RSI compares the magnitude of a stock's recent gains to the magnitude of its recent losses. The RSI has the benefit of being a very elegant indicator, in that its movements are even, and it can fit into a neat package between 0 and 100. It has the added advantage of being utilized by several traders out there, which is not only a testament to its abilities, but it also makes its signals self-fulfilling at times. When used to specify divergences, it can be moderately influential.

Rate of Change (ROC): The Rate of Change (ROC) indicator is a very simple, yet an efficient momentum oscillator that measures the per cent change in price from one period to the next. It is defined as the price change in particular fixed time duration. The Rate of Change symbolizes the momentum and therefore the acceleration or slowing down of a trend. Higher Rate of Change of price (ROC) denotes that stocks are overbought and lower Rate of Change of price (ROC) indicates oversold stock position.

StochRSI: Chande and Kroll developed StochRSI to increase sensitivity and generate more overbought/oversold signals. In their 1994 book, The New Technical Trader, Chande and Kroll explain that RSI can oscillate between 80 and 20 for extended periods without reaching extreme levels. Notice that 80 and 20 are used for overbought and oversold instead of the more traditional 70 and 30. Traders looking to enter a stock based on an overbought or oversold reading in RSI might find themselves continuously on the sidelines.

Ultimate Oscillator: Developed by Larry Williams in 1976 and featured in Stocks & Commodities Magazine in 1985, the Ultimate Oscillator is a momentum oscillator designed to capture momentum across three different timeframes. The multiple timeframe objective seeks to avoid the pitfalls of other oscillators. Many momentum oscillators surge at the beginning of a strong advance and then form bearish divergence as the advance continues. This is because they are stuck with one time frame. The Ultimate Oscillator attempts to correct this fault by incorporating longer timeframes into the basic formula. Williams identified a buy signal based on a bullish divergence and a sell signal based on a bearish divergence.

5 THE PROPOSED TECHNIQUE OF STOCK MARKET PREDICTION USING FUZZY CLUSTERING AND ARTIFICIAL NEURAL NETWORKS

The research procedure will be introduced in this section and the overall framework is shown in Figure 1.

Our proposed work comprises the following phases:

1. the training phase and
2. the testing phase.

The training phase includes the following three major steps:

Step 1: For an arbitrary data set DS, it is firstly converted to data interpretation, i.e. interpretation is performed based on the four technical indicators. Subsequently, the interpretation based dataset is first divided into training set (TR) and testing set (TS). Then the different training subsets $TR_1, TR_2, \ldots, TR_k$ are created from TR with fuzzy clustering module.

Step 2: For each training subset TR_i $(i = 1, 2, \ldots, k)$, the ANN model, ANN_i $(i = 1, 2, \ldots, k)$ is training by the specific learning algorithm to formulate k different base ANN models.

Step 3: In order to reduce the error for every ANN_i, we simulate the ANN_i using the whole training set TR and get the results. Then we use the membership grades, which were generated by fuzzy clustering module, to combine the results. Subsequently, we train another new ANN using combined results.

In the testing phase, we directly input the testing set data into the k different ANN_i and get outputs. Based on these outputs, the final results can then be achieved by the last fuzzy aggregation module.

The three stages of our proposed framework raise four important issues:

1. how to create interpretation of raw data based on technical indicators
2. how to create k different training subsets from the original training dataset TR;
3. how to create different base model ANN_i with different training subsets;
4. how to aggregate the different results produced by different base model ANN_i.

These issues will be addressed in the following sections.

5.1 Interpretation of Raw Data Based on Technical Indicators

When interpreting the raw historical data, the first issue of the proposed approach, is performed to make the data adaptable for further analysis. Here, the interpretation is performed based on the five technical indicators, namely, Money Flow Index (MFI), Relative Strength Index (RSI), Rate of Change (ROC), StochRSI, and Ultimate Oscillater.

A) The Money Flow Index (MFI) based data interpretation is given as follows:

- Typical Price = $\frac{(\text{High}+\text{Low}+\text{Close})}{3}$
- Raw Money Flow = Typical Price $\times$ Volume
- Positive Money Flow = Sum of positive Raw Money Flow over 7 periods
- Negative Money Flow = Sum of negative Raw Money Flow over 7 periods
- Money Flow Ratio = (Positive Money Flow)/(Negative Money Flow)
- Money Flow Index = $100 - \frac{100}{(1+\text{Money Flow Ratio})}$

B) The Relative Strength Index (RSI) based data interpretation is given as follows:

- $RSI = 100 - \frac{100}{1+RS}$ where $RS = \frac{\text{Average Gain}}{\text{Average Loss}}$

C) The Rate of Change (ROC) based interpretation is done by means of calculating the rate of change of each day as follows:

- $ROC_i(m) = 100\left(\frac{c_k}{c_{k-m}} - 1\right); m < k \leq n$ where

– c_k is the closing price
– c_{k-m} is the closing price of m periods ago

D) The Stochastic Relative Strength Index (StochRSI) based data interpretation is given as follows:

- StochRSI = $\frac{(RSI\text{−Lowest Low RSI})}{(\text{Highest High RSI−Lowest Low RSI})}$

StochRSI measures the value of RSI relative to its high/low range over a set number of periods. The number of periods used to calculate StochRSI is transferred to RSI in the formula.

E) The Ultimate Oscillator based data interpretation is given as follows:

- BP = Close – Minimum(Low or Prior Close)
- TR = Maximum(High or Prior Close) – Minimum(Low or Prior Close)
- Average 7 = (7 – period BP Sum)/(7 – period TR Sum)
- Average 14 = (14 – period BP Sum)/(14 – period TR Sum)
- Average 28 = (28 – period BP Sum)/(28 – period TR Sum)
- UO = 100 × [(4 × Average 7) + (2 × Average 14) + Average 28]/(4 + 2 + 1).

Example 1. Consider a stock movement of a company Z. As stated earlier, the data is comprised of low price, high price, opening price, closing price and volume of the company in a particular stock market.

A sample stock movement data for 9 days is given in Table 1.

Over the raw historical data, the technical indicators are applied and the interpreted dataset is obtained. The obtained interpreted data based on the four different technical indicators are shown in Table 2.

5.2 Data Clustering using Fuzzy C-Means Clustering

The purpose of data clustering is to cluster the set of financial time series data into different groups, and data in each group will have more homogeneous characteristics. Through fuzzy clustering module, the training set is clustered into several subsets. Due to the fact that the size and complexity of every training subset is reduced, the efficiency and effectiveness of subsequent ANN module can be improved. Fuzzy C-means is a data clustering algorithm in which each data point belongs to a clustering to a degree specified by a membership grade [23]. In fuzzy clustering module, it is based on the minimization of the following objective function

$$J_m^{TR} = \sum_{i=1}^{N} \sum_{j=1}^{C} u_{ij}^{TR^m} \left\| T_i^{TR} - C_j^{TR} \right\|, 1 \leq m \leq \infty \tag{1}$$

where m is any real number greater than 1, u_{ij} is the degree of membership of T_i in the cluster j, T_i is the i^{th} of d-dimensional measured data, C_j is the d-dimension center of the cluster, and $\|*\|$ is any norm expressing the similarity between any measured data and the center.

Fuzzy partitioning is carried out through an iterative optimization of the objective function shown above, with the update of membership u_{ij} and the cluster centers C_j by:

$$u_{ij}^{TR} = \frac{1}{\sum_{k=1}^{C} \left[\frac{\|T_i - C_j\|}{\|T_i - C_k\|}\right]^{\frac{2}{m-1}}}, \quad (2)$$

$$C_j = \frac{\sum_{i=1}^{N} u_{ij}^{TR} T_i^{TR}}{\sum_{i=1}^{N} u_{ij}^{TR}}. \quad (3)$$

This iteration will stop, when $\max_{ij}\left\{\left|u_{ij}^{TR}(q+1) - u_{ij}^{TR}(q)\right|\right\} < \varepsilon$, where ε is a termination criterion between 0 and 1, whereas q are the iteration steps. This procedure converges to a local minimum or a saddle point of J_m^{TR}.

Based on the above analysis, the fuzzy cluster module is composed of the following steps:

Step 1: Initialize $U^{TR} = \left[u_{ij}^{TR}\right]$ matrix $U^{TR}(0)$ and $q = 1$

Step 2: Calculate the centers vectors $C^{TR}(q)$ and $[C_j^{TR}]$

$$C_j = \frac{\sum_{i=1}^{N} u_{ij}^{TRm} T_i^{TR}}{\sum_{i=1}^{N} u_{ij}^{TR}}$$

Step 3: Update $U(q+1)$

$$u_{ij}^{TR} = \frac{1}{\sum_{k=1}^{C} \left[\frac{||T_i - C_j||}{||T_i - C_k||}\right]^{\frac{2}{m-1}}}$$

Step 4: If $\left|\left|U^{TR}(q+1) - U^{TR}(q)\right|\right| < \varepsilon$ then STOP; otherwise return to Step 2.

After the above five steps, the training set TR can be divided into k subsets TR_k. Subsequently, ANN_i is needed to train using these subsets TR_k. In the next section, we will discuss how to create different base model ANN_i with different training subset TR_k.

5.3 Artificial Neural Network Module

ANN is an artificial intelligence technique that is used for generating training data set and testing the applied input data [22]. A feed forward type NN is used for the proposed method. Normally, a feed-forward neural network has an input layer, an output layer, with one or more hidden layers in between the input and output layer. The input layer consists of five inputs, i.e. T_1, T_2, T_3, T_4, T_5. The ANN functions as follows: each node H_i in the input layer has a signal T_i as network input, multiplied by a weight value between the input layer and the hidden layer.

The training phase is classified into the following three major steps. The training steps involved in neural network are as follows.

Step 1:

- Initialize the input, output and weight of each neuron. Here, T_1, T_2, T_3, T_4, T_5 are technical indicators, i.e. input of the network and $(P_k)\,output$ is the predict value, i.e. output of the network.

Step 2:

- In each node H_i the hidden layer receives the signal $\ln(H_i)$ according to:

$$\ln(H_i) = \eta_i + \sum_{j=1}^{N} T_j W_{nl}. \tag{4}$$

- Then passed through the bipolar sigmoid activation function:

$$f(T) = \frac{2}{(1 + \exp(-T_i))} - 1. \tag{5}$$

- The output of the activation function $f(\ln(H_i))$ is then broadcast to all of the neurons to the output layer:

$$(P_k)\,output = \eta_k + \sum_{n=1}^{N} W_{2nl} P_k(n) \tag{6}$$

  where η_i and η_k are the biases in the hidden layer and the output layer.

Step 3:

- The inputs of training dataset are T_1, T_2, T_3, T_4, T_5 to classify and determine the error function as follows:

$$E_v = (P_k)\,target - (P_k)\,output. \tag{7}$$

  In Equation (7) $(P_k)\,target$ is the target output and $(P_k)\,output$ is the network output

Step 4:

- Adjust the weights of all neurons as $w = w + \Delta w$, where Δw is the change in weight which can be determined as follows:

$$\Delta w = \beta \cdot P_k \cdot E_v. \tag{8}$$

  In Equation (8), β is the learning rate, usually it ranges from 0 to 1.

Step 5:

- Repeat the process from Step 2, until error gets minimized to a least value, i.e.

$$E_v < 0.1 \tag{9}$$

Based on the feed-forward neural networks trained with the back-propagation algorithm, every ANN_i can complete training using different subsets TR_k. However, next question is how to aggregate the different results produced by different base model ANN_i.

5.4 Error Based Neural Network

The target of fuzzy aggregation module is to aggregate different ANN's result and reduce the detection errors as every ANN_i in ANN module only learns from the subset TR_i. Because the errors are nonlinear, in order to achieve the objective, we use another new ANN to learn the errors as follows:

Step 1:

- Let the whole training set TR as data to input the every trained ANN_i and get the outputs:

$$Y_k^{TR} = \left[Y_{k1}^{TR}, Y_{k2}^{TR}, \ldots, Y_{kN}^{TR}\right], \quad k = 1, 2, \ldots n \tag{10}$$

  where n is the number of training set: TR, y_{kN}^{TR} is the output of ANN_k.

Step 2:

- Form the input for new ANN:

$$Y_{\text{input}} = \left[Y_1^{TR}.U_1^{TR}, Y_2^{TR}.U_1^{TR}, \ldots, Y_n^{TR}.U_n^{TR}\right] \tag{11}$$

  where U_n^{TR} is the membership grade of TR_n belonging to C^{TR}.

Step 3:

- Train the new ANN. We can use Y_{input} as input and use the whole training set TR's class label as output to train the new ANN. During the stage of testing, work procedure of ANN module and fuzzy aggregation module is similar to the above. First we calculate the membership grade, based on the cluster centers C^{TR}. For a new input T_i^{TS} is coming, firstly based on C^{TR} , the membership U^{TR} can be calculated by

$$U_{ij}^{TS} = \frac{1}{\sum_{p=1}^{k}\left(\left\|\frac{T_i^{TS}-C_j^{TR}}{T_i^{TS}-C_p^{TR}}\right\|\right)^{\frac{2}{m-1}}}. \tag{12}$$

Then, using ANN module and error based fuzzy module, the output Y^{TS}_{output} can be gotten.

6 SIMULATION RESULTS AND DISCUSSION

This section describes the experimental results of our proposed stock price prediction technique using stock market dataset. Our proposed approach is implemented in Matlab environment on Core 2 Duo, processor speed 1.6 GHz (Matlab version 7.10). The performance of the technique has been evaluated using different historical stock market data.

6.1 Dataset Description

In analyzing the technique, we have used the day-by-day stock price of American Express Company (AXP), Advanced Micro Devices, Inc. (AMD), Bank of America Corporation (BAC), General Motors Company (GM), The Coca-Cola Company (KO). For these companies, we have the stock movement price for 10 years, 2002 to 2012. So, we have generated five stock price datasets for both companies. Each dataset has opening price, closing price, high, low price and volume of the 273 days (for 12 months). i.e. $n = 273$ days.

6.2 Performance Measures

The evaluation of stock market prediction in different stock market datasets is carried out using the following metrics as suggested by the following equations:

- Mean Square Error (MSE)

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(p_i - d_i)^2$$

- Root Mean Square Error (RMSE)

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(p_i - d_i)^2}$$

- Mean Absolute Deviation (MAD)

$$MAD = \frac{1}{N}\sum_{i=1}^{N}|p_i - d_i|$$

- Mean Absolute Percentage Error (MAPE)

$$MAPE = \frac{1}{N}\sum_{i=1}^{N}\left|\frac{p_i - d_i}{p_i}\right|$$

where

- p_i – predicted value
- d_i – desired value
- N – total number of test data

6.3 Experimental Results

Stock market prediction is essential and of great interest because successful prediction of stock prices may promise smart benefits. An efficient stock market prediction technique is proposed to predict the future price. The obtained experimental results include sample results after converting data with technical indicators (dataset 1 to dataset 5), centroid values of dataset 1 and neural networks parameters. Tables 3.1 to 5.5 show sample results after converting data with technical indicators for dataset 1, sample dataset 4 and dataset 5. Tables 6.1 to 6.5 show centroid values of dataset 1. Table 7 shows neural network parameters.

6.4 Performance Evaluation of the Proposed Technique

The performance of our proposed technique is evaluated in terms of the evaluation metrics value, here, with the aid of the input training and testing dataset, the values of MAPE, RMSE, MAD, and MSE. By analyzing the results, our proposed approach is better performance. Table 8 shows MAPE, RMSE, MAD and MSE values of our proposed technique and comparison technique for five different datasets.

6.5 Comparative Analysis

In this section, we have compared our proposed technique with single neural network and existing technique [24]. The performance analysis has been made by plotting the graphs of evaluation metrics such as mean absolute deviation, mean square error, root mean square error, mean absolute percentage. By analyzing the plotted graph, the performance of the proposed technique has significantly improved the stock market prediction compared with single Neural Network (NN), and the existing technique. The evaluation graphs of the mean absolute deviation, mean square error, root mean square error, and mean absolute percentage graph for five stock market datasets are shown in Figures 3 to 7; but the accuracy level proved that the proposed algorithm graph is good in stock market prediction.

7 CONCLUSION

Predicting the stock market index return is important and of great interest because successful prediction of stock prices may promise attractive benefits. A FCM-NN based system is presented in the paper by applying a linear combination of the significant technical index as a consequent to predict the stock price. The results for the stock market prediction is validated through evaluation metrics, namely, mean absolute deviation, mean square error, root mean square error, mean absolute percentage error used to estimate the forecasting accuracy in the stock market. The comparative analysis is carried out on single Neural Network (NN) and existing technique neural. The obtained results show that the proposed approach produces better results than the other techniques in terms of accuracy.

REFERENCES

[1] Huang, H.—Pasquier, M.—Quek, C.: Financial Market Trading System with a Hierarchical Coevolutionary Fuzzy Predictive Model. IEEE Transactions on Evolutionary Computation, Vol. 13, 2009, No. 1, pp. 56–70.

[2] Zhang, D.—Zhou, L.: Discovering Golden Nuggets: Data Mining in Financial Application. IEEE Transactions on Systems, Man, and Cybernetics, Vol. 34, 2004, No. 4, pp. 513–522.

[3] Senthamarai Kannan, K.—Sailapathi Sekar, P.—Mohamed Sathik, M.—Arumugam, P.: Financial Stock Market Forecast Using Data Mining Techniques. Proceedings of the International Multi Conference of Engineers and Computer Scientists (IMECS 2010), Vol. 1, 2010, pp. 555–559.

[4] Hajizadeh, E.—Ardakani, H. D.—Shahrabi, J.: Application of Data Mining Techniques in Stock Markets: A Survey. Journal of Economics and International Finance, Vol. 2, 2010, No. 7, pp. 109–118.

[5] Schumaker, R. P.—Chen, H.: Textual Analysis of Stock Market Prediction Using Financial News Articles. In Proceedings of 12$^{\text{th}}$ Americas Conference on Information Systems (AMCIS '06), Acapulco, Mexico 2006.

[6] Bao, D.—Yang, Z.: Intelligent Stock Trading System by Turning Point Confirming and Probabilistic Reasoning. Expert Systems with Applications, Vol. 34, 2008, No. 1, pp. 620–627.

[7] Klassen, M.: Investigation of Some Technical Indexes in Stock Forecasting Using Neural Networks. World Academy of Science, Engineering and Technology, Vol. 5, 2005, pp. 75–79.

[8] Lee, A. J. T.—Lin, M. C.—Kao, R. T.—Chen, K. T.: An Effective Clustering Approach to Stock Market Prediction. Proceedings of Pacific Asia Conference on Information Systems (PACIS 2010), Paper 54.

[9] Tan, T. Z.—Quek, C.—Ng, G. S.: Brain-Inspired Genetic Complementary Learning for Stock Market Prediction. In Proceedings of IEEE Conference on Evolutionary Computation, Vol. 3, 2005, pp. 2653–2660.

[10] Pearson, R. A.: Tree Structures for Predicting Stock Price Behaviour. ANZIAM Journal, Vol. 45 (E), 2004, pp. C950–C963.

[11] Gorgani, M. E.—Moradi, M.—Yazdi, H. S.: An Empirical Modeling of Companies Using Support Vector Data Description. International Journal of Trade, Economics and Finance, Vol. 1, 2010, No. 2, pp. 221–224.

[12] VANSTONE, B.—TAN, C.: A Survey of the Application of Soft Computing to Investment and Financial Trading. In Proceedings of the 8th Australian & New Zealand Intelligent Information Systems Conference (ANZIIS '03), Sydney, Australia 2003, pp. 211–216.

[13] SHETA, A.: Software Effort Estimation and Stock Market Prediction Using Takagi-Sugeno Fuzzy Models. In Proceedings of the IEEE International Conference on Fuzzy Systems, Vancouver, 2006, pp. 171–178.

[14] SU, C. L.—CHEN, C. J.—YANG, S. M.: A Self-Organized Neuro-Fuzzy System for Stock Market Dynamics Modeling and Forecasting. WSEAS Transactions on Information Science and Applications, Vol. 7, 2010, No. 9, pp. 1137–1149.

[15] FAZEL ZARANDI, M. H.—REZAEE, B.—TURKSEN, I. B.—NESHAT, E.: A Type-2 Fuzzy Rule-Based Expert System Model for Stock Price Analysis. Expert Systems with Applications, Vol. 36, 2009, No. 1, pp. 139–154.

[16] LAI, R. K.—FAN, C. Y.—HUANG, W. H.—CHANG, P. C.: Evolving and Clustering Fuzzy Decision Tree for Financial Time Series Data Forecasting. An International Journal of Expert Systems with Applications, Vol. 36, 2009, No. 2, pp. 3761–3773.

[17] CHEN, S. M.—CHANG, Y. C.: Multi-Variable Fuzzy Forecasting Based on Fuzzy Clustering and Fuzzy Rule Interpolation Techniques. Information Sciences, Vol. 180, 2010, No. 24, pp. 4772–4783.

[18] HO, W. L.—TUNG, W. L.—QUEK, C.: Brain-Inspired Evolving Neuro-Fuzzy System for Financial Forecasting and Trading of the S & P 500 Index. Lecture Notes in Computer Science, Vol. 6230, 2010, pp. 601–607.

[19] HADAVANDI, E.—SHAVANDI, H.—GHANBARI, A.: Integration of Genetic Fuzzy Systems and Artificial Neural Networks for Stock Price Forecasting. Knowledge-Based Systems, Vol. 23, 2010, No. 8, pp. 800–808.

[20] SRINIVASA, K. G.—VENUGOPAL, K. R.—PATNAIK, L. M.: An Efficient Stock Market Prediction Using Fuzzy Based Neuro-Genetic Algorithm. International Journal of Hybrid Intelligent Systems, Vol. 3, 2006, No. 2, pp. 63–81.

[21] SOON, L. K.—LEE, H.: An Empirical Study of Similarity Search in Stock Data. Proceedings of the 2nd International Workshop on Integrating Artificial Intelligence and Data Mining (AIDM '07), Gold Coast, Queensland, Australia 2007, pp. 29–36.

[22] KUMAR, V.-S.—KAVITHA, D.—KALAISELVI, K.—KANNAN, P. S.: Harmonic Mitigation and Power Factor Improvement Using Fuzzy Logic and Neural Network Controlled Active Power Filter. Journal of Electrical Engineering & Technology, Vol. 3, 2008, No. 3, pp. 520–527.

[23] CHIU, S. L.: Fuzzy Model Identification Based on Cluster Estimation. Journal of Intelligent and Fuzzy Systems, Vol. 2, 1994, pp. 267–278.

[24] KUMAR, K.—KAILAS, J. A.: Prediction of Future Stock Close Price using Proposed Hybrid ANN Model of Functional Link Fuzzy Logic Neural Model. IAES International Journal of Artificial Intelligence (IJ-AI), Vol. 1, 2012, No. 1, pp. 25–30.

[25] BUNDZEL, M.—KASANICKY, T.—PINCAK, R.: Experimental Analysis of the Prediction Model Based on String Invariants. Computing and Informatics, Vol. 32, 2013, No. 6, pp. 1131–1146.

[26] Kumar, V.—Sharma, A.—Kumar, R.: Applying Soft Computing Approaches to Predict Defect Density in Software Product Releases: An Empirical Study. Computing and Informatics, Vol. 32, 2013, No. 1, pp. 203–224.

[27] Bartok, J.—Babič, F.—Bednár, P.—Paralič, J.—Kováč, J.—Bartoková, I.—Hluchý, L.—Gera, M.: Data Mining for Prediction and Low Clouds Detection. Computing and Informatics, Vol. 31, 2012, No. 6+, pp. 1441–1464.

Rajendran Sugumar received the B. E. degree from the University of Madras, Chennai, India in 2003, the M. Tech. degree from Dr. M. G. R. Educational and Research Institute, Chennai, India, in 2007, and the Ph. D. degree from Bharath University, Chennai, India, in 2011. From 2003 to 2014, he worked at different positions in various reputed engineering colleges across India. He is currently working as an Associate Professor in the Department of Computer Science and Engineering at Velammal Institute of Technology, Chennai, India. His research interests include data mining, cloud computing and networks. He has published more than 20 research articles in various international journals and conference proceedings. He is acting as a reviewer in various national and international journals. He has chaired various international and national conferences. He is a life time member of ISTE and CSI.

Alwar Rengarajan received the B. E. degree from the Madurai Kamaraj University, Madurai, India in 2000, the M. E. degree from Sathyabama University, Chennai, India, in 2005, and the Ph. D. degree from Bharath University, Chennai, India, in 2011. From 2000 to 2011, he worked at different levels in various reputed engineering colleges across India. He is currently an Associate Professor in the Department of Information Technology at Veltech Multitech SRS Engineering College, Chennai, India. His research interests include network security, mobile communication and data warehousing and data mining. He has published more than 20 research articles in various international journals and conference proceedings. He is acting as a reviewer in various national and international journals. He chaired various international and national conferences.

Chinnappan Jayakumar has more than 14 years of teaching and research experience. He did his Postgraduate in ME in Computer Science and Engineering at College of Engineering, Guindy, and Ph. D. in Computer Science and Engineering at Anna University, Chennai. Currently he is working as Professor in the Department of Computer Science and Engineering, RMK Engineering College. He has published more than 35 research papers in high impact factor international journals, national and international conferences and visited many countries such as USA and Singapore. He has guided a number of research scholars in the area ad hoc network, security in sensor networks, mobile database and data mining at Anna University (Chennai), Anna University of Technology, Sathyabama University and Bharathiyar University. He conducted various national conferences, staff development program, workshops, seminars associated with industries such as Infosys and TCS. He has received Rs 22 Lakhs Grant from AICTE for RPS Project and Staff Development Program. He chaired various international and national conferences. He was Advisor and Technical Committee Member for many international and national conferences.

S. No	Date	Prev Close	Open Price	High Price	Low Price	Volume
1	3, Dec 2012	68.9	74.45	70.75	74.45	3 673 100
2	4, Dec 2012	74.45	76.95	71	73.75	5 309 200
3	5, Dec 2012	73.75	71.9	68	69.05	7 455 000
4	6, Dec 2012	69.05	74.2	68.55	71.05	5 222 600
5	7, Dec 2012	71.05	71.5	66.5	67.4	6 563 200
6	10, Dec 2012	67.4	69.9	67.5	68.05	5 926 900
7	11, Dec 2012	68.05	70.85	67	67.95	5 108 400
8	12, Dec 2012	67.95	70.5	68.7	69.4	13 371 100
9	13, Dec 2012	69.4	71.45	68.05	68.3	5 317 600

Table 1

Days	T1	T2	T3	T4	T5
Day 1 to 7	51.155	74.329	1.000	9.854	54.922
Day 2 to 8	40.1629	62.000	0.282	3.250	49.823
Day 3 to 9	32.519	61.538	0.255	6.521	52.936
Day 4 to 10	30.819	50.667	0	−2.634	52.636
Day 5 to 11	25.391	39.549	0	−8.677	50.1631
Day 6 to 12	21.088	31.804	0	−11.285	48.413
Day 7 to 13	21.556	28.415	0	−16.159	45.611

Table 2. Interpreted dataset based on Money Flow Index, Relative Strength Index, Rate of Change, Relative Strength Index, and Ultimate Oscillator

	29 day T1	**30 day T1**	**31 day T1**	**32 day T1**	**33 day T1**	**34 day T1**	**35 day T1**	**36 day T1**
1	86.1287	80.7653	80.6989	80.1496	89.2384	94.3499	94.0294	89.3933
2	80.7653	80.6989	80.1496	89.2384	94.3499	94.0294	89.3933	89.2282
3	80.6989	80.1496	89.2384	94.3499	94.0294	89.3933	89.2282	82.667
4	80.1496	89.2384	94.3499	94.0294	89.3933	89.2282	82.667	76.0351
5	89.2384	94.3499	94.0294	89.3933	89.2282	82.667	76.0351	78.0368
6	94.3499	94.0294	89.3933	89.2282	82.667	76.0351	78.0368	81.882
7	94.0294	89.3933	89.2282	82.667	76.0351	78.0368	81.882	76.4271
8	89.3933	89.2282	82.667	76.0351	78.0368	81.882	76.4271	76.3868
9	89.2282	82.667	76.0351	78.0368	81.882	76.4271	76.3868	82.2335
10	82.667	76.0351	78.0368	81.882	76.4271	76.3868	82.2335	77.5746

Table 3.1

	29 day T2	**30 day T2**	**31 day T2**	**32 day T2**	**33 day T2**	**34 day T2**	**35 day T2**	**36 day T2**
1	71.7489	63.908	65.6455	64.3182	63.8249	86.7232	80.2198	80.3815
2	63.908	65.6455	64.3182	63.8249	86.7232	80.2198	80.3815	79.5148
3	65.6455	64.3182	63.8249	86.7232	80.2198	80.3815	79.5148	77.9661
4	64.3182	63.8249	86.7232	80.2198	80.3815	79.5148	77.9661	66.5025
5	63.8249	86.7232	80.2198	80.3815	79.5148	77.9661	66.5025	57.265
6	86.7232	80.2198	80.3815	79.5148	77.9661	66.5025	57.265	65.6751
7	80.2198	80.3815	79.5148	77.9661	66.5025	57.265	65.6751	59.6774
8	80.3815	79.5148	77.9661	66.5025	57.265	65.6751	59.6774	58.6777
9	79.5148	77.9661	66.5025	57.265	65.6751	59.6774	58.6777	68.6842
10	77.9661	66.5025	57.265	65.6751	59.6774	58.6777	68.6842	58.9744

Table 3.2

	29 day T3	**30 day T3**	**31 day T3**	**32 day T3**	**33 day T3**	**34 day T3**	**35 day T3**	**36 day T3**
1	1	0.7047	0.7701	0.7201	0.7015	1	0.8434	0.8473
2	0.7047	0.7701	0.7201	0.7015	1	0.8434	0.8473	0.7854
3	0.7701	0.7201	0.7015	1	0.8434	0.8473	0.7854	0.7062
4	0.7201	0.7015	1	0.8434	0.8473	0.7854	0.7062	0.2
5	0.7015	1	0.8434	0.8473	0.7854	0.7062	0.2	0
6	1	0.8434	0.8473	0.7854	0.7062	0.2	0	0.2855
7	0.8434	0.8473	0.7854	0.7062	0.2	0	0.2855	0.0819
8	0.8473	0.7854	0.7062	0.2	0	0.2855	0.0819	0.048
9	0.7854	0.7062	0.2	0	0.2855	0.0819	0.048	0.3876
10	0.7062	0.2	0	0.2855	0.0819	0.048	0.3876	0.058

Table 3.3

	29 day T4	**30 day T4**	**31 day T4**	**32 day T4**	**33 day T4**	**34 day T4**	**35 day T4**	**36 day T4**
1	7.5173	5.5145	5.8423	5.4669	11.7527	12.4239	10.5954	12.1567
2	5.5145	5.8423	5.4669	11.7527	12.4239	10.5954	12.1567	10.0756
3	5.8423	5.4669	11.7527	12.4239	10.5954	12.1567	10.0756	9.6434
4	5.4669	11.7527	12.4239	10.5954	12.1567	10.0756	9.6434	3.1553
5	11.7527	12.4239	10.5954	12.1567	10.0756	9.6434	3.1553	2.2771
6	12.4239	10.5954	12.1567	10.0756	9.6434	3.1553	2.2771	2.4674
7	10.5954	12.1567	10.0756	9.6434	3.1553	2.2771	2.4674	2.1159
8	12.1567	10.0756	9.6434	3.1553	2.2771	2.4674	2.1159	4.3864
9	10.0756	9.6434	3.1553	2.2771	2.4674	2.1159	4.3864	5.1058
10	9.6434	3.1553	2.2771	2.4674	2.1159	4.3864	5.1058	2.9358

Table 3.4

	29 day T5	**30 day T5**	**31 day T5**	**32 day T5**	**33 day T5**	**34 day T5**	**35 day T5**	**36 day T5**
1	46.5782	44.2613	46.9465	47.0266	48.6836	49.3817	52.883	54.4898
2	44.2613	46.9465	47.0266	48.6836	49.3817	52.883	54.4898	48.9987
3	46.9465	47.0266	48.6836	49.3817	52.883	54.4898	48.9987	54.685
4	47.0266	48.6836	49.3817	52.883	54.4898	48.9987	54.685	50.0617
5	48.6836	49.3817	52.883	54.4898	48.9987	54.685	50.0617	47.0496
6	49.3817	52.883	54.4898	48.9987	54.685	50.0617	47.0496	45.4393
7	52.883	54.4898	48.9987	54.685	50.0617	47.0496	45.4393	44.6048
8	54.4898	48.9987	54.685	50.0617	47.0496	45.4393	44.6048	46.7152
9	48.9987	54.685	50.0617	47.0496	45.4393	44.6048	46.7152	47.0432
10	54.685	50.0617	47.0496	45.4393	44.6048	46.7152	47.0432	49.2605

Table 3.5

Tables 3.1–3.5: Sample results after converting data with technical indicators (dataset 1)

	29 day T1	**30 day T1**	**31 day T1**	**32 day T1**	**33 day T1**	**34 day T1**	**35 day T1**	**36 day T1**
1	59.0596	68.3502	66.7468	57.8111	49.818	58.7556	50.3284	57.5017
2	68.3502	66.7468	57.8111	49.818	58.7556	50.3284	57.5017	56.059
3	66.7468	57.8111	49.818	58.7556	50.3284	57.5017	56.059	48.545
4	57.8111	49.818	58.7556	50.3284	57.5017	56.059	48.545	47.9737
5	49.818	58.7556	50.3284	57.5017	56.059	48.545	47.9737	49.1681
6	58.7556	50.3284	57.5017	56.059	48.545	47.9737	49.1681	48.255
7	50.3284	57.5017	56.059	48.545	47.9737	49.1681	48.255	42.0233
8	57.5017	56.059	48.545	47.9737	49.1681	48.255	42.0233	43.7635
9	56.059	48.545	47.9737	49.1681	48.255	42.0233	43.7635	45.1583
10	48.545	47.9737	49.1681	48.255	42.0233	43.7635	45.1583	45.6119

Table 4.1

	29 day T2	**30 day T2**	**31 day T2**	**32 day T2**	**33 day T2**	**34 day T2**	**35 day T2**	**36 day T2**
1	44.0154	48.552	55.6482	48.7047	40.9669	42.6343	38.1519	39.9271
2	48.552	55.6482	48.7047	40.9669	42.6343	38.1519	39.9271	47.1154
3	55.6482	48.7047	40.9669	42.6343	38.1519	39.9271	47.1154	32.1702
4	48.7047	40.9669	42.6343	38.1519	39.9271	47.1154	32.1702	41.087
5	40.9669	42.6343	38.1519	39.9271	47.1154	32.1702	41.087	39.2377
6	42.6343	38.1519	39.9271	47.1154	32.1702	41.087	39.2377	39.2817
7	38.1519	39.9271	47.1154	32.1702	41.087	39.2377	39.2817	33.0401
8	39.9271	47.1154	32.1702	41.087	39.2377	39.2817	33.0401	26.2766
9	47.1154	32.1702	41.087	39.2377	39.2817	33.0401	26.2766	27.2326
10	32.1702	41.087	39.2377	39.2817	33.0401	26.2766	27.2326	32.6531

Table 4.2

	29 day T3	**30 day T3**	**31 day T3**	**32 day T3**	**33 day T3**	**34 day T3**	**35 day T3**	**36 day T3**
1	0.2875	0.4877	0.8008	0.4944	0.153	0.2265	0.0288	0.1337
2	0.4877	0.8008	0.4944	0.153	0.2265	0.0288	0.1337	0.5298
3	0.8008	0.4944	0.153	0.2265	0.0288	0.1337	0.5298	0
4	0.4944	0.153	0.2265	0.0288	0.1337	0.5298	0	0.3798
5	0.153	0.2265	0.0288	0.1337	0.5298	0	0.3798	0.301
6	0.2265	0.0288	0.1337	0.5298	0	0.3798	0.301	0.3029
7	0.0288	0.1337	0.5298	0	0.3798	0.301	0.3029	0.0371
8	0.1337	0.5298	0	0.3798	0.301	0.3029	0.0371	0
9	0.5298	0	0.3798	0.301	0.3029	0.0371	0	0.0325
10	0	0.3798	0.301	0.3029	0.0371	0	0.0325	0.2843

Table 4.3

	29 day T4	**30 day T4**	**31 day T4**	**32 day T4**	**33 day T4**	**34 day T4**	**35 day T4**	**36 day T4**
1	0.3486	−0.2097	−0.2868	−1.4994	−2.4291	−3.2912	−3.2584	−1.5305
2	−0.2097	−0.2868	−1.4994	−2.4291	−3.2912	−3.2584	−1.5305	−2.3149
3	−0.2868	−1.4994	−2.4291	−3.2912	−3.2584	−1.5305	−2.3149	−2.0536
4	−1.4994	−2.4291	−3.2912	−3.2584	−1.5305	−2.3149	−2.0536	−2.6854
5	−2.4291	−3.2912	−3.2584	−1.5305	−2.3149	−2.0536	−2.6854	−1.3816
6	−3.2912	−3.2584	−1.5305	−2.3149	−2.0536	−2.6854	−1.3816	−2.8571
7	−3.2584	−1.5305	−2.3149	−2.0536	−2.6854	−1.3816	−2.8571	−6.0867
8	−1.5305	−2.3149	−2.0536	−2.6854	−1.3816	−2.8571	−6.0867	−5.4224
9	−2.3149	−2.0536	−2.6854	−1.3816	−2.8571	−6.0867	−5.4224	−7.903
10	−2.0536	−2.6854	−1.3816	−2.8571	−6.0867	−5.4224	−7.903	−4.5392

Table 4.4

	29 day T5	**30 day T5**	**31 day T5**	**32 day T5**	**33 day T5**	**34 day T5**	**35 day T5**	**36 day T5**
1	50.542	41.986	47.4666	46.8198	49.3549	47.5208	48.4822	49.5862
2	41.986	47.4666	46.8198	49.3549	47.5208	48.4822	49.5862	52.3238
3	47.4666	46.8198	49.3549	47.5208	48.4822	49.5862	52.3238	50.9385
4	46.8198	49.3549	47.5208	48.4822	49.5862	52.3238	50.9385	48.9376
5	49.3549	47.5208	48.4822	49.5862	52.3238	50.9385	48.9376	48.7442
6	47.5208	48.4822	49.5862	52.3238	50.9385	48.9376	48.7442	49.6257
7	48.4822	49.5862	52.3238	50.9385	48.9376	48.7442	49.6257	47.8053
8	49.5862	52.3238	50.9385	48.9376	48.7442	49.6257	47.8053	47.6053
9	52.3238	50.9385	48.9376	48.7442	49.6257	47.8053	47.6053	42.9363
10	50.9385	48.9376	48.7442	49.6257	47.8053	47.6053	42.9363	46.5275

Table 4.5

Tables 4.1–4.5: Sample results after converting data with technical indicators (dataset 4)

	29 day T1	**30 day T1**	**31 day T1**	**32 day T1**	**33 day T1**	**34 day T1**	**35 day T1**	**36 day T1**
1	72.3932	80.4555	80.8265	79.2528	73.9298	73.8678	67.5802	74.0692
2	80.4555	80.8265	79.2528	73.9298	73.8678	67.5802	74.0692	80.0416
3	80.8265	79.2528	73.9298	73.8678	67.5802	74.0692	80.0416	79.9511
4	79.2528	73.9298	73.8678	67.5802	74.0692	80.0416	79.9511	88.6481
5	73.9298	73.8678	67.5802	74.0692	80.0416	79.9511	88.6481	88.75
6	73.8678	67.5802	74.0692	80.0416	79.9511	88.6481	88.75	80.594
7	67.5802	74.0692	80.0416	79.9511	88.6481	88.75	80.594	72.0838
8	74.0692	80.0416	79.9511	88.6481	88.75	80.594	72.0838	72.9456
9	80.0416	79.9511	88.6481	88.75	80.594	72.0838	72.9456	73.1976
10	79.9511	88.6481	88.75	80.594	72.0838	72.9456	73.1976	73.9373

Table 5.1

	29 day T2	**30 day T2**	**31 day T2**	**32 day T2**	**33 day T2**	**34 day T2**	**35 day T2**	**36 day T2**
1	57.4909	65.8175	69.5836	71.3078	59.3071	56.423	47.3404	46.5969
2	65.8175	69.5836	71.3078	59.3071	56.423	47.3404	46.5969	60.4205
3	69.5836	71.3078	59.3071	56.423	47.3404	46.5969	60.4205	50.8333
4	71.3078	59.3071	56.423	47.3404	46.5969	60.4205	50.8333	64.6781
5	59.3071	56.423	47.3404	46.5969	60.4205	50.8333	64.6781	64.3739
6	56.423	47.3404	46.5969	60.4205	50.8333	64.6781	64.3739	61.552
7	47.3404	46.5969	60.4205	50.8333	64.6781	64.3739	61.552	51.9134
8	46.5969	60.4205	50.8333	64.6781	64.3739	61.552	51.9134	45.3245
9	60.4205	50.8333	64.6781	64.3739	61.552	51.9134	45.3245	51.3222
10	50.8333	64.6781	64.3739	61.552	51.9134	45.3245	51.3222	51.1067

Table 5.2

	29 day T3	**30 day T3**	**31 day T3**	**32 day T3**	**33 day T3**	**34 day T3**	**35 day T3**	**36 day T3**
1	0.7469	0.9679	1	1	0.4551	0.3241	0	0
2	0.9679	1	1	0.4551	0.3241	0	0	0.5594
3	1	1	0.4551	0.3241	0	0	0.5594	0.1714
4	1	0.4551	0.3241	0	0	0.5594	0.1714	0.7317
5	0.4551	0.3241	0	0	0.5594	0.1714	0.7317	0.7194
6	0.3241	0	0	0.5594	0.1714	0.7317	0.7194	0.6052
7	0	0	0.5594	0.1714	0.7317	0.7194	0.6052	0.2151
8	0	0.5594	0.1714	0.7317	0.7194	0.6052	0.2151	0
9	0.5594	0.1714	0.7317	0.7194	0.6052	0.2151	0	0.2308
10	0.1714	0.7317	0.7194	0.6052	0.2151	0	0.2308	0.2225

Table 5.3

	29 day T4	**30 day T4**	**31 day T4**	**32 day T4**	**33 day T4**	**34 day T4**	**35 day T4**	**36 day T4**
1	6.6801	6.5714	8.596	6.8559	2.5174	0.1613	−0.6593	1.924
2	6.5714	8.596	6.8559	2.5174	0.1613	−0.6593	1.924	2.5203
3	8.596	6.8559	2.5174	0.1613	−0.6593	1.924	2.5203	3.0981
4	6.8559	2.5174	0.1613	−0.6593	1.924	2.5203	3.0981	6.5046
5	2.5174	0.1613	−0.6593	1.924	2.5203	3.0981	6.5046	5.3306
6	0.1613	−0.6593	1.924	2.5203	3.0981	6.5046	5.3306	4.5482
7	−0.6593	1.924	2.5203	3.0981	6.5046	5.3306	4.5482	−1.5419
8	1.924	2.5203	3.0981	6.5046	5.3306	4.5482	−1.5419	−2.024
9	2.5203	3.0981	6.5046	5.3306	4.5482	−1.5419	−2.024	−1.8929
10	3.0981	6.5046	5.3306	4.5482	−1.5419	−2.024	−1.8929	−1.3733

Table 5.4

	29 day T5	**30 day T5**	**31 day T5**	**32 day T5**	**33 day T5**	**34 day T5**	**35 day T5**	**36 day T5**
1	53.743	45.0905	50.0606	49.963	49.8867	51.4701	51.752	49.7291
2	45.0905	50.0606	49.963	49.8867	51.4701	51.752	49.7291	51.8238
3	50.0606	49.963	49.8867	51.4701	51.752	49.7291	51.8238	48.0604
4	49.963	49.8867	51.4701	51.752	49.7291	51.8238	48.0604	51.0674
5	49.8867	51.4701	51.752	49.7291	51.8238	48.0604	51.0674	45.6836
6	51.4701	51.752	49.7291	51.8238	48.0604	51.0674	45.6836	49.9746
7	51.752	49.7291	51.8238	48.0604	51.0674	45.6836	49.9746	44.325
8	49.7291	51.8238	48.0604	51.0674	45.6836	49.9746	44.325	40.6888
9	51.8238	48.0604	51.0674	45.6836	49.9746	44.325	40.6888	43.2976
10	48.0604	51.0674	45.6836	49.9746	44.325	40.6888	43.2976	43.6046

Table 5.5

Tables 5.1–5.5: Sample results after converting data with technical indicators (dataset 5)

	Data set 1								
	C1	66.4804	67.0994	67.3342	67.134	66.7491	66.0359	64.9485	63.5327
	C2	16.9803	15.9579	14.9929	14.3166	14.091	14.3038	14.9064	15.9101
T1	C3	49.2957	49.0937	48.9135	48.8799	48.9728	49.1669	49.4557	49.7618
	C4	47.3493	46.6176	45.8052	44.8934	43.9891	43.0386	41.9307	40.9437
	C5	48.511	48.3403	48.1855	48.1661	48.2835	48.5107	48.8563	49.2315

Table 6.1

	Data set 1								
	C1	64.9405	64.9689	64.7277	64.1682	63.5278	62.836	61.608	60.277
	C2	21.7071	20.1113	18.965	18.3962	18.6018	19.5643	21.1315	23.2734
T2	C3	55.3036	55.484	55.6319	55.6929	55.7991	55.9523	56.0262	55.9828
	C4	39.1964	38.8311	38.7051	38.8001	39.0857	39.5265	39.9099	40.4143
	C5	55.2387	55.4921	55.7044	55.799	55.9197	56.0831	56.1669	56.1373

Table 6.2

	Data set 1								
	C1	0.5416	0.5126	0.478	0.4387	0.4048	0.3752	0.3404	0.3078
	C2	0.1748	0.1593	0.1618	0.1856	0.2326	0.2958	0.3639	0.4317
T3	C3	0.5333	0.5321	0.5332	0.5349	0.5377	0.5408	0.5422	0.5357
	C4	0.2822	0.3052	0.3313	0.359	0.3874	0.4186	0.4475	0.4752
	C5	0.5533	0.5512	0.5515	0.5515	0.5524	0.5538	0.5537	0.5463

Table 6.3

	Data set 1								
	C1	7.3523	7.3844	7.2369	6.917	6.5913	6.1304	5.5293	4.8556
	C2	−12.712	−13.580	−14.225	−14.614	−14.662	−14.303	−13.578	−12.504
T4	C3	2.8172	2.8794	2.9913	3.0559	3.1514	3.1884	3.1801	3.1041
	C4	−6.4851	−6.6332	−6.6423	−6.4927	−6.2027	−5.9131	−5.545	−5.0213
	C5	2.8022	2.9073	3.0509	3.1291	3.2379	3.2828	3.2759	3.2053

Table 6.4

	Data set 1								
	C1	49.9771	50.3136	50.5871	50.7308	50.8306	50.9135	50.9765	50.9579
	C2	51.9954	51.6547	51.3358	50.9582	50.6059	50.3437	50.1303	49.98
T5	C3	49.0461	49.033	49.0562	49.0641	49.0605	49.0331	49.0035	48.9699
	C4	48.9646	49.0422	49.1836	49.366	49.6595	49.9438	50.2887	50.6978
	C5	49.0448	49.0234	49.0435	49.0486	49.0373	48.9981	48.9473	48.8972

Table 6.5

Tables 6.1–6.5. Centroid values of dataset 1

	Epoch	No of iteration	Error	Training algorithm	Number of hidden layers
NN1	77	227	32.0395	traingdm	[40, 25, 5, 1]
NN2	123	273	23.9603	traingdm	[40, 25, 5, 1]
NN3	3	153	54.31	traingdm	[40, 25, 5, 1]
NN4	69	219	28.0044	traingdm	[40, 25, 5, 1]
NN5	26	176	33.1953	traingdm	[40, 25, 5, 1]
Final neural network	36	186	34.3375	trainrp	[6, 50, 25, 10, 2, 1]

Table 7. Neural network parameters

Dataset 1				
	MAPE	**RMSE**	**MAD**	**MSE**
Proposed algorithm	21.084525	9.089708	8.294652	82.622794
Single NN	20.811117	9.14673	8.187094	83.662792
Existing technique	21.773754	9.433502	8.565795	88.990962

Table 8.1

Dataset 2				
	MAPE	**RMSE**	**MAD**	**MSE**
Proposed algorithm	24.402376	10.008378	8.794616	100.167627
NN	25.659568	10.773817	9.247708	116.075135
Existing technique	24.691831	10.087870	8.898936	101.765123

Table 8.2

Datasets 3				
	MAPE	**RMSE**	**MAD**	**MSE**
Proposed algorithm	22.935421	9.426674	8.265926	88.862188
NN	26.303954	10.974510	9.479945	120.439880
Existing technique	24.299513	9.909179	8.757545	98.191822

Table 8.3

Dataset 4				
	MAPE	**RMSE**	**MAD**	**MSE**
Proposed algorithm	21.818093	22.993447	20.068282	528.698605
NN	24.506785	27.556833	22.541341	759.379023
Existing technique	23.057404	24.951657	21.208200	622.585182

Table 8.4

Dataset 5				
	MAPE	**RMSE**	**MAD**	**MSE**
Proposed algorithm	14.374456	18.208935	14.661945	331.565315
NN	20.169413	25.765647	20.572801	663.868590
Existing technique	15.184488	19.175812	15.488177	367.711755

Table 8.5

Tables 8.1 to 8.5: MAPE, RMSE, MAD, MSE values of our proposed technique and comparison technique for five different datasets

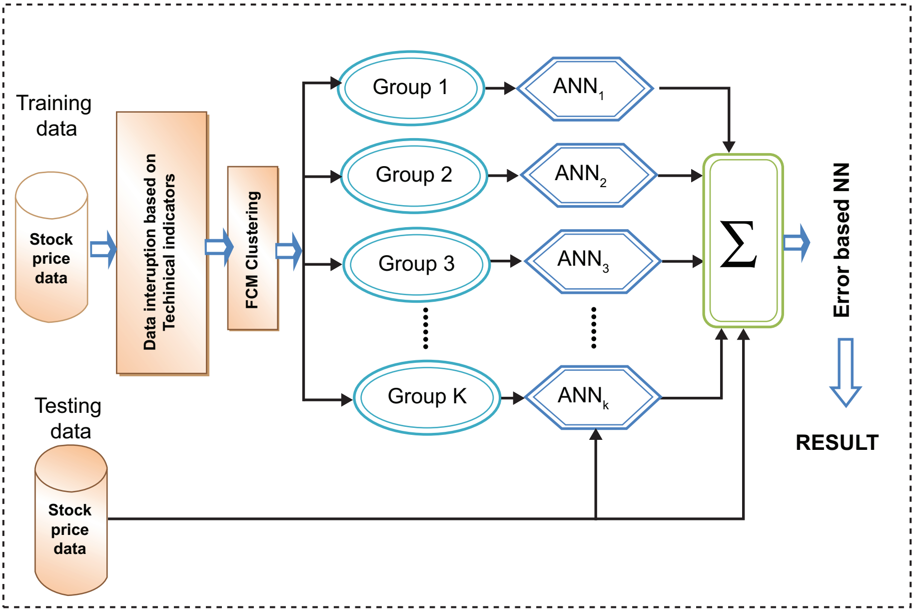

Figure 1. Overall block diagram of our proposed approach

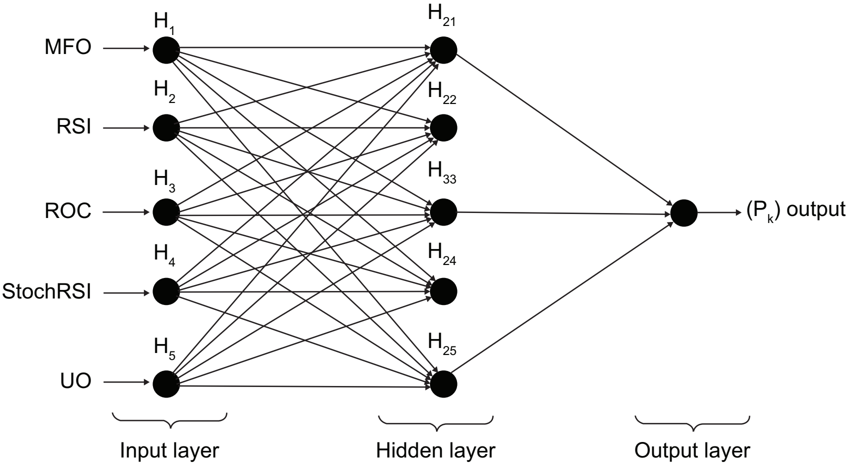

Figure 2. The Structure of NN for proposed method

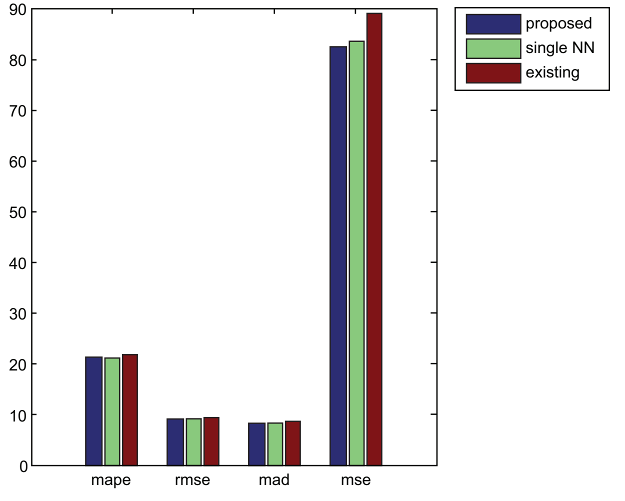

Figure 3. Comparision graph of dataset 1

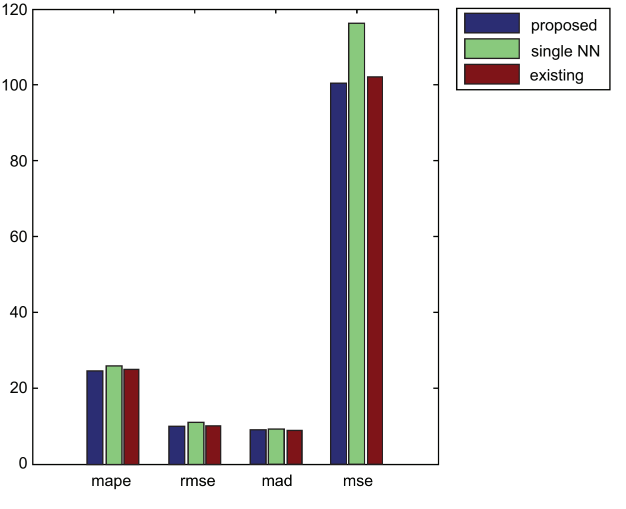

Figure 4. Comparision graph of dataset 2

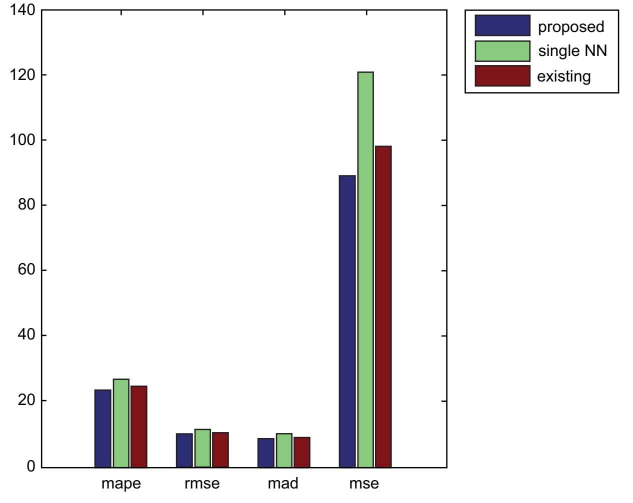

Figure 5. Comparision graph of dataset 3

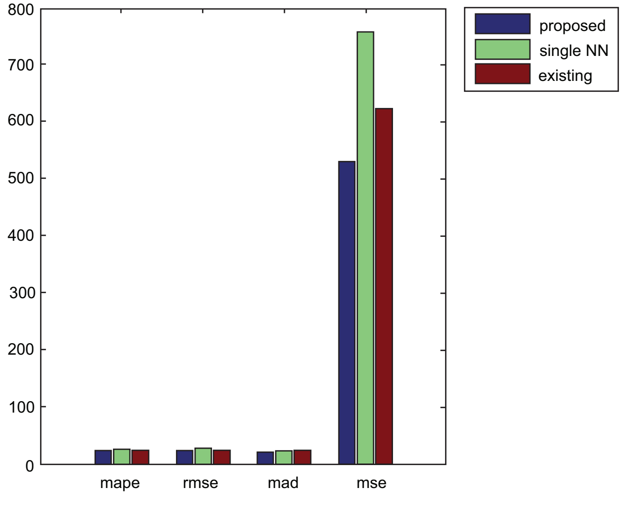

Figure 6. Comparision graph of dataset 4

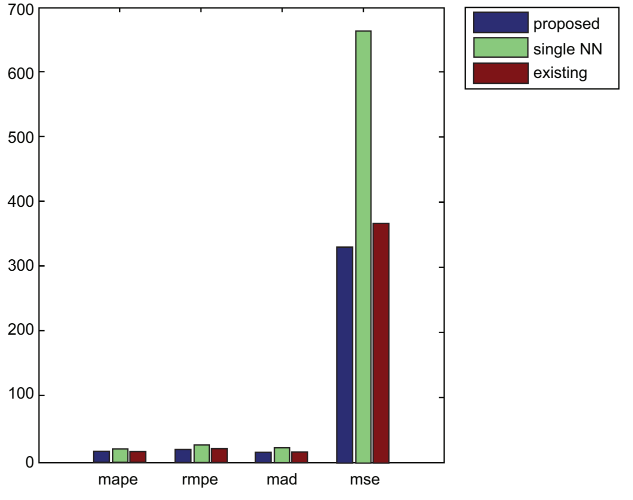

Figure 7. Comparision graph of dataset 5