Research Article

International Journal of Advanced Technology and Engineering Exploration, Vol 9(95)
ISSN (Print): 2394-5443 ISSN (Online): 2394-7454
http://dx.doi.org/10.19101/IJATEE.2021.875852

Stock market prediction in Bangladesh perspective using artificial neural network

Md. Ashikur Rahman Khan[1*], Md. Furkan Uzzaman[1], Ishtiaq Ahammad[2], Ratul Prosad[1], Zayed-Us-Salehin[1], Tanvir Zaman Khan[1], Md. Sabbir Ejaz[1] and Main Uddin[1]
Department of Information and Communication Engineering, Noakhali Science and Technology University, Noakhali, Bangladesh[1]
Department of Computer Science and Engineering, Prime University, Dhaka, Bangladesh[2]

Received: 25-April-2022; Revised: 19-September-2022; Accepted: 24-September-2022

Abstract
Stock market price prediction is now a prominent and significant issue in financial and academic studies as the stock market plays a vital role in the economy. The process of attempting to anticipate the future valuation of a company's share is known as stock market price prediction. Share prices are time-series information, and artificial neural networks (ANNs) can uncover non-linear associations among time-series information. This makes ANN the best method for predicting stock market values. Many researchers are working on this topic and trying to find the best algorithm which is suitable for predicting the stock price. But significant improvement in prediction is still not achieved. Therefore, in this work, an ANN model is proposed and implemented by using a multilayer feedforward backpropagation method. In this work, data of fifteen companies over a six years span have been analyzed. To predict the specific result, the proposed model has been trained with four different algorithms: Levenberg Marquardt (LM), Bayesian regularization (BR), scaled conjugate gradient (SCG) and Quasi Newton) by changing their parameters. Number of hidden layers, hidden neurons and percentage of training data have been changed to get better output. The split ratio of training, testing and validation data sets is 70:15:15. The projected results are then compared to the actual data after the training and testing procedure to determine the accuracy. The accuracy of LM is 95.64%, BR is 91.26%, SCG accuracy is 88.91% and Quasi Newton is 84.20%. The result showed that, LM algorithm provides better accuracy than other models. In addition, less error has been found from the LM algorithm, making it the best algorithm for prediction in our proposed model.

Keywords
Stock market, Price prediction, Artificial neural network, Levenberg Marquardt(LM), Bayesian regularization(BR), Scaled conjugate gradient(SCG), Quasi newton.

1.Introduction
In the last several decades, advances in fundamental components of information technology (IT) have changed the course of business [1]. Stock market becomes a focal point of interest in recent years, owing in great part to technological advancements. According to the World Bank, the global stock market capitalisation has topped 68.654 trillion USD in 2018. The share in the global stock market has risen since then, from 68.654 trillion dollars in 2018 to 116.78 trillion dollars in H1 2021 [2]. In January 2021, Bangladesh's market valuation was 49.942 billion dollars [3]. Investors look for strategies and techniques that may boost profits while lowering risk [4].

Because of its non-linear, variable, chaotic, and unpredictable nature; share market forecasting is not an easy process [5]. Quality standards should be properly followed in order to optimize the performance of any form of service [6]. Statistics on stock market prices is generated in large quantities and fluctuates every second. As a result, the stock market is a sophisticated and difficult system in which people might make profit or end up losing entire savings [7]. It's a disordered system. The general perception of the share market in public is that, it is either too hazardous to invest in or too difficult to trade. Therefore, most individuals are uninterested. The stock market share price prediction

*Author for correspondence

will be the finest for projecting future values to tackle these kinds of problems. However, building an appropriate model is tough since price fluctuation is influenced by a variety of elements including media (both formal media, and social media), data from the internet, principles, corporate production, treasury bonds, historical pricing, and local economics. A prediction approach that just considers one component may not be effective. As a result, combining several inputs such as news, data from the internet, and historical pricing data could improve the performance of the approach.

It's been human's common ambition since the dawn of creation to keep his life more convenient. Because the prevailing belief in society is that, riches offer comfort and luxury. One way to achieve this comfortable and luxury life is investing in stock market properly. In that case, prediction on stock market comes in handy. However, because of the market's extreme volatility, stock market is regarded too unstable to be predictable. Therefore, the job of share market prediction is captivating, and experts and academics are separated into two camps: (i) Those who think they can develop systems to forecast the market, and (ii) Those who think the share market is productive and consumes new knowledge by fixing itself, leaving no room for prediction. Among these two camps, believer of the first group is increasing. Because over the next few years, the most recent advancements in artificial intelligence (AI) (such as neural networks (NNs), machine learning (ML), deep learning (DL), etc.) will have a greater impact on how we engage with business, industry, and the economy [8]. The stock market is crucial for investors and industries, as well as for a country's economies also. But today's investors generally have a trading dilemma since they don't know which stocks to purchase or which stocks to trade in order to achieve the best results. As a result, stock market price forecasting is in high demand. Therefore, it's no surprise that, there has been too much research into stock market forecasting.

As discussed, one of the most recent issues for the AI field has been predicting stock prices [9]. With variable outcomes; a variety of technical, basic, and evaluation metrics of AI field have been presented and applied to solve the issue of prediction. However, neither one of those strategies, or a mixture of them, has proven to be sufficient. Prediction studies have largely surpassed the capabilities of conventional AI research, which has mostly concentrated on constructing expert machines that are meant to mimic human intellect. The share market is complicated and unpredictable by its very nature. Therefore, scholars and investors alike are expecting that, the emergence of NNs can help them to solve share market issues [10]. Artificial neural networks (ANNs), which are based on the activity of human neurons, can understand patterns in the data and extend their expertise to identify upcoming relevant patterns.

Pattern recognition as well as ML tasks such as regression and classification are well-suited NNs, according to research [11]. Nowadays, NNs are widely used in disciplines such as economics, business, manufacturing, and research as a data mining tool. Because of its certain unique qualities, NNs have a bright future in prediction challenges. These unique qualities are:

- To begin with, traditional approaches like regression analysis and logistic regression are model-based, but NNs are self-adjusting techniques that are dependent on training data. As a result, they can resolve the issue with only a basic understanding of the model and without confining the prediction system with additional assumptions.
- Furthermore, since they are basic model approximators, NNs can identify the relationship among the system's input and output, although this relationship is complex. As a result, NNs are well suited to situations in which identifying data correlations is challenging but a massive adequate training data set is available. Even though the principles or patterns we're searching for may not always be obvious, or the data may be distorted because of the system's procedure or assessments noise; inductive training or data-driven approaches are still seen to be the most appropriate way to combat with real-world prediction challenges.
- Next, NNs have the capability to generalize, which means that even after training, it could identify new patterns that weren't in the training set. Because forecasting the upcoming events (unobserved data) is depended on existing data (training set) for most pattern recognition situations; therefore, NNs could be very useful.
- Finally, it has been stated that NNs are basic model approximators. It has been demonstrated that a multilayer feedforward NN can mimic any difficult continuous function, allowing us to understand any difficult interaction among the system's input and output.

As an emerging sector, many scholars are working on this prediction issue and attempting to find the best method for predicting stock prices. However, most of the prior works (given in the literature review section) concentrate just on one organization stock data, or consider only few variables which is insufficient for prediction. In addition, some study focuses mostly on short-term perspective research or takes only few days data, which is also insufficient for price forecasting. Furthermore, the accuracy of stock market forecast utilizing the approaches used in these earlier studies is also very low. Therefore, in our work, a total of fifteen (15) companies six years real data have been used to analyze the stock market price. Data is gathered from reliable sources that has the essential and useful information for the forecast. The primary data source was the Dhaka stock exchange (DSE) website [12]. The investing website [13] is another source of information. This website offers all of the information for every financial firm, including the date, starting value, completing value, peak value, lowest score, number of shares, and changes in stock values. To evaluate the accuracy in our work, four algorithms namely, Levenberg Marquardt (LM), Bayesian regularization (BR), scaled conjugate gradient (SCG), and Quasi Newton (QN) with shifting hidden layer, neuron, and training data are utilized after data collection and training. The training, testing, and validation data sets are split 70:15:15 ratio in this study, and the final output shows that the LM trained model is more accurate and gives less error than others. Investors would simply understand the characteristics of the financial stock market and make predictions using this approach. In summary, the major contributions of this work are:

i. An in-depth literature survey is provided along with the previous work’s shortcomings. An ANN model is proposed in our work to overcome these shortcomings. This depicts the necessity and novelty of our work.
ii. All the necessary steps for the proposed ANN model’s design and implementation are broadly explained. The implementation includes: construction, training, testing and validation of the proposed ANN model.
iii. For training our proposed network, we have used four algorithms. For each algorithm, we have performed the training on different configurations comprising different number of hidden layers and different training data percentage. Regression value, and error histogram is presented for each configuration. Finally, error testing and accuracy testing is performed for best three configurations of each algorithm.
iv. Finally, we have analyzed the results of these four algorithms and their error and accuracy is compared. After comparison, less error has been found from LM algorithm. In addition, LM provides better accuracy than the others.

The following is a breakdown of the paper: Section 1 provides an overview of the stock market and AI, and ANN approaches in order to pique the reader's curiosity in this fascinating subject. A goal is determined and a research aim is presented by explaining and analyzing the surroundings. We addressed some of the current research in the field of stock market forecasting using AI, and ANN approaches in section 2. Readers will learn about previous studies, study types, and drawbacks in this section. Section 3 explains the work's requirements and how it will be carried out. This part serves as the foundation for the research strategy and design. The results and explanation are presented in section 4. It uses figures to show the prediction performance of the four ANN algorithms utilized, which helps investors to know about the market position. Ultimately, section 5 discusses the work's overall accomplishments and concludes it.

2.Literature review

The literature review is performed on the studies which relates on the sectors of our work. When we want to study in depth on a research topic, then we first have to focus on the recent review papers on that topic [14]. That’s why in this literature work, initially we performed the study on review works which are related to stock market prediction using AI and its communities. Then, we have performed the study on actual research work on stock market prediction. The novelty of our work compared to the related previous works can be easily understood from the summary of literature, which is presented at the end of the section. In [15], the authors give a systematic overview of ML techniques to stock market prediction. Their research looks into a variety of methodologies for predicting stock prices, ranging from classical ML and DL to neural nets and graph-based strategies. It examines the methodologies in depth, and the obstacles they imply, as well as the sector's potential scope of study. The authors of [16] present a decade-long assessment of approaches, recent advancements, and future perspectives in stock price prediction using ML approaches. The studies from the previous decade (2011–2021) were scrutinized. The study will aid upcoming researchers in understanding the fundamentals and progress of this new field, allowing them to pursue additional research in potential paths. Another article in the same genre is presented in [17]. Based on a survey of latest research, the goal of this work is to indicate areas for future ML share market forecasting. A systematic literature review (SLR) process is utilized to find appropriate peer-reviewed journal papers from the last two decades and classify studies with similar techniques and topics. There are four types of studies: ANN studies, support vector machine (SVM) studies, genetic algorithms mixed with other techniques, and hybrid as well as other AI approaches. All genre's studies are examined for generic findings, unique findings, limits, and areas that require additional research. The final part concludes with a summary of the findings and recommendations for future research.

The work [18] presents a SLR of DL algorithms for stock price prediction in the European Union. The planned SLR has 12 articles, reflecting that there still isn't a lot of development in this domain, which suggests that there is still need for more research. The researchers in [19] presents yet another overview of stock forecasting using ML approaches. This overview study explores several ML strategies (supervised or unsupervised) and techniques for investors to learn when stock values rise or fall. It was completed in five stages: data collection, datasets pre-processing, extraction of features, share price forecasting using several algorithms, and visualization of the results. In [20], the authors also present a comparable piece of work. In this literature study, many ML strategies are explained. The authors look at how ML techniques could be used to forecast stock prices. There are a variety of parameters which can be used to train the algorithm for this goal. A few of the other aspects that can affect the stock price are also highlighted.

After performing the deep literature survey, we now focus on the actual research works on stock price prediction. Here, we discussed the proposed models/approaches by the researchers, their accuracy, and the limitations of these works.

Pyo et al. [21] conduct a study that uses nonparametric ML approaches such as ANN, SVMs with polynomials and radial basis functional kernels to forecast the trend of the KOSPI 200 values. Utilizing Google trends, the results revealed a highest accuracy of 52% for short durations. That is, the ensemble approaches did not increase the prediction performance. For real-world investors, the accuracy of this research is insufficient. The researchers offered various ML techniques to anticipate the price of Dhaka share market in [22]. They evaluate this procedure in order to improve accuracy. They use deep neural network (DNN), recurrent neural network (RNN), long short-term memory (LSTM), and SVM methods to anticipate market rate. DNN have the highest accuracy rate of 88%. Although precision is good for this work, but it is insufficient. Furthermore, for this research, sentiment analysis employing posts on social media does not yield particularly reliable results. The work [23] explains how to anticipate a stock using ML. The authors suggest a ML approach which will be taught using publicly available market information to gather intelligence, and then utilize that information to make a reliable prediction. In this regard, the study used SVM to estimate share prices for big and small capitalizations, as well as in three separate markets, using daily and up-to-the-minute data. Chen et al. [24] shown a variety of ML techniques for stock price prediction, including NNs, genetic algorithms, SVM, and others. LSTM was used to analyze and simulate Chinese stock market returns. The model was trained on 900000 patterns and then evaluated on the remaining 311361 patterns. Their LSTM model enhanced the efficiency of stock market returns forecast from 14.3% to 27.2% as compared to an arbitrary prediction technique. The attempts revealed the potential of LSTM in stock price forecasting in China, which is technical yet far more uncertain. This model just looked at stock market predictions in China, and it left out a bunch of other aspects. Additional data, such as global market indices, mass commodity prices, recent business news, and perhaps even social media atmosphere, must be considered as learning features.

The authors used an ANN to anticipate the NASDAQ share market in [25]. They utilized several years of collected data and multiple ANN techniques to train the systems, varying the hidden layer quantity, hidden neuron, and proportion of training data. In learning using five inputs, ten neurons with in input layer, and one in the output layer, the system's accuracy was 99%. 1378.0411 is the highest validation result (i.e., mean squared error). Even though the share closing values are greater than 3000, the inaccuracy is less than 2%. However, there are certain limitations to the work. To improve accuracy, the important impact of particular domain analytical parameters on the quality of share price prediction must be identified. The work [26] does Indian share market forecasting utilizing ANNs on tick data.

Authors in [27] presents another stock price forecast using NN with backpropagation technique. The finest structure achieved after the training was 8: 9: 1. The test was then performed on test data utilizing the optimal network model, and the MSE was measured to be 0.1830. However, various forms of NN can be used to forecast, as well as altering the training algorithm to better manage the modest price differences across periods, leading in a prediction that is closer to the real data. The authors examine the outcomes of NNs based on three distinct learning approaches, namely, LM, SCG, and BR, regarding stock price prediction using tick statistics and 15-min statistics from an Indian company. Utilizing tick statistics, all three approaches have a 99.9% accuracy rate. The accuracy for LM, SCG, and BR across a 15-minute dataset falls to 96.2 percent, 97.0 percent, and 98.9 percent, respectively, which is much lower than the findings achieved from tick data. However, the authors just utilize data from the last 30 working days. Seasonal and annual elements that influence share prices can be incorporated into a larger dataset. Furthermore, because well-known persons' comments and views are believed to influence share prices, certain sentiment analysis might aid in share price prediction. The researchers in [28] want to construct a share price prediction model by merging ANN and decision trees (DTs). The integrated DT+ANN model has 77% accuracy, which is greater than the standalone ANN and DT designs in the electrical sector, according to the results obtained. However, there are several limits to this approach that must be taken into account. For more assessments, other approaches such as SVMs, genetic algorithms, and so on can be considered.

A work is done at [29] for forecasting the share closing price employing ML approaches. In this study, ANN and random forest approaches were used to forecast the following day closing price of five companies operating in various industries. Standard strategic metrics such as root mean squared error (RMSE) and mean absolute percentage error (MAPE) are used to assess the systems. These two parameters have low values, indicating that the systems are good at predicting share closing prices. The authors of [30] addressed numerous approaches for predicting future closing share prices that may increase or fall more accurately than the significant level. They made a lot of trading in the markets utilizing automated computer algorithms that incorporated data mining and forecasting technology. This technique is intended to assist investors in uncovering hidden correlations in historical data which have the potential to anticipate their investing decisions. In [31], the authors describe a convolutional neural network (CNN)-bidirectional LSTM (BiSLSTM)-based share closing price forecasting model. The model uses a CNN to retrieve sophisticated elements that affects stock price, and then uses BiSLSTM to forecast share closing price when CNN has analyzed the data. The CNN-BiSLSTM is trained and tested using past data from the Shenzhen Component Index spanning 1991 to 2020 to validate the model's efficiency. MLP, RNN, LSTM, BiLSTM, CNN-LSTM, and CNN-BiLSTM are contrasted to CNN-BiSLSTM. The MAE, root-MSE (RMSE), and R-square (R2) assessment parameters of the CNN-BiSLSTM are all ideal, according to the observed measurements.

The authors provide a ML-assisted approach for assessing the equity's long-term price in [32]. In 76.5% of the time, their approach can correctly forecast if a share performance would increase by 10% or not over the course of a year. The study's shortcoming is that the designs were not built using data that was not time-limited. The authors [33] employ a sophisticated DL method to anticipate short-term share market price trends. The authors gathered data from the Chinese share market for two years and provided an advanced feature design and DL-dependent framework for forecasting share market price trends. Shen et al. [34] introduced a novel prediction technique that takes SVM to forecast the next-day share trend by exploiting the temporal association between universal share markets and numerous financial items. ML methods like as SVM and reinforcement learning are used in several research to forecast the trends in the share market. The Nasdaq, the S&P 500, and the dow jones industrial average (DJIA) are three indices that are regarded as indicators of market efficiency on any particular day. They serve as the foundation for numerous investment goods that are based on the daily changes in their prices. The Nasdaq has a predictive performance of 74.4%, the S&P500 has 76%, and the DJIA has 77.6%. Furthermore, it is necessary to investigate additional innovative and effective approaches, as this is insufficient. The model can be adjusted to account for taxes and fees in the trading system.

From the above studies, we can see that; the previous works has some limitations which we presented alongside every work. Some provide less accuracy; some takes only a few days data; some mainly focused on short-time perspective analysis; some takes only one or two input variables, and some considers one or two algorithms for training the data. These features are not perfect for stock price prediction. To overcome these limitations, we performed our research. In our work, a total of fifteen (15) companies six years real data have been used to analyze the stock market price. To improve accuracy, four algorithms with shifting hidden layer, neuron, and training data are utilized after data collection and training. A large amount and long duration spanning dataset, and utilizing four different algorithms for finding the best prediction performance, provides unique novelty of our work compared to the previous ones.

3.Methodology

This section depicts the core part of our research. This research's intended strategy for constructing the prediction system comprises mostly of three parts. (i) To begin, data from multiple sources is gathered and sorted for relevance. (ii) Second, the acquired data is analyzed by looking at the current market trends, following the industry body and particular businesses, and then the data is reported and rated appropriately. (iii) Finally, an ANN is created, and the most accurate method for predicting stock value is picked.

3.1Data collection and processing

This study aims to forecast the stock's worth based on its prior value and patterns. It necessitates stock market data. As a result, a reliable source with relevant and sufficient data for the forecast is essential. The main source of data would be the DSE [12]. The investing website [13] is another source of information. This site offers all of the information for every financial firm, including the date, starting value, ending value, peak value, minimum score, amount of stock, and changes in share prices. This also shows the DSE's overall effectiveness as well as the performance of corporations in several sectors. In this research, the considered attributes are day, week, month, year, company name, opening price, lowest price, and highest price. The attribute day, week, month, and year means particular time for which we collect and consider the data. The company name specifies the name of a company which we consider for data collection. The opening price means the price of share on any particular day when the share market starts. The highest and lowest price means the maximum and minimum value for a share in a particular day. The considered output is price.

Here, we have collected 15 company's 5 years data. For any particular company, we have collected approximately 190 data per year. This makes (5×190) = 950 data for 5 years for each company. Therefore, total (15×950) = 14, 250 data are collected for our research. *Table 1* shows a sample of the collected data.

Data pre-processing is an essential step that must be done with caution and properly. First and foremost, the excel file contains all of the obtained raw datasets. Then, to exclude any duplicate entries in our data, use the remove duplicate records operator. Then implement the normalized rules to all of the columns. The rule is as follows Equation 1:

$$X_{normalize} = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{1}$$

The two aspects of stock market analysis are: (i) fundamental analysis, and (ii) technical analysis.

Table 1 Sample of collected data

Date	Price	Opening	High	Low	Share(K)	Change (%)
Jan 31, 2019	87.00	89.00	89.50	86.00	191.83	-1.69
Jan30, 2019	88.50	89.40	89.80	88.30	180.75	-1.01
Jan 29, 2019	89.40	90.40	90.40	88.90	225.96	0.34
Jan 28, 2019	89.10	87.10	90.90	87.10	471.23	1.37
Jan 27, 2019	87.90	88.00	88.20	87.20	181.90	-0.23
Jan 24, 2019	88.10	87.60	88.30	87.50	335.99	1.15
Jan 23, 2019	87.10	87.20	88.20	86.70	321.09	0.23
Jan 22, 2019	86.90	86.00	87.50	85.80	408.40	0.58
Jan 21, 2019	86.40	86.00	86.60	85.00	430.36	0.38
Jan 20, 2019	85.80	85.50	86.30	84.50	372.09	1.30
Jan 17, 2019	84.70	84.70	85.40	84.30	237.16	-0.82
Jan 16, 2019	85.40	84.90	86.40	84.70	655.90	0.95
Jan 15, 2019	84.60	85.50	86.00	81.10	535.30	-1.05
Jan 14, 2019	85.50	86.10	86.90	79.70	276.71	-1.16
Jan 13, 2019	86.50	86.10	87.00	85.60	575.16	0.46
Jan 10, 2019	86.10	86.60	86.60	85.50	254.72	0.23

Date	Price	Opening	High	Low	Share(K)	Change (%)
Jan 09, 2019	85.90	86.40	86.60	85.50	521,94	-0.58
Jan 08, 2019	84.90	85.50	86.80	84.10	644.31	1.77
Jan 07, 2019	84.90	86.00	86.00	83.00	517.83	-1.05
Jan 06, 2019	85.80	84.30	86.20	82.80	811.80	1.78
Jan 03, 2019	84.30	81.80	85.50	81.80	1210	3.69

Fundamental analysts are interested in the business that underpins the stock. They assess a company's prior performance and the accuracy of its financial statements. Many performing metrics are developed to assist fundamental analysts in determining a stock's legitimacy. Fundamental analysis in the share market is the process of determining a stock's actual worth, which could then be matched to the price at which it is transacted on stock exchanges, to determine, if the shares are undervalued or otherwise [35]. Different methods with essentially the same idea can be used to find the actual worth. The central principle would be that a corporation is worth the sum of its potential earnings. These potential gains must be adjusted to their current worth as well. This notion aligns with the belief that a company's sole purpose is to maximize profits [36].

Fundamental analysis, in contrast to technical analysis, is viewed as a long-term policy. Fundamental analysis is based on the idea that human civilization need money to advance. Therefore, if a business does well, this should be awarded with more capital, resulting in a share price increase. Fundamental analysis, similar to financial statement analysis [37], is extensively utilized by investment firms because it's the most acceptable, balanced, and based on openly available facts.

Technical analysts, sometimes known as chartists, are unconcerned about a company's basics. They try to predict the future value of a share simply looking at previous price movements. Methods like as the exponential moving average (EMA), oscillations, support and resistance points, and power and quantity indications are employed in conjunction with the patterns. Candlestick patterns, which are thought to have originated with Japanese rice traders, are now commonly utilized by chartists [38].

Short-term tactics are more commonly used instead of long-term tactics in technical analysis. As a result, it's significantly more common in goods and foreign exchange (FX) marketplaces, where traders are more concerned with short-term price changes. This analysis relies on a few fundamental assumptions. Firstly, everything important about a firm is already factored into its stock price. Other factors include price movement in waves and the fact that past (of prices) seems to repeat themselves, according to market sentiment [39].

3.2Neural network model for prediction

The structure of the human brain inspired the development of ANNs. Nodes, links, and weighting in an ANN reflect the neurons, synapses, and electrical signal levels in the nervous system of brain, correspondingly [40]. An ANN can acquire many different shapes, with links, loops, and contacts in multiple patterns allowed. But only a particular type of NN (the feed-forward) is utilized in this work.

Every data from the input nodes is fed up to every node on the hidden units, then to every node in the output units in an ANN. Each level can have any quantity of nodes, and there are frequently several hidden levels to travel through before hitting the final output level. When constructing a NN, it's critical to pick the proper number of nodes as well as layers. The utilized NN for this work is feed-forward network and it's presented in *Figure 1*.

The ANN's input level could be considered as its "sensing organ." It's the location where user controls the environment's attributes (i.e., the data ANN to make a decision on). Because the parameters of the neurons throughout this level are established by an external source, and they don't have any incoming links. This information is sent to the neurons of the following level in the hierarchy via the outgoing links. In most cases, "hidden" layer exists among the incoming and outgoing levels. Arriving links from the previous layer and outward links to the following layer are shared by the neurons in these levels. The system's "cognitive brain" is represented by the hidden units. The output unit is the outermost layer, in which the ANN's results are displayed as the final outcome. In our proposed NN structure (shown in *Figure 1*), the input layer parameters are company no., day no., day of week, day of month, month, and year. The output is the predicted evening price for these input data.

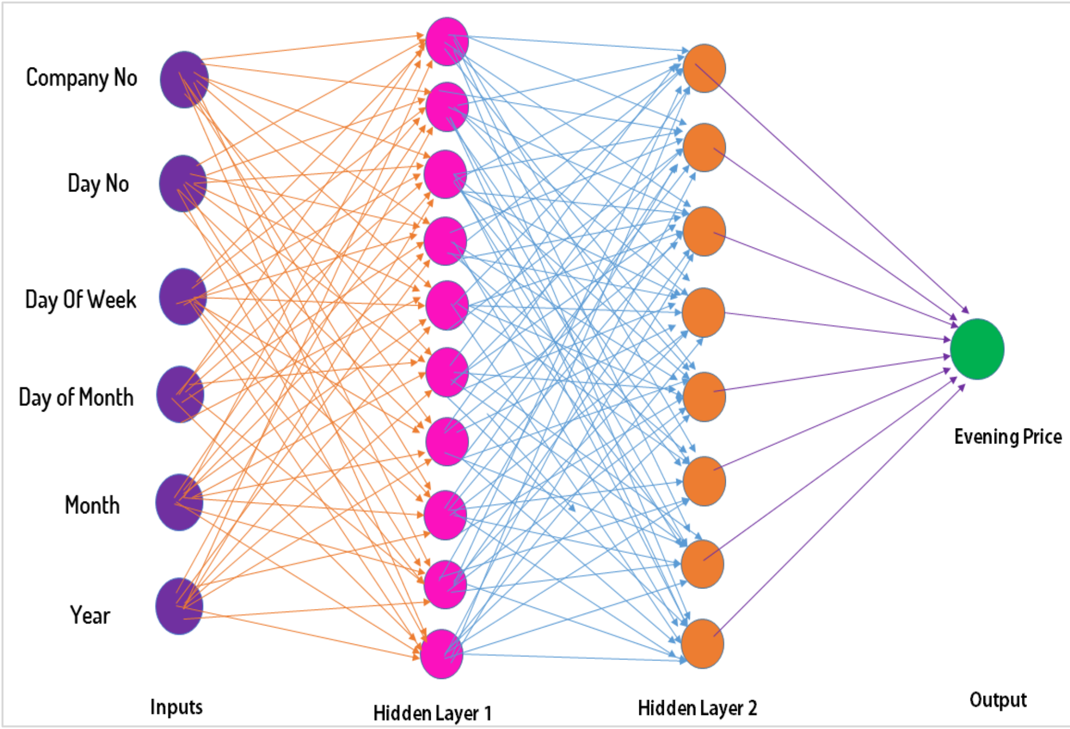

Figure 1 Proposed ANN Structure for this paper's Work

Feed-forward neural network
One of most common and widely utilized structure is the feed-forward neural network (FFNN), which is an ANN with a feedforward layout. The only requirement for this network is that data should move in just one way, through input to output, without any cycles or repetitions in between. A FFNN is made up of connected neurons, with every neuron in the input nodes linked to every neuron in the next level nodes, and every neuron within that layer linked to every neuron in the next level, and so on, till the output level is reached [41]. There have been no restrictions on the number of levels that can be used. However, expanding the number of levels increases the network's complexity.

The reason that, FFNNs have been effectively used for function optimization is, they have contributed to their appeal. That is, with the correct number of neurons in the hidden level, they may estimate any functional form extremely well. As a result, FFNNs can be used in scenarios with uncertain nonlinear interactions, making them suited for use in the stock market.

An ANN must be trained and taught how to manage input data before it can be used. This education process relates to adjusting the weights as well as biases. Because then the difference among the outputs and predicted values will be as little as possible. Various training approaches for the ANN could be utilized based on the structure and activation function. Since its simply generated and continuous derivation; the backpropagation method is usually utilized in FFNNs in conjunction with a sigmoid function. Furthermore, because an FFNN always feeds data forward, errors being transmitted backwards whereas the weights and biases are adjusted based on the error. Lastly, as the time required to train the network rises exponentially, FFNNs using the backpropagation technique do not contain numerous levels for practical purposes.

There are numerous strategies for training and evaluating a ANN. However, not all of these have shown positive results. For our experiment; LM, BR Algorithm, SCG Algorithm, and Quasi Newton Algorithm are considered as they provide comparatively good result.

Levenberg Marquardt(LM)
This approach usually necessitates extra memory, but it takes less time. Once generalization stops increasing, as seen by a rise in the MSE of the test dataset, training comes to an end automatically. The LM algorithm was created particularly for loss

functions that are expressed as a sum-of-squared-errors (SOSE). It functions, even if the actual Hessian matrix isn't computed [42]. Rather, it uses the Jacobian matrix as well as the gradient vector. The LM approach is designed for SOSE functions. While training NNs based on such errors, this results in an extremely quick training time.

Bayesian regularization (BR)

BR is a training method that generates LM optimization to modify the weights as well as bias values. It identifies the suitable combination to construct a network which generalizes well by minimizing a mixture of squared errors and weights [43]. In the training optimization method, network weights are introduced which is represented as $F(\omega)$ in Equation 2.

$$F(\omega)=\alpha E\_\omega+\beta E\_D \quad (2)$$

Where, E_ω is the sum of the squared network weights and E_D is the sum of network errors. Both α and β are the objective function parameters. The network weights are treated as variables in the BR paradigm, and the dispersion of the network weights and test sets is treated as a Gaussian distribution.

The variables that had to be improved were α and β. Whenever the optimal parameters for α and β for a specific weight field are established, the BR algorithm enters the LM phase, which involves Hessian matrix computation and updating the weight field to optimize the objective function. However, if the solution is not achieved, the BR algorithm estimates fresh numbers for α and β and repeats the process unless convergence is achieved.

Scaled conjugate gradient (SCG)

The weights are adjusted in the gradient descent route, that is the most negative of the gradient, by the fundamental backpropagation method. This is the quickest-decreasing route for the evaluation function. It shows that, while the function reduces the most in negative gradient, it doesn't always result in the quickest convergence. The conjugate gradient (CG) strategies operate in a route that offers usually faster convergence compare the gradient descent route while maintaining the error mitigation done in the previous phases [44]. The conjugate route is the name given to this direction. The step-size is modified throughout most CG techniques at each cycle. To identify the step size which will reduce the evaluation function within that path, a search is conducted along the CG directions. The search is carried out utilizing conjugate directions within CG training technique, which accelerate convergence rate over gradient descent methods. In terms of the Hessian matrix, such training orientations are connected. In the training of NNs, this strategy has considered to be more efficient than gradient descent. The CG is also preferred if we have large NNs because it does not need the Hessian matrix.

Quasi Newton

Because evaluating the Hessian matrix and computing its inverse involves numerous processes, Newton's technique is computationally costly. To address this flaw, alternative methodologies called as quasi-Newton methods have been developed. Rather than explicitly computing the Hessian and afterwards assessing its inverse, such methods construct an estimation to the inverse Hessian at every step of the process [45]. Only the initial derivatives of the gradient descent are used to generate this estimation.

3.3 Implementation (Construction, Training, and Testing)

The main difficulty in forecasting the stock market is that, it is a chaotic system. There are numerous factors that could have an explicit or implicit impact on the stock market. The variables have no significant relationships with the price. We are unable to establish a mathematical relationship between the variables. There seem to be no formulas that can be used to forecast the value of a stock using such data. The NN technique is appropriate for this type of chaotic system since we do not need to know the solution. This is really a significant benefit of NN techniques. We only need to present the desired outputs for the provided inputs with the NN. With enough training, the system will be able to duplicate the function. Another benefit of NN is that it will learns to reject any inputs which do not contributing to the output during the learning process. The parameter's designated weights are provided in the training stage of our developed system, and also the Backpropagation approach is utilized for this training stage. The very same formulas that were utilized in the training stage are applied in the prediction step to utilize these weights.

The proposed ANN model of our work is designed and implemented by MATLAB. The implementation included construction, training, testing and validation of the ANN. The steps which are followed:

i. At first data is inserted in the network considered input and target layer, and then these data are loaded. After inserting the data of input layer and target layer, we can define the variable for these input and output data. Six data has been considered as input and these are: company no, day no, day of week, day of month, month, and year. Different number of output layer has been considered but using one output layer got best accuracy. Therefore, six input and one output layer is selected for this network.

ii. After defining input and output variable, we have to select the training algorithm. For training this network, four training algorithms have been used and these are: LM, BR, SCG, and QN algorithm.

iii. The third step defines the hidden layer of the network. In our research, different number of hidden layers have been used. One hidden layer with 10 neuron, two hidden layer with 10 and 8 neuron, two hidden layer with 5 and 5 neuron, and two hidden layers with 10 and 5 neurons have been used. From two hidden layer with 10 and 8 neurons, we get better output. Anyone can use different number of hidden layers in this network design and then configure the network.

iv. In fourth step, the input and target data are randomly divided into three sets for training, testing and validation. In this designed network, various percentage of training, testing and validation dataset have been used, such as 70% or 80% or 90% for training; 15% or 10% or 5% will be used to validate that the network is generalizing and to stop training before overfitting. Then last 15% or 10% or 5% will be used as a completely independent test of network generalization.

v. At the last step, we train the network and view the network and save the output value for check the accuracy.

Table 2 shows the different combination of changing hidden layer with hidden neuron and training, testing and validation data to train the network.

Table 2 Changing hidden layer with hidden neuron and training data

S.N.	Hidden layer	Neuron	Training data	Testing data	Validation data
01	1	10	70%	15%	15%
02	1	10	80%	10%	10%
03	1	10	90%	5%	5%
04	2	5 and 5	70%	15%	15%
05	2	10 and 5	70%	15%	15%
06	2	10 and 5	80%	10%	10%
07	2	10 and 8	70%	15%	15%
08	2	10 and 8	80%	10%	10%

3.4 Performance analysis

Different sets of measures can be utilized and compared to examine the effectiveness of our suggested ANN model. These are: regression analysis, error histogram, MSE and accuracy testing.

Regression Analysis: A group of statistical procedures for estimating relationship among variables (both dependent and independent) is known as regression analysis. An independent variable is indeed a modification in an input, hypothesis, or trigger that is used to evaluate its influence on a dependent variable.

Error Histogram: The histogram of errors among intended values and estimated values upon training a FFNN is known as the error histogram. These error numbers can be negative because they represent how anticipated values depart from target values.

Mean Square Error: The MSE of an estimator (a process for predicting an unknown quantity) is a metric that quantifies the mean of the squared errors, or the mean squared difference among the expected and actual numbers.

Accuracy Testing: The primary purpose of this work is to forecast stock prices in order to assess their accuracy. This calculation could be used to determine the accuracy from the estimated value Equation 3:

$$accuracy = \frac{\text{Target value} - \text{Predicted value}}{\text{Target Value}} \times 100 \tag{3}$$

The target value and estimated value are contrasted in accuracy testing. We have used some sample data for accuracy testing that had not been previously trained. *Table 3* displays the sample of these data.

Table 3 Sample data for accuracy testing

Company name	Day No	Day of week	Day of month	Month	Year	Evening price
ICBS	54	5	4	8	16	6.3
DBHF	12	1	16	1	20	109.8
GP	19	1	19	1	20	263.3
LANKBNG	142	4	5	8	15	14.78
SQUARE	99	5	28	5	15	163.49
NAVANA	14	4	20	1	16	49.6
IDLC	29	4	11	2	15	47.66
PRAN	29	2	29	1	19	250.4
LANKBNG	119	2	3	7	19	19.62
CITY BANK	55	1	24	2	19	15.81
SQUARE	45	3	14	2	17	218.24
RUPALI LIFE	229	2	12	12	16	34.13
DBHF	177	3	26	19	16	105
ICBS	27	1	29	11	15	5.7
CITY BANK	200	5	19	7	18	27.62
CITY BANK	277	5	4	10	18	31.05
DBHF	85	3	9	5	19	123
IFIC	147	2	2	8	17	17.77
IPDC	62	4	30	3	16	19
SQUARE	84	2	25	3	19	251.78
SQUARE	100	1	31	5	15	161.06
IFIC	45	5	5	3	15	20
RUPALI LIFE	162	2	27	8	17	37.4
DBHF	165	3	16	9	19	120.5
NAVANA	135	4	29	7	19	45.2

4.Results

This study focuses on predicting stock market prices in the evening. We utilized four algorithms to train the network and calculated the evening pricing. In this section, the results from these four algorithms are presented, and then they are briefly discussed. At first, individual result discussion of each algorithm is performed. Their regression value, error histogram is presented, and finally error testing, and accuracy testing is presented for each of them. We have carried out experiment on enormous configurations for each of the algorithm. Since, LM is our first experimented algorithm and in addition it provides the best accuracy result; therefore, we have presented a total of 5 experimented configurations (which shown better result) for this algorithm. For the rest of the three algorithms (i.e., BR, SCG, and QN); we have presented a total of 3 experimented configurations (which shown better result). At the end of this section, the performance comparison of accuracy among these four algorithms are presented and discussed.

4.1Levenberg Marquardt(LM)

After train the proposed ANN using LM algorithm, we get regression, error histogram and MSE value that measure the performance of proposed model.

With two hidden layer (6-5-5-1) and 70% training data

The regression plot depicts the relationship among the proposed system's outcomes and the targets. The plots depict data from training, validation, and testing. Each plot's dashed line represents the ideal result. The best-fit linear regression line among outputs and targets is represented by the solid line. R=1 denotes that outputs and targets have an exact linear relationship. There has no linear relationship among output and targets when R is nearly zero.

Figure 2 shows that, the training data indicates a good fit. Training regression value is 0.97671, validation regression value is 0.97612, and testing regression value is 0.97818 and. Combined regression value for these three is 0.97685.

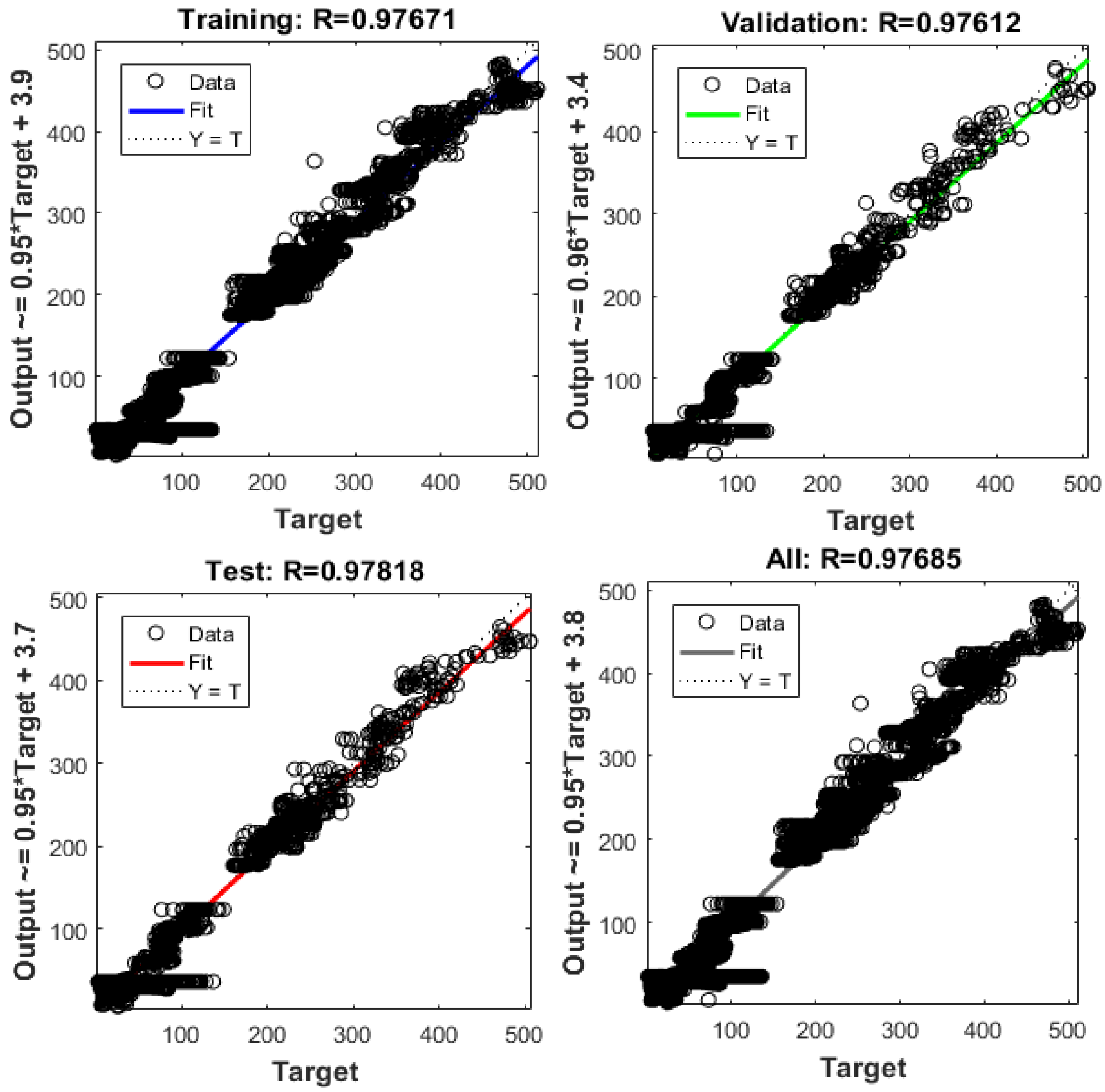

Figure 2 Regression value of two hidden layer (6-5-5-1) and 70% training data

Now, we discuss about the error histogram which is presented in *Figure 3*. The blue bar reflects training data, the green bar reflects validation data, and the red bar reflects testing data in the Error histogram. The histogram can reveal outliers, which are data points with a fit that is much worse than the rest of the data. The quantity of vertical lines visible on the graph is referred to as bins. Here, the overall error spectrum is broken down into 20 separate bins. The number of samples from the dataset that fall into each bin is represented on the Y-axis. In *Figure 3*, there is a bin related to an inaccuracy of 0.4895 in the middle of the graph, and the elevation of that bin for training sample is approximately 4000, while the elevation of that bin for validation as well as test datasets is among 5000 and 6000. It signifies that numerous samples from various collected sample sets have an inaccuracy in the range shown below. The zero-error plot corresponds to the error axis's zero error result. In this situation, the zero-error spot is contained within the bin with centered at 0.4895. In *Figure 3*, most errors fall between the range -31.16 to 32.13.

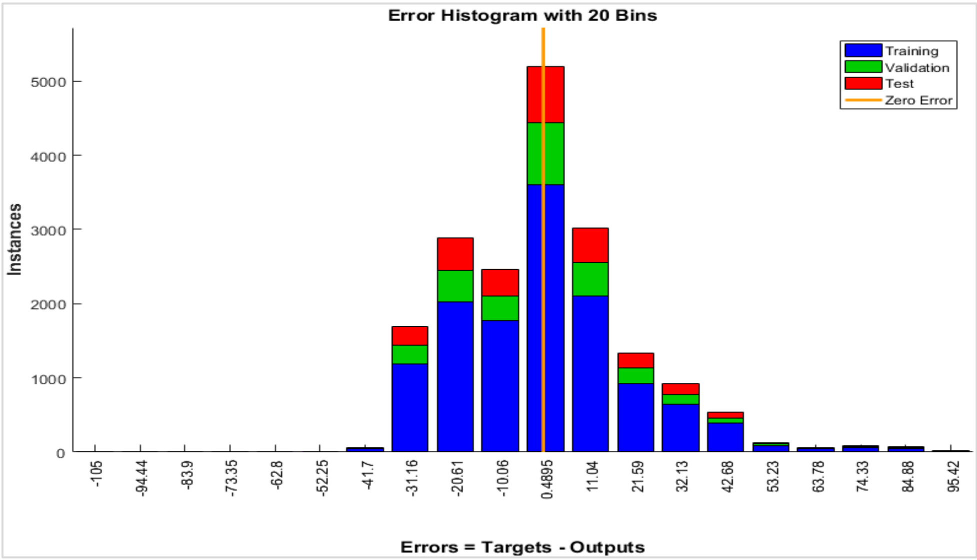

Figure 3 Error histogram of two hidden layer (6-5-5-1) and 70% training data

With one hidden layer (6-10-1) and 90% training data

Figure 4 shows that the training data indicates a good fit. Training regression value is 0.95317, validation regression value is 0.9539, and testing regression value is 0.94997.

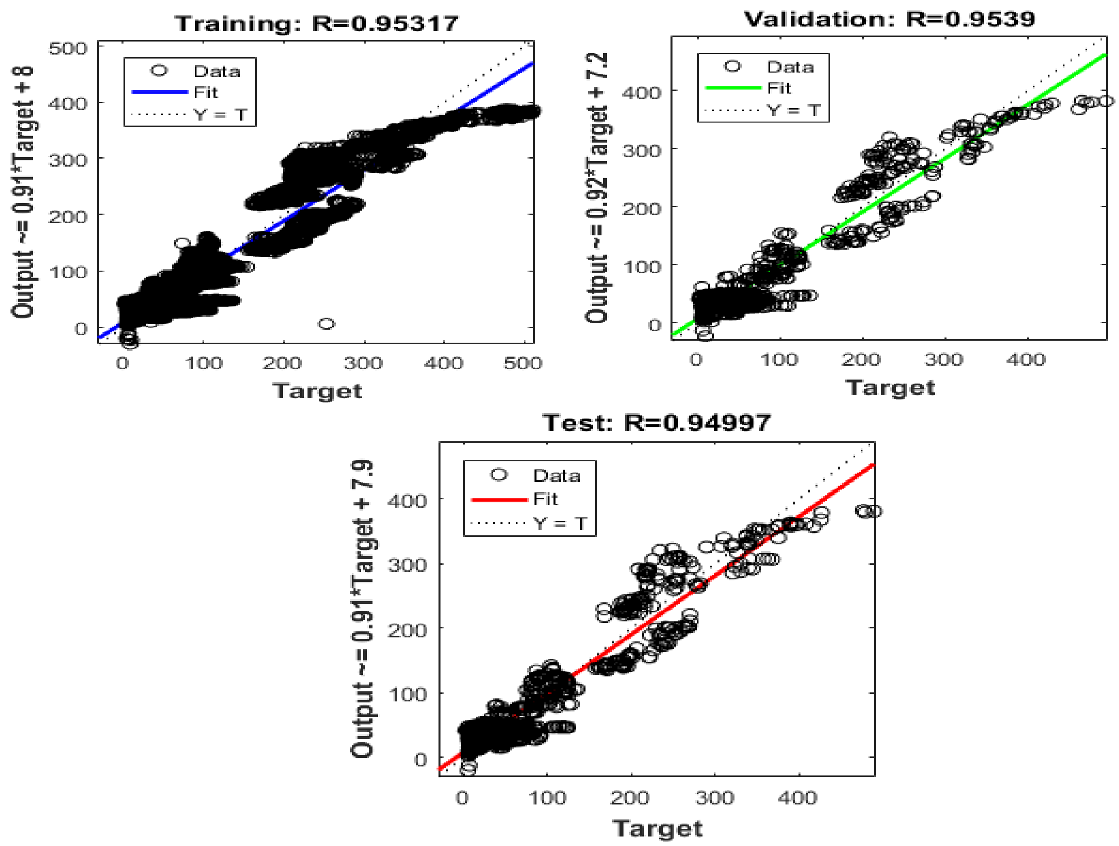

Figure 4 Regression value of two hidden layer (6-10-1) and 90% training data

Just like the above two configurations; we have performed the similar operations with three more configurations. These three configurations are: (i) With one hidden layer (6-10-1) and 70% training Data, (ii) With two hidden layer (6-10-5-1) and 70% training Data, and (iii) With two hidden layer (6-10-8-1) and 70% training data. For these three configurations, only the summarized value is taken as the generated figures would be identical to the detailed described configurations. The summary of the performance of LM algorithm for all the experimented configurations are presented in *Table 4*.

Table 4 shows that in LM algorithm, two hidden layer (6-10-8-1) and 70% training, 15% testing and 15% validation data given the higher regression value and lower error range.

Table 4 Summary of LM

Algorithm	Hidden layer	Hidden neuron	Training data	Regression value	Error range
	2	5-5	70%	0.9768	-31.16 to 32.13
Levenberg Marquardt (LM)	1	10	90%	0.9510	-32.94 to 51.99
	1	10	70%	0.9771	-28.94 to 27.75
	2	10-8	70%	0.9976	-5.79 to 5.26
	2	10-5	70%	0.9929	-4.98 to 6.50

Figure 5 shows a bin related to an inaccuracy of 1.031 in the middle of the graph, with an elevation of 4000 for training samples and 4500 to 5000 for validation as well as test datasets. Zero error line corresponding to the zero-error value on the error axis (i.e., X-axis). In this case zero error point falls under the bin with center 1.031. In this figure, most errors fall between -32.94 and 51.99.

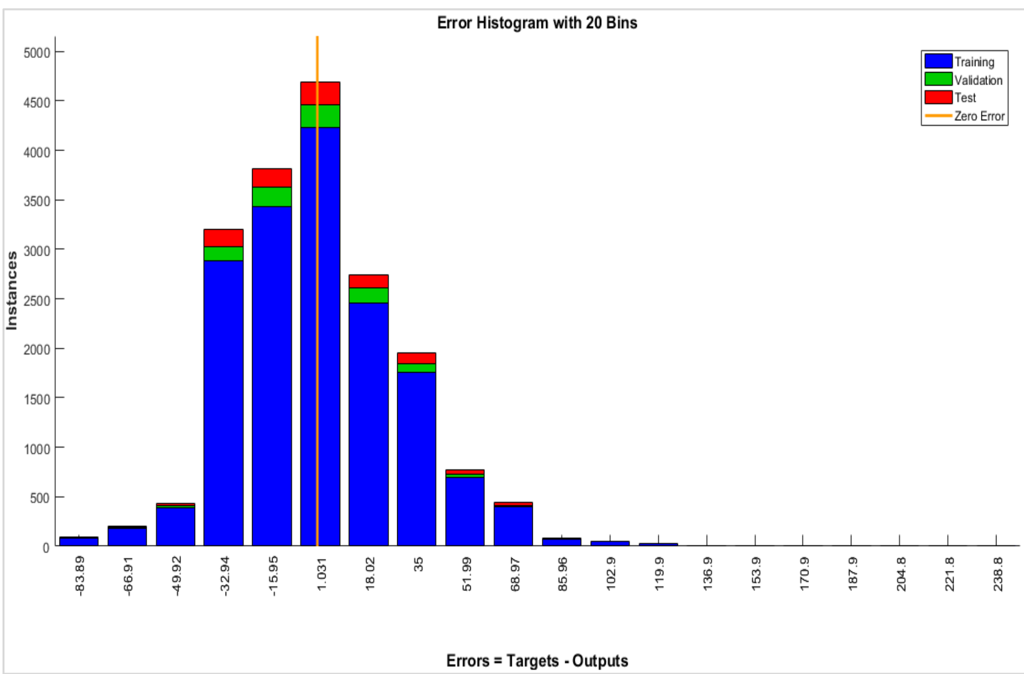

Figure 5 Error histogram of one hidden layer (6-10-1) and 90% training data

Table 5 shows the error and average error of LM training algorithm. Here targets price is actual value and predicted price is our proposed system's predicted value. Error is calculated from targets and predicted value, and then average them. *Table 5* shows the average error of LM with two hidden layer (6-10-8-1) and 70% training is 4.36, two hidden layer (6-10-5-1) and 70% training is 5.27 and one hidden layer (6-10-1) and 70% training data is 8.81. This work performs experiments with more configurations by changing hidden layer and training data but presents only best three configurations value at *Table 5*. From *Table 5,* we get the average error which is presented at the bottom of the table.

Table 6 shows the average error, and accuracy of LM algorithm for the best performing configurations. The accuracy of one hidden layer (6-10-1) with 70% training data is 91.19, two hidden layers (6-10-5-1) with 70% training data is 94.73, and two hidden layers (6-10-8-1) with 70% training data is 95.64. The last configuration has the best accuracy and lowest average error among them.

Table 5 Error testing of LM

Sample No.	Targets	Predicted	Error (%)	Predicted	Error (%)	Predicted	Error (%)
	-	For 6-10-1 and 70% training data	-	6-10-5-1 and 70% training data	-	For 6-10-8-1 and 70% training data	-
1	6.3	5.37	14.76	5.7	9.52	5.75	8.69
2	109.8	119.7	9.01	116.36	5.97	115.36	5.06
3	263.3	275.92	4.79	272.92	3.65	270.92	2.89
4	14.78	11.21	24.15	14.21	3.80	14.21	3.80
5	163.49	170.67	4.39	167.8	2.63	167.89	2.63
6	49.6	43.9	11.49	51.3	3.42	47.80	3.62
7	47.66	42.8	10.19	46.20	3.05	46.20	3.05
8	250.4	257.78	2.94	252.78	0.95	251.78	0.55
9	19.62	17.61	10.23	17.4	11.31	17.61	10.23
10	15.81	13.7	13.34	14.5	8.28	15.81	6.13
11	218.24	234.49	7.44	222.49	3.78	229.49	0.79
12	34.13	32.82	3.83	33.1	3.01	33.82	0.89
13	105	101.84	3.01	103.95	1	101.84	1.09
14	5.7	4.9	14.03	4.9	14.03	4.9	14.03
15	27.62	30.01	8.65	29.87	8.14	29.87	4.52
16	31.05	27.1	12.72	29	6.60	27.1	6.26
17	123	128.4	4.39	125.8	2.27	128.4	1.13
18	17.77	14.9	16.15	16.7	6.02	15.54	6.87
19	19	18.1	4.73	20.3	6.84	17.09	4.75
20	251.78	245.6	2.45	246.76	1.99	245.76	1.99
21	161.06	168.32	4.50	165.5	2.75	153.22	3.82
22	20	17.8	11	18.23	8.85	17.8	5.96
23	37.4	36.1	3.47	35.2	5.88	35.43	2.57
24	120.5	126.3	4.81	125.84	4.43	125.3	4.43
25	45.2	38.9	13.93	47.7	5.53	40.3	4.63
		Avg error	8.81	Avg error	5.27	Avg error	4.36

Table 6 Accuracy testing of LM

Hidden layer with neuron and training data	Average error (%)	Accuracy (%)
For 6-10-1 and 70% training data	8.81	91.19
For 6-10-5-1 and 70% training data	5.27	94.73
For 6-10-8-1 and 70% training data	4.36	95.64

4.2 Bayesian Regularization (BR)

With one hidden layer (6-10-1) and 70% training data

Figure 6 shows regression value. The training data indicates a good fit. Training regression value is 0.94206 and testing regression value is 0.94091. Combined regression value for these two is 0.942.

Figure 7 has a bin related to an inaccuracy of 2.184 in the middle of the graph, and the elevation of that bin for training sample is close to 6000, as well as the elevation of that bin for such testing samples is indeed above 6000. It signifies that numerous samples of various datasets have had an inaccuracy in the range shown below. Zero error line corresponding to the zero-error value on the error axis (i.e., X-axis).

In this case, zero error point falls under the bin with center 2.18. In *Figure 7*, maximum errors fall between -47.66 and 52.03.

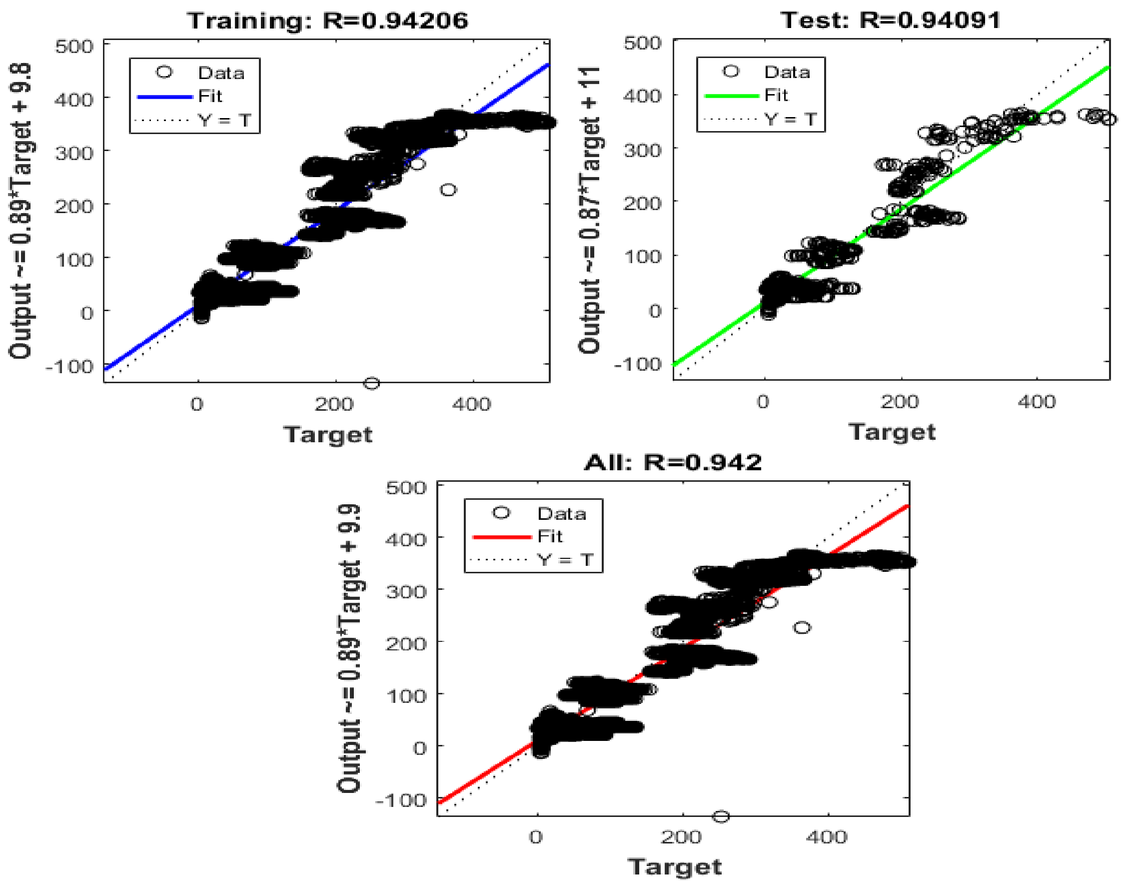

Figure 6 Regression value of one hidden layer (6-10-1) and 70% training data

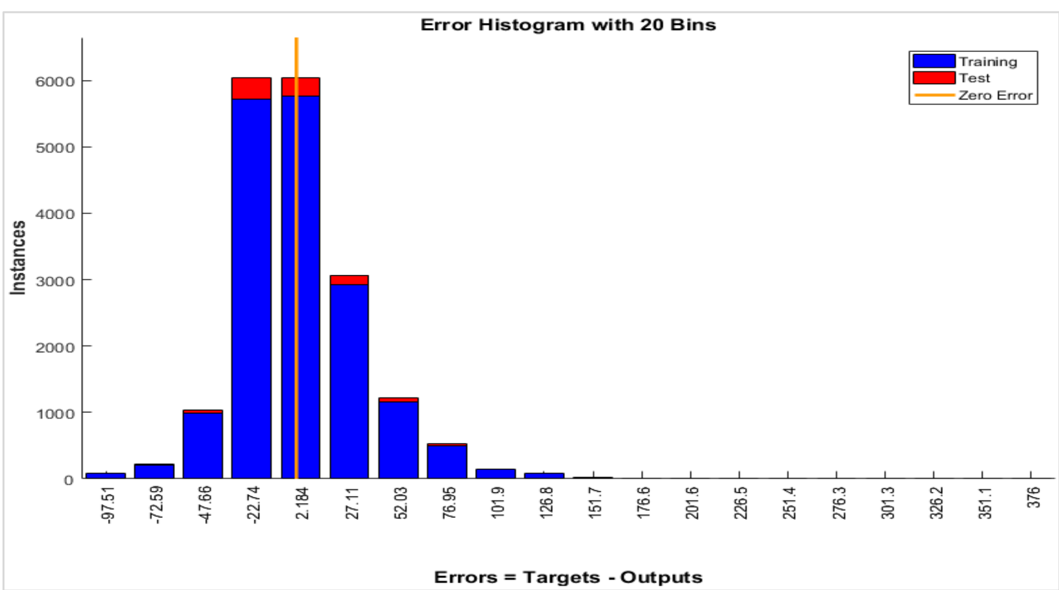

Figure 7 Error histogram of one hidden layer (6-10-1) and 70% training data

With two hidden layer (6-10-5-1) and 70% training data

Figure 8 shows regression value. The training data indicates a good fit. Training regression value is 0.96291 and testing regression value is 0.95781. Combined regression value for these two is 0.96214.

Figure 9 has a bin relating to the inaccuracy of -10.52 in the middle of the graph, with the elevation of that bin for training samples above but around 5000 as well as the elevation of that bin for such testing samples among 5000 and 6000. Zero error spot falls under center of bin between 10.52 and11.89. In *Figure 9*, maximum errors fall between -32.93 and 34.31.

Just like the above two configurations; we have performed the similar operations with one more configuration. This configuration is: With two hidden layer (6-10-8-1) and 70% training data. The summary of the performance of BR algorithm for all the experimented configurations are presented in *Table 7*.

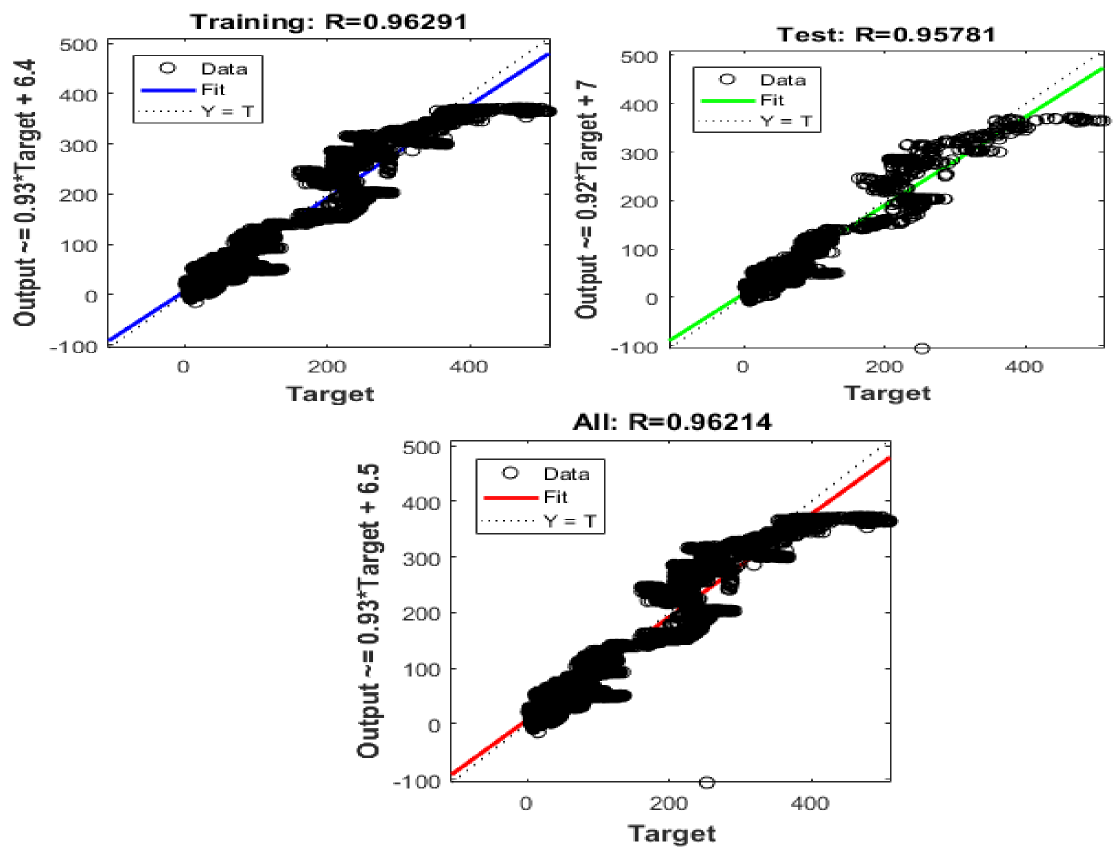

Figure 8 Regression value of two hidden layer (6-10-5-1) and 70% training data

Table 7 shows that, in BR algorithm; two layer (6-10-8-1) and 70% training, 15% testing and 15% validation data given the higher regression value and lower error range.

Table 8 shows the error and average error of BR training algorithm. This table shows that, the average error of BR with two hidden layer (6-10-8-1) and 70% training data is 8.73, with two hidden layer (6-10-5-1) and 70% training data is 11.44 and with one hidden layer (6-10-1) and 70% training data is 12.35. This work performs experiments with more configurations by changing hidden layer and training data but presents only best three configurations value at *Table 8*.

Table 9 shows the average error, and accuracy of BR algorithm for the best performing configurations. The accuracy of one hidden layer (6-10-1) with 70% training data is 87.65, two hidden layers (6-10-5-1) with 70% training data is 88.56 and two hidden layers (6-10-8-1) with 70% training data is 91.26. The configuration (6-10-8-1) with 70% training data has the best accuracy and lowest average error among them.

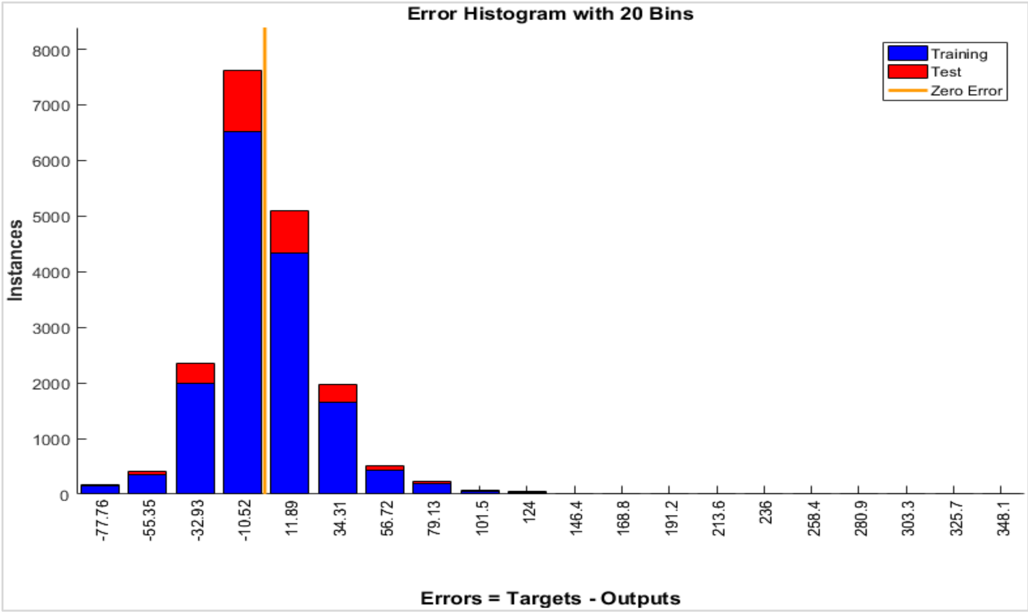

Figure 9 Error Histogram of two hidden layer (6-10-5-1) and 70% training data

Table 7 Summary of BR

Algorithm	Hidden layer	Hidden neuron	Training data	Regression value	Error range
	1	10	70%	0.942	-47.66 to 52.03
Bayesian Regularization (BR)	2	10-5	70%	0.96214	-32.93 and 34.31
	2	10-8	70%	0.979	-26.56 to 29.05

Table 8 Error testing of BR

Sample No	Targets	Predicted	Error (%)	Predicted	Error (%)	Predicted	Error (%)
	-	For 6-10-8-1 and 70% training data	-	For 6-10-5-1 and 70% training data	-	For 6-10-1 and 70% training data	-
1	6.3	5.3	15.87	5.1	19.04	5	20.63
2	109.8	119.7	9.01	117.7	7.19	101.7	7.37
3	263.3	277.92	5.55	149.91	5.08	247.5	6
4	14.78	11.21	24.15	11.21	24.15	11.21	24.15
5	163.49	170.67	4.39	173.67	6.22	178.67	9.28
6	49.6	45.12	9.03	44.12	11.04	44.23	10.82
7	47.66	43.11	9.54	43.11	9.54	43	9.77
8	250.4	251.78	0.55	252.9	0.99	240.3	3.79
9	19.62	17.61	10.23	16.5	15.90	16.9	13.86
10	15.81	13.83	12.48	12.7	19.67	12.6	20.30
11	218.24	234.49	7.44	205.49	5.84	233.48	6.98
12	34.13	32.82	3.83	29.9	12.39	29.89	12.39
13	105	101.84	3.0	98.84	5.86	112.84	7.4
14	5.7	4.9	14.03	6.5	14.03	4.9	14.03
15	27.62	29.87	8.14	30.85	11.69	30.69	11.11
16	31.05	27.1	12.72	27.5	11.43	27.2	12.39
17	123	128.4	4.39	135.4	10.08	135.4	10.08

Sample No	Targets	Predicted	Error (%)	Predicted	Error (%)	Predicted	Error (%)
	-	For 6-10-8-1 and 70% training data	-	For 6-10-5-1 and 70% training data	-	For 6-10-1 and 70% training data	-
18	17.77	15.01	15.53	13.9	21.77	12.9	27.40
19	19	17.09	10.05	17.3	8.94	21.4	12.63
10	251.78	245.76	2.39	278.96	10.08	220.96	12.24
21	161.06	153.22	4.86	149.7	7.05	147.7	8.29
22	20	17.8	11	16	20	16.3	18.5
23	37.4	35.43	5.26	35.1	6.14	34.1	8.82
24	120.5	125.3	3.98	129.7	7.63	111.6	7.30
25	45.2	40.3	10.84	51.3	13.49	51.2	13.27
		Avg error	8.73	Avg error	11.44	Avg error	12.35

Table 9 Accuracy testing of BR

Hidden layer with neuron and training data	Average error (%)	Accuracy (%)
For 6-10-1 and 80% training data	12.35	87.65
For 6-10-5-1 and 70% training data	11.44	88.56
For 6-10-8-1 and 70% training data	8.73	91.26

4.3 Scaled conjugate gradient (SCG)

With two hidden layer (6-5-5-1) and 70% training data

Figure 10 shows regression value. The training data indicates a good fit. Training regression value is 0.91009, validation regression value is 0.90558, and testing regression value is 0.91468.

Figure 11 has a bin relating to the inaccuracy of -23.29 in the middle of the graph, with the elevation of that bin for training samples above but around 3000 as well as the elevation of that bin for such testing samples among 4000 and 4500. Zero-error spot falls under the bin 5.72. Figure 11 shows that, the maximum error fall between -40.86.78 to 46.99.
With one hidden layer (6-10-1) and 80% training data

Figure 12 shows the regression value. The training data indicates a good fit. Training regression value is 0.91335, validation regression value is 0.91211, and testing regression value is 0.91315. Combined regression value for these three is 0.91318. *Figure 13* has a bin relating to the inaccuracy of -10.36 in the middle of the graph, with the elevation of that bin for training samples above but around 3000 as well as the elevation of that bin for such testing samples among 3500 and 4000. Zero-error spot falls under the center bin between -10.3 and 8.7. *Figure 13* shows that the most of the errors fall between -48.53 and 65.97.

Just like the above two configurations; we have performed the similar operations with one more configuration. This configuration is: With two hidden layer (6-10-8-1) and 70% training data. The summary of the performance of SCG algorithm for all the experimented configurations are presented in *Table 10*. *Table 10* shows that, in SCG algorithm, two hidden layer (6-10-8-1) and 70% training, 15% testing and 15% validation data set given the higher regression value and comparative lower error range.

Table 11 shows the error and average error of SCG algorithm. This table shows that, the average error of SCG with two hidden layer (6-10-5-1) and 70% training is 13.67, with two hidden layer (6-10-8-1) and 70% training is 11.09, and with one hidden layer (6-10-1) and 70% training data is 13.81. This work performs experiments with more configurations by changing hidden layer and training data but presents only best three configurations value at *Table 11*.

Table 12 shows the average error, and accuracy of SCG algorithm. The accuracy of two hidden layer (6-5-5-1) with 70% training data is 86.19, with one hidden layer (6-10-1) with 70% training data is 86.33, and with two hidden layers (6-10-8-1) with 70% training data is 88.91. The last configuration delivers the best accuracy with lowest average error of SCG algorithm.

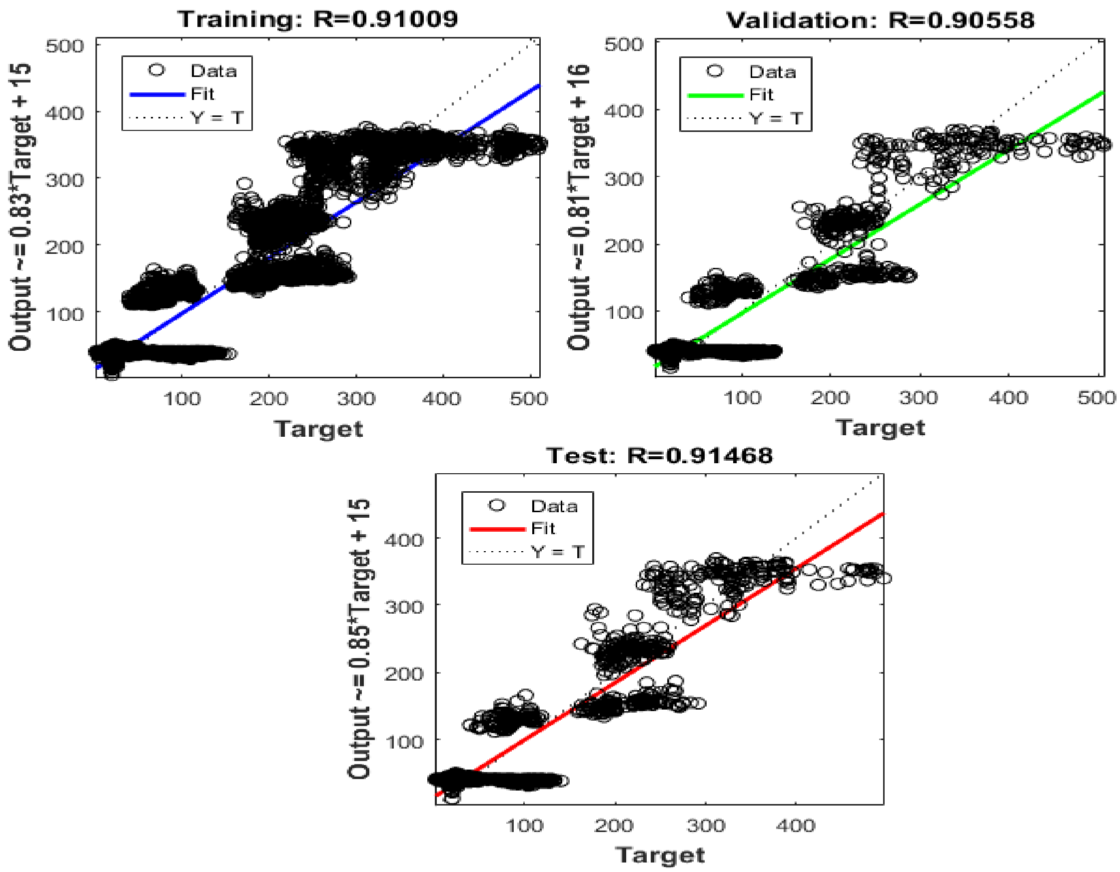

Figure 10 Regression value of two hidden layer (6-5-5-1) and 70% training data

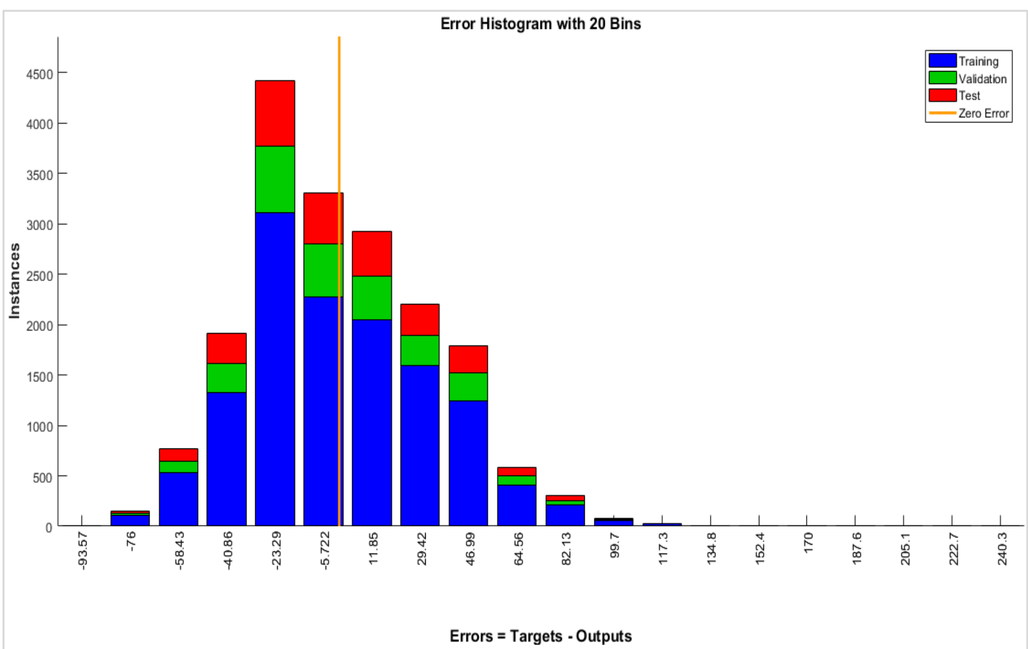

Figure 11 Error histogram of two hidden layer (6-5-5-1) and 70% training data

Table 10 Summary of SCG algorithm

Algorithm	**Hidden layer**	**Hidden neuron**	**Training data**	**Regression value**	**Error range**
	2	10-5	70%	0.9101	-40.86 to 46.99
Scale Conjugate Gradient	1	10	80%	0.91318	-48.53 and 65.97
	2	10-8	70%	0.9468	-31.78 to 31.45.

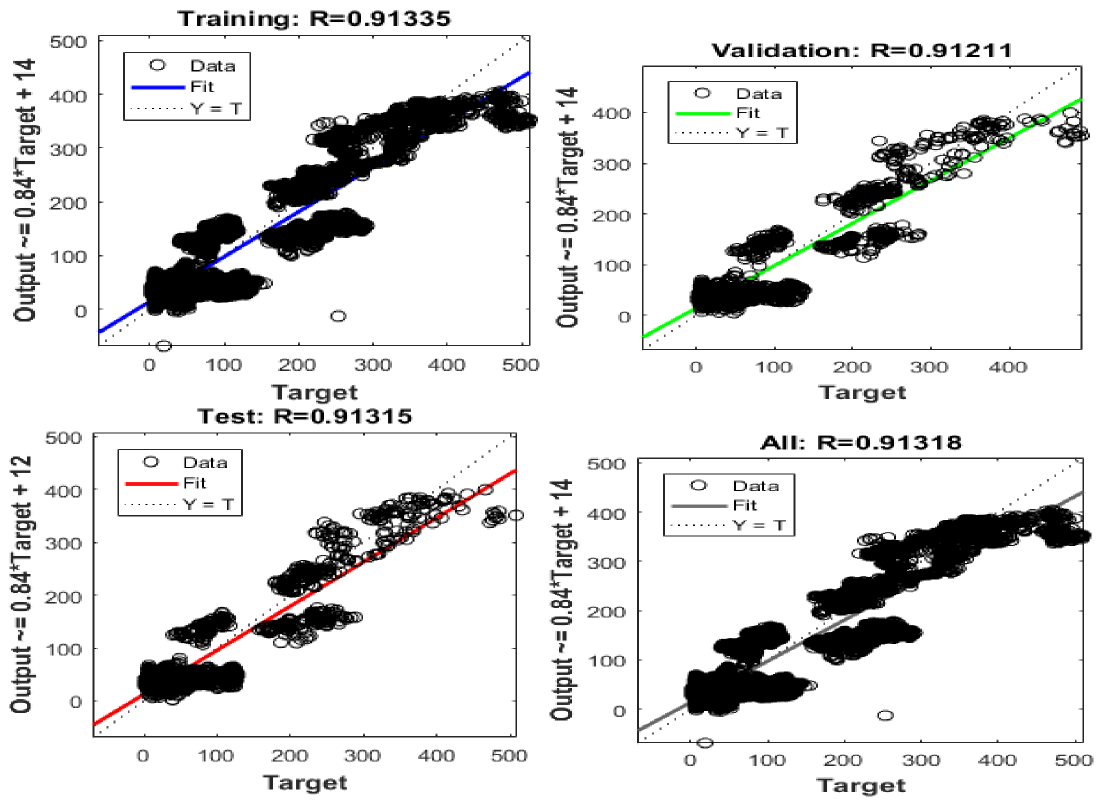

Figure 12 Regression value of two hidden layer (6-10-1) and 80% training data

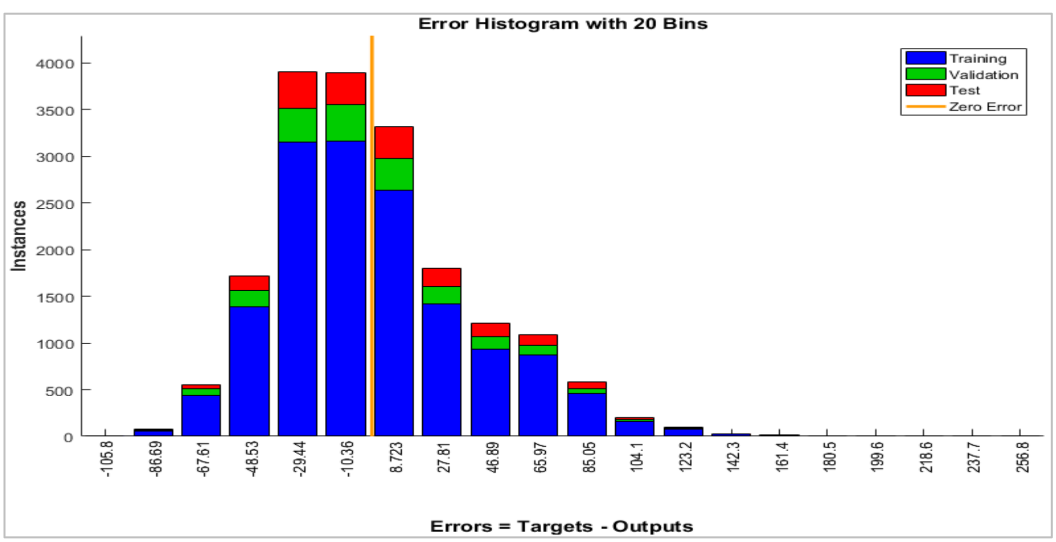

Figure 13 Error histogram of two hidden layer (6-10-1) and 80% training data

Table 11 Error testing of SCG

Sample No.	Targets	Predicted	Error (%)	Predicted	Error (%)	Predicted	Error (%)
-	-	For 6-10-5-1 and 70% training data	-	6-10-8-1 and 70% training data	-	For 6-10-5-1 and 80% training data	-
1	6.3	4.9	22.22	5.1	19.04	7.89	25.23
2	109.8	120.7	9.92	117.7	7.19	100.7	8.28
3	263.3	291.92	10.86	277.92	5.55	230.92	12.29
4	14.78	10.9	26.25	11.21	24.15	10.9	26.25
5	163.49	193.67	18.45	173.67	6.22	195.67	19.68
6	49.6	44.21	10.86	44.12	11.04	44.5	10.28
7	47.66	52.11	9.33	43.11	9.54	44.11	7.44
8	250.4	231.9	7.38	252.9	0.99	230.1	8.10
9	19.62	16.2	17.43	16.5	15.90	16.2	17.43
10	15.81	18.7	18.27	12.7	19.67	12.9	18.40
11	218.24	240.49	10.19	238.49	9.27	244.49	12.02
12	34.13	30.7	10.04	31.1	8.87	30.9	9.46
13	105	110.7	5.42	100.84	3.96	110	4.76
14	5.7	4.8	15.78	4.9	14.03	4.8	15.78
15	27.62	34.85	26.17	31.85	15.31	34.85	26.17
16	31.05	27.1	12.72	27.1	12.72	27.1	12.72
17	123	111.4	9.43	131.4	6.82	111.4	9.43
18	17.77	23.55	32.52	13.55	23.74	22.55	26.89
19	19	16.7	12.10	17.10	10	16.69	12.10
20	251.78	241.96	3.90	241.96	3.90	231.96	7.87
21	161.06	152.5	5.31	153.7	4.56	173.5	7.72
22	20	15.8	21	15.8	21	24.2	21
23	37.4	34.2	8.55	35.2	5.88	34.15	8.68
24	120.5	112.7	6.47	128.7	6.80	129	7.05
25	45.2	50.3	11.28	40.3	10.84	40.6	10.17
		Avg error	13.67	Avg error	11.09	Avg error	13.81

Table 12 Accuracy testing of SCG

Hidden layer with neuron and training data	Average error (%)	Accuracy (%)
For 6-5-5-1 and 70% training data	13.81	86.19
For 6-10-1 and 80% training data	13.67	86.33
For 6-10-8-1 and 70% training data	11.09	88.91

4.4 Quasi Newton

With one hidden layer (6-10-1) and 70% training data

Figure 14 shows regression value of QN algorithm. The training data indicates not a good fit. Training regression value is 0.69919, validation regression value is 0.68781, and testing regression value is 0.68511. The combined regression value of these three is 0.69555.

Figure 15 has a bin relating to the inaccuracy of 9.287 in the middle of the graph, with the elevation of that bin for training samples above but around 3000 as well as the elevation of that bin for such testing samples among 5000 and 6000. Zero-error spot falls under the bin 9.27. *Figure 15* shows that, the most errors fall between -106.4 and 67.13.

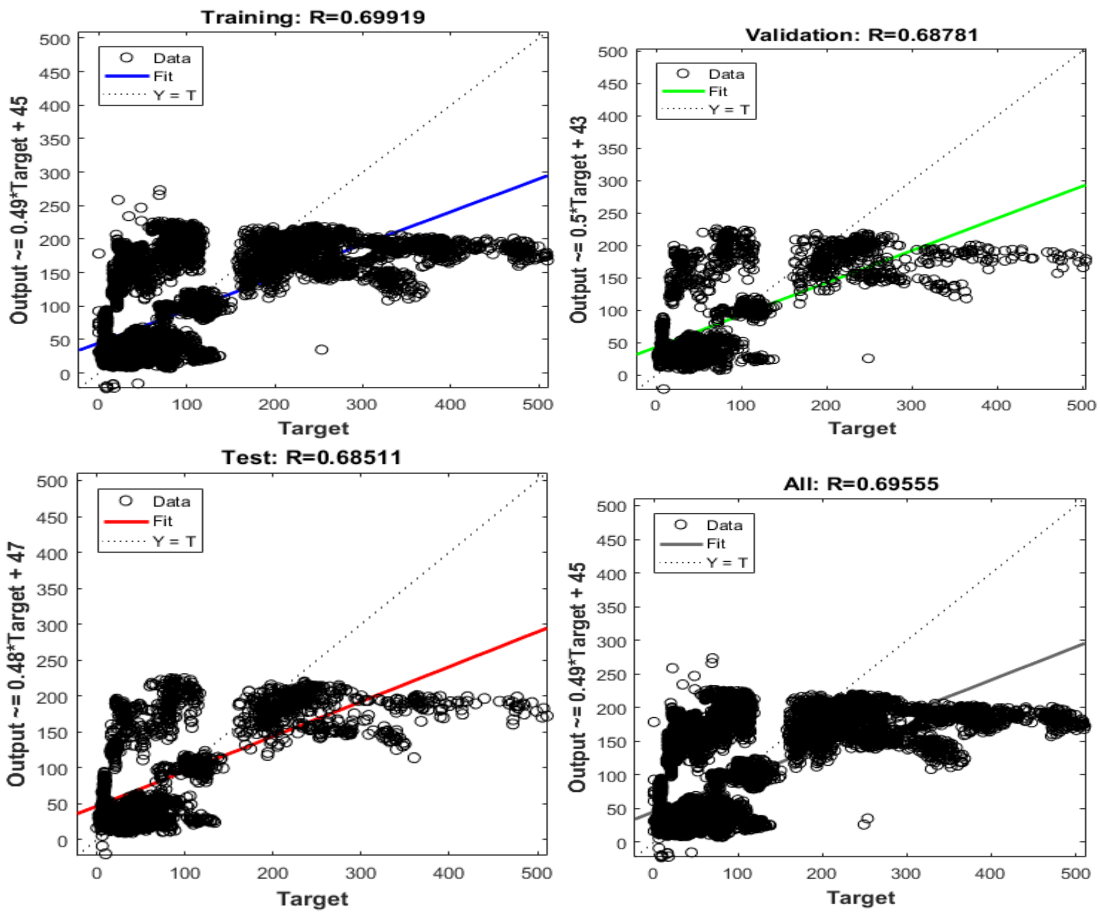

Figure 14 Regression value of one hidden layer (6-10-1) and 70% training data

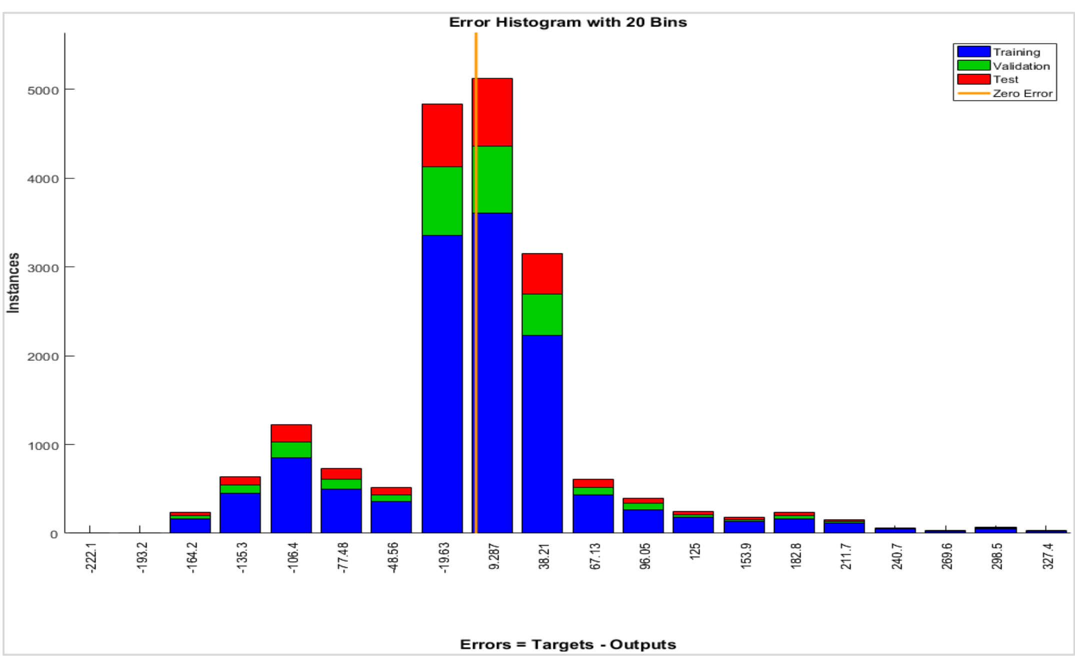

Figure 15 Error histogram of one hidden layer (6-10-1) and 70% training data

With two hidden layer (6-10-5-1) and 70% training data

Figure 16 shows the regression value. The training data indicates not a good fit. Training regression value is 0.82372, validation regression value is 0. 81819, and testing regression value is 0.80701. The combined regression value of these three is 0.82047.

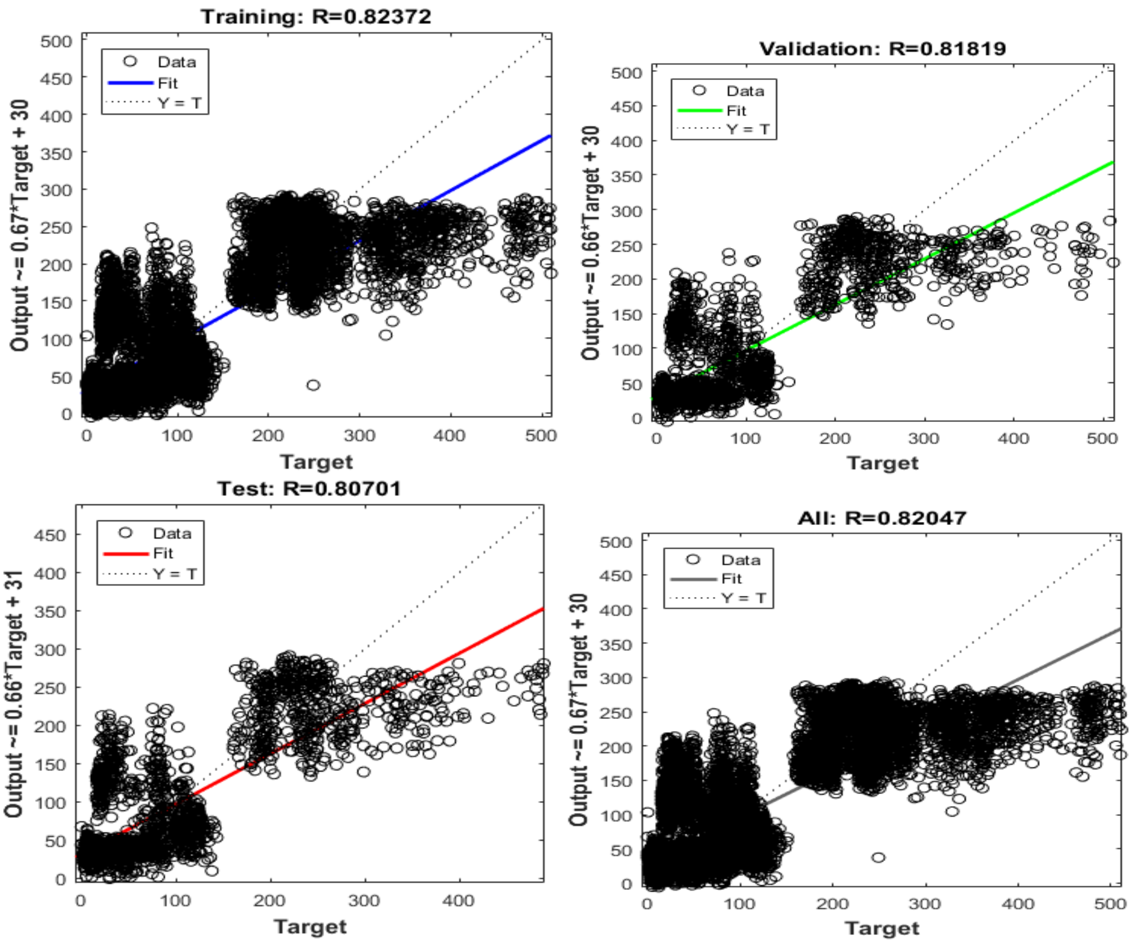

Figure 16 Regression value of two hidden layer (6-10-5-1) and 70% training data

Figure 17 has a bin relating to the inaccuracy of -34.57 in the middle of the graph, with the elevation of that bin for training samples above but around 3000 as well as the elevation of that bin for such testing samples among 4500 and 5000. Zero error spot falls under the bin with center 1.892. *Figure 17* shows that, the maximum errors fall between -74.55 and 85.36. Just like the above two configurations; we have performed the similar operations with one more configuration. This configuration is: With two hidden layer (6-10-8-1) and 70% training data. The summary of the performance of QN algorithm for all the experimented configurations are presented in *Table 13*. *Table 13* shows that, in QN algorithm, two hidden layer (6-10-8-1) and 70% training, 15% testing and 15% validation data set given the higher regression value and lower error range.

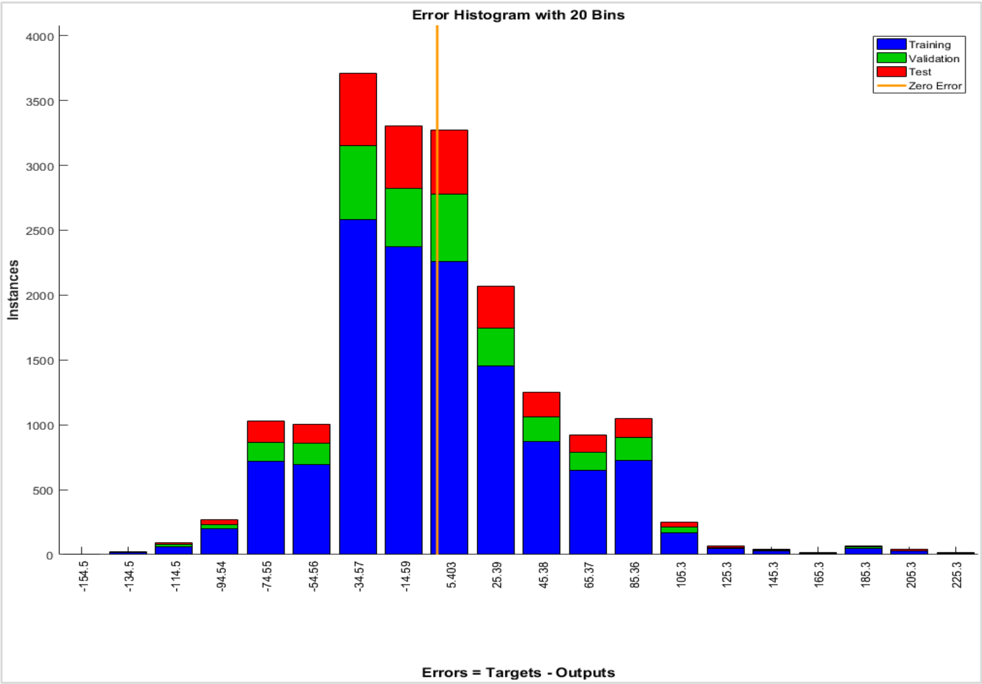

Figure 17 Error histogram of two hidden layer (6-10-5-1) and 70% training data

Table 13 Summary of QN

Algorithm	Hidden layer	Hidden neuron	Training data	Regression value	Error range
	1	10	70%	0.69555	-106.4 and 67.13
Quasi Newton	2	10-5	70%	0.82047	-74.55 and 85.36
	2	10-8	70%	0.867	-74.78 to 85.35.

Table 14 shows the error and average error of QN training. This table shows that, the average error of QN with one hidden layer (6-10-1) and 70% training data is 25.97, with two hidden layer (6-10-8-1) and 70% training is 15.8, and with two hidden layer (6-10-5-1) and 70% training is 18.22. This work performs experiments with more configurations by changing hidden layer and training data but presents only best three configurations value at *Table 14.*

Table 15 shows the average error, and accuracy of QN training algorithm. The accuracy of one hidden layer (6-10-1) with 70% training data is 74.03, two hidden layers (6-10-5-1) with 70% training data is 81.78, and two hidden layers (6-10-8-1) with 70% training data is 84.20 and this is the best accuracy of QN.

Table 14 Error testing of QN

Sample No.	Targets	Predicted	Error (%)	Predicted	Error (%)	Predicted	Error (%)
-	-	For 6-10-1 and 70% training data	-	For 6-10-8-1 and 70% training data	-	For 6-10-5-1 and 70% training data	-
1	6.3	8.9	41.26	5	20.63	4.8	23.80
2	109.8	90.12	17.92	125.12	13.95	130.12	18.50
3	263.3	357.92	35.93	297.92	13.15	309.92	17.70
4	14.78	9.45	36.06	10.45	29.29	10.45	29.29
5	163.49	193.3	18.23	140.3.67	14.18	185.3	13.34
6	49.6	38.8	21.77	55.13	11.15	41.8	15.72
7	47.66	38.1	20.05	41.09	13.74	39.1	17.96
8	250.4	277.78	10.93	255.78	2.16	260.8	4.16

Sample No.	Targets	Predicted	Error (%)	Predicted	Error (%)	Predicted	Error (%)
-	-	For 6-10-1 and 70% training data	-	For 6-10-8-1 and 70% training data	-	For 6-10-5-1 and 70% training data	-
9	19.62	13	33.74	15.1	22.93	15	23.54
10	15.81	23.2	46.74	11.3	28.52	11.2	29.15
11	218.24	274.49	25.77	260.5	19.35	267.49	22.56
12	34.13	25.2	26.16	27.19	20.33	27.2	20.30
13	105	130.84	24.60	99.79	4.91	110.84	5.56
14	5.7	3.99	30	4.21	26.31	4.02	29.47
15	27.62	37.25	34.86	30.9	11.87	31.25	13.14
16	31.05	22.05	28.98	26.1	15.72	26.05	16.10
17	123	148.4	20.65	135.4	10.08	135.4	10.08
18	17.77	12.5	29.65	13.01	15.53	13.2	25.71
19	19	23.09	21.52	17.09	10.05	21.09	11
20	251.78	210.76	16.29	215.76	14.30	216.76	13.90
21	161.06	129.22	19.76	135.22	16.04	132.22	17.9
22	20	14.3	28.5	15.7	21.3	25.8	29
23	37.4	44.43	18.79	35.2	10.61	43.3	16.12
24	120.5	101.3	15.93	135.7	16.59	105.3	12.61
25	45.2	33.8	25.22	38.3	16.59	53.8	19.02
		Average error	25.97	Average error	15.8	Average error	18.22

Table 15 Accuracy testing of QN

Hidden layer with neuron and training data	Average error (%)	Accuracy (%)
For 6-10-1 and 70% training data	25.97	74.03
For 6-10-5-1 and 70% training data	18.22	81.78
For 6-10-8-1 and 70% training data	15.8	84.20

5. Discussion

5.1 Analysis of result

For measuring the performance of our four algorithms, we have utilized regression value, error histogram, and accuracy as metrices.

Figure 18 shows the regression dataset of LM, BR, SCG, and QN. The regression value of LM is 0.99769, BR is 0.97962, SCG is 0.94681 and QN is 0.86725. This shows that the LM has the highest regression value.

Figure 19 shows the comparison of error histogram of the experimented algorithm. In LM, the maximum error lies in -5.79 to 5.26. In BR, the maximum error lies in -26.56 to 15.15. In SCG, the maximum error lies in -31.78 to 31.45 and in QN, the maximum error lies in -49.48 to 53.27. From above values, we can see that the less error has been found from LM.

Table 16 shows the best achieved accuracy for each of the algorithms. Here, for each algorithm; only the best accuracy value among several experimented (which is presented in above description) value is taken.

From the above result and discussions; it’s found out that, the LM has highest regression value, and minimum error range. In addition, LM provides better accuracy with lowest error than the other algorithms on the stock market share price prediction. In this work, the split ratio of training, testing and validation data sets is 70:15:15, and the resultant output is found that, LM training algorithm delivers the best performance. Therefore, using this works proposed process, investors can easily understand about the behavior of financial stock market and can easily make their prediction.

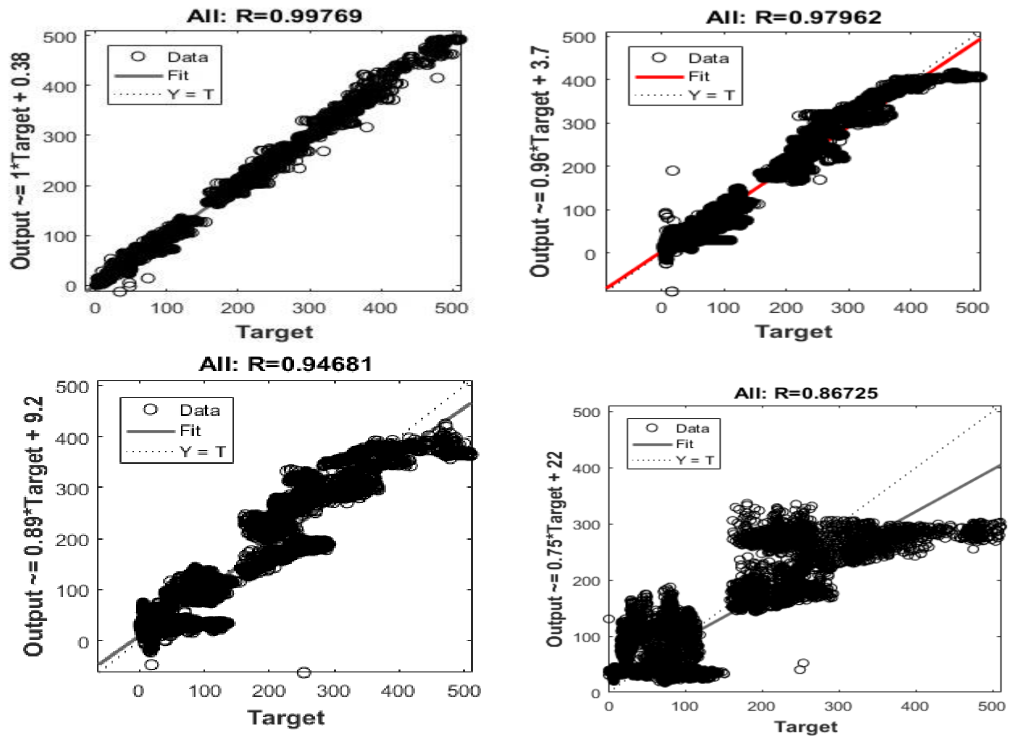

Figure 18 Comparison of regression values

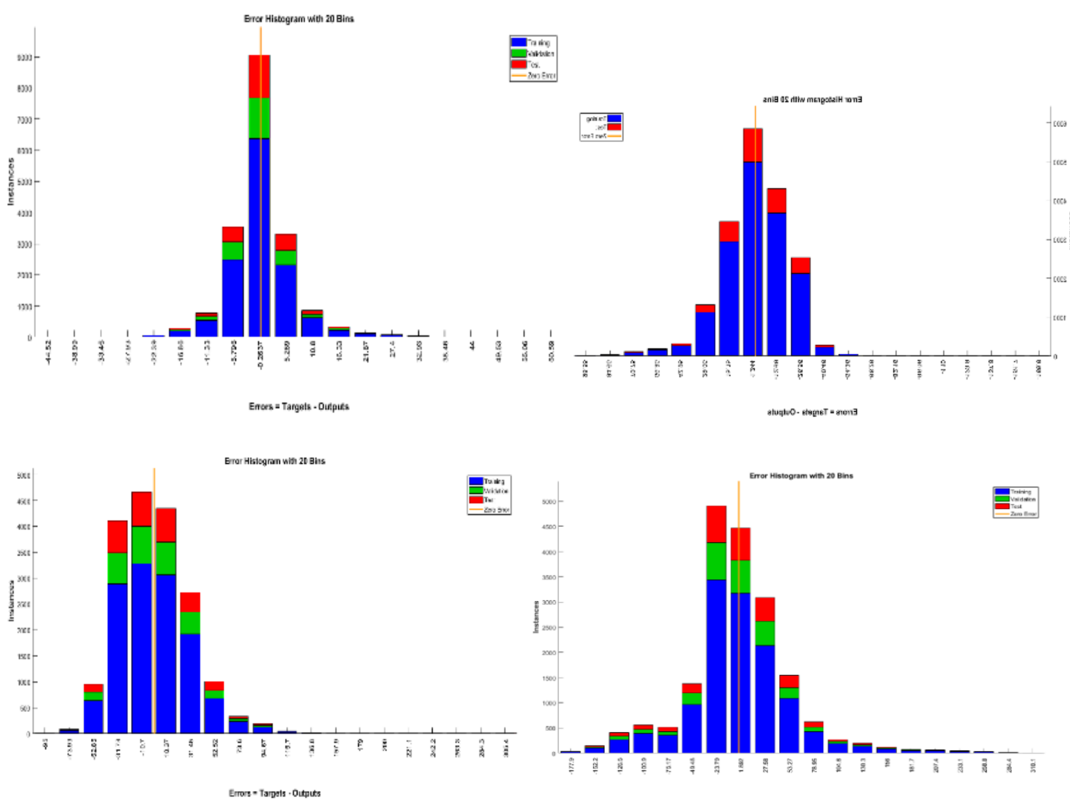

Figure 19 Comparison of error histogram

Table 16 Accuracy and lowest average error comparison

Algorithm	Configuration	Training data	Avg. error	Best accuracy
LM	6-10-8-1	70%	4.36	95.64
BR	6-10-8-1	70%	8.73	91.26
SCG	6-10-8-1	70%	11.09	88.91
QN	6-10-8-1	70%	15.80	84.20

5.1 Limitations

An ANN approach is used by our proposed model to make predictions. Even if the proposed model does have a better prediction performance than many other earlier and contemporary approaches, the prediction accuracy still seems to be not that significantly higher. Additionally, a lower accuracy rate is discovered for some particular data sets. This means that it might not draw all types of stock market investors unless the prediction accuracy for all data sets reaches 98-99% or above.

A complete list of abbreviations is shown in *Appendix I.*

6. Conclusion

Though it is impossible to estimate the actual price of a share in the future; a projected price that is near to the actual value can be highly valuable for a trader looking to earn a profit on the share market. Therefore, the major goal of this research is to accurately anticipate future stock prices. For this, we proposed a model dependent on ANN. In our proposed model, any regular data from the share market can easily be trained, checked, and then this taught network can be utilized to forecast stock prices. To train the obtained data, we used four training algorithms (LM, BR, SCG, and QN) in our research. All of the experiments in this research are based on real data, and the experiments are conducted using MATLAB software. We have collected and evaluated six years of DSE data for the top fifteen firms and found that the LM algorithm provides the best accuracy and least error. It is evident from the results of our research that, stock market prices can be forecast, and the accuracy rate indicates that it may be a very valuable tool for those who involved in the stock market.

Any solution must constantly be improved if it is to stay relevant in the tough competition. As a result, future adjustments to our work are possible. In the future, additional ANN algorithms may be taken into consideration for prediction to determine if they provide more accuracy. Additionally, new data from organizations could be incorporated to improve the prediction.

Acknowledgment

We are thankful to the persons working in the Laboratory of Information and Communication Engineering Department, Noakhali Science and Technology University, who have always helped us complete this work successfully. Finally, but not least, we are very much thankful to the Research Cell, Noakhali Science and Technology University, for her small financial support.

Conflicts of interest

The authors have no conflicts of interest to declare.

Author's contribution statement

Md. Ashikur Rahman Khan: Conceptualization, investigation, data curation, writing – original draft, writing – review and editing. **Md. Furkan Uzzaman:** Data collection, conceptualization, writing – original draft, model training, analysis and interpretation of results. **Ishtiaq Ahammad and Ratul Prosad:** Draft manuscript preparation, writing – review and editing. **Zayed-Us-Salehin, Tanvir Zaman Khan, Md. Sabbir Ejaz and Main Uddin:** Study conception, design, data collection, supervision and investigation on challenges.

References

[1] Hiransha M, Gopalakrishnan EA, Menon VK, Soman KP. NSE stock market prediction using deep-learning models. Procedia Computer Science. 2018; 132:1351-62.

[2] https://www.statista.com/statistics/274490/global-value-of-share-holdings-since 2000/#:~:text=This%20statistic%20presents%20the%20global,U.S.%20dollars%20in%20H1%202021. Accessed 18 September 2022.

[3] https://www.ceicdata.com/en/indicator/bangladesh/market-capitalization. Accessed 18 September 2022.

[4] Upadhyay A, Bandyopadhyay G, Dutta A. Forecasting stock performance in indian market using multinomial logistic regression. Journal of Business Studies Quarterly. 2012; 3(3):16-39.

[5] Tan TZ, Quek C, Ng GS. Biological brain- inspired genetic complementary learning for stock market and bank failure prediction. Computational Intelligence. 2007; 23(2):236-61.

[6] Ahammad I, Khan MA, Salehin ZU. Advancement of IoT system QoS by integrating cloud, fog, roof, and dew computing assisted by SDN: basic framework architecture and simulation. International Journal of Ambient Computing and Intelligence. 2021; 12(4):132-53.

[7] Nayak A, Pai MM, Pai RM. Prediction models for Indian stock market. Procedia Computer Science. 2016; 89:441-9.

[8] Falat L, Pancikova L. Quantitative modelling in economics with advanced artificial neural networks. Procedia Economics and Finance. 2015; 34:194-201.

[9] Ferreira FG, Gandomi AH, Cardoso RT. Artificial intelligence applied to stock market trading: a review. IEEE Access. 2021; 9:30898-917.

[10] Naeini MP, Taremian H, Hashemi HB. Stock market value prediction using neural networks. In 2010 international conference on computer information systems and industrial management applications 2010 (pp. 132-6). IEEE.

[11] Grudnitski G, Osburn L. Forecasting S&P and gold futures prices: an application of neural networks. Journal of Futures Markets. 1993; 13(6):631-43.

[12] http://www.dse.com. Accessed 18 September 2022.

[13] https://www.investing.com/. Accessed 18 September 2022.

[14] Ahammad I, Khan AR, Salehin ZU. A review on cloud, fog, roof, and dew computing: IoT perspective. International Journal of Cloud Applications and Computing. 2021; 11(4):14-41.

[15] Soni P, Tewari Y, Krishnan D. Machine learning approaches in stock price prediction: a systematic review. In journal of physics: conference series 2022 (pp. 1-10). IOP Publishing.

[16] Rouf N, Malik MB, Arif T, Sharma S, Singh S, Aich S, et al. Stock market prediction using machine learning techniques: a decade survey on methodologies, recent developments, and future directions. Electronics. 2021; 10(21):1-25.

[17] Strader TJ, Rozycki JJ, Root TH, Huang YH. Machine learning stock market prediction studies: review and research directions. Journal of International Technology and Information Management. 2020; 28(4):63-83.

[18] Ketsetsis AP, Kourounis C, Spanos G, Giannoutakis KM, Pavlidis P, Vazakidis D, et al. Deep learning techniques for stock market prediction in the European union: a systematic review. In international conference on computational science and computational intelligence 2020 (pp. 605-10). IEEE.

[19] Patel R, Choudhary V, Saxena D, Singh AK. Review of stock prediction using machine learning techniques. In international conference on trends in electronics and informatics 2021 (pp. 840-6). IEEE.

[20] Vadlamudi S. Stock market prediction using machine learning: a systematic literature review. American Journal of Trade and Policy. 2017; 4(3):123-8.

[21] Pyo S, Lee J, Cha M, Jang H. Predictability of machine learning techniques to forecast the trends of market index prices: hypothesis testing for the Korean stock markets. PloS One. 2017; 12(11).

[22] Anwar M, Rahman S. Forecasting stock market prices using advanced tools of machine learning (Doctoral Dissertation, Brac University).

[23] Reddy VK. Stock market prediction using machine learning. International Research Journal of Engineering and Technology. 2018; 5(10):1033-5.

[24] Chen K, Zhou Y, Dai F. A LSTM-based method for stock returns prediction: a case study of China stock market. In international conference on big data (pp. 2823-4). IEEE.

[25] Yetis Y, Kaplan H, Jamshidi M. Stock market prediction by using artificial neural network. In world automation congress 2014 (pp. 718-22). IEEE.

[26] Selvamuthu D, Kumar V, Mishra A. Indian stock market prediction using artificial neural networks on tick data. Financial Innovation. 2019; 5(1):1-12.

[27] Prastyo A, Junaedi D, Sulistiyo MD. Stock price forecasting using artificial neural network:(Case Study: PT. Telkom Indonesia). In international conference on information and communication technology 2017 (pp. 1-6). IEEE.

[28] Tsai CF, Wang SP. Stock price forecasting by hybrid machine learning techniques. In proceedings of the international multiconference of engineers and computer scientists 2009.

[29] Vijh M, Chandola D, Tikkiwal VA, Kumar A. Stock closing price prediction using machine learning techniques. Procedia Computer Science. 2020; 167:599-606.

[30] Kannan KS, Sekar PS, Sathik MM, Arumugam P. Financial stock market forecast using data mining techniques. In proceedings of the international multiconference of engineers and computer scientists 2010 (pp. 1-5).

[31] Wang H, Wang J, Cao L, Li Y, Sun Q, Wang J. A stock closing price prediction model based on CNN-bislstm. Complexity. 2021.

[32] Milosevic N. Equity forecast: predicting long term stock price movement using machine learning. arXiv preprint arXiv:1603.00751. 2016.

[33] Shen J, Shafiq MO. Short-term stock market price trend prediction using a comprehensive deep learning system. Journal of Big Data. 2020; 7(1):1-33.

[34] Shen S, Jiang H, Zhang T. Stock market forecasting using machine learning algorithms. Department of Electrical Engineering, Stanford University, Stanford, CA. 2012:1-5.

[35] Fischer T, Krauss C. Deep learning with long short-term memory networks for financial market predictions. European Journal of Operational Research. 2018; 270(2):654-69.

[36] Attigeri GV, MM MP, Pai RM, Nayak A. Stock market prediction: a big data approach. In TENCON 2015 (pp. 1-5). IEEE.

[37] Kirkpatrick II CD, Dahlquist JA. Technical analysis: the complete resource for financial market technicians. FT Press; 2010.

[38] Saad EW, Prokhorov DV, Wunsch DC. Comparative study of stock trend prediction using time delay, recurrent and probabilistic neural networks. IEEE Transactions on Neural Networks. 1998; 9(6):1456-70.

[39] Vlasenko A, Vlasenko N, Vynokurova O, Bodyanskiy Y, Peleshko D. A novel ensemble neuro-fuzzy model for financial time series forecasting. Data. 2019; 4(3):1-11.
[40] Abiodun OI, Jantan A, Omolara AE, Dada KV, Mohamed NA, Arshad H. State-of-the-art in artificial neural network applications: a survey. Heliyon. 2018; 4(11):1-41.
[41] Hemeida AM, Hassan SA, Mohamed AA, Alkhalaf S, Mahmoud MM, Senjyu T, et al. Nature-inspired algorithms for feed-forward neural network classifiers: a survey of one decade of research. Ain Shams Engineering Journal. 2020; 11(3):659-75.
[42] De JRJ. Stability analysis of the modified Levenberg–Marquardt algorithm for the artificial neural network training. IEEE Transactions on Neural Networks and Learning Systems. 2020; 32(8):3510-24.
[43] Ticknor JL. A bayesian regularized artificial neural network for stock market forecasting. Expert Systems with Applications. 2013; 40(14):5501-6.
[44] Farizawani AG, Puteh M, Marina Y, Rivaie A. A review of artificial neural network learning rule based on multiple variant of conjugate gradient approaches. In journal of physics: conference series 2020 (pp. 1-13). IOP Publishing.
[45] Liu Q, Liu J, Sang R, Li J, Zhang T, Zhang Q. Fast neural network training on FPGA using Quasi-Newton optimization method. IEEE Transactions on Very Large Scale Integration Systems. 2018; 26(8):1575-9.

Md. Ashikur Rahman Khan is a Professor at the Department of Information and Communication Engineering, Noakhali Science and Technology University, Bangladesh. He has been working at Noakhali Science and Technology University since 2006. Professor khan received his Bachelor's and Master of Science from Rajshahi University of Engineering and Technology, Rajshahi, Bangladesh, in 1999 and Bangladesh University of Engineering and Technology, Dhaka, Bangladesh, in 2004, respectively. He achieved his PhD degree in 2012 from University Malaysia Pahang, Malaysia. He also served as an Assistant Engineer in IT Directorate, Bangladesh Rural Electrification Board, Dhaka, from 2001 to 2006. His research interests include Advance Machining, Artificial Intelligence, Neural networks, Machine Learning, Modelling, and Renewable Energy. Dr Khan is performing events as a reviewer and editorial board member supportive of distinct scholarly journals.
Email: ashik@nstu.edu.bd

Md. Furkan Uzzaman is a Bangladeshi computer science engineer. He has completed his graduation and post-graduation from Department of Information and Communication Engineering (ICE), Noakhali Science and Technology University (NSTU), Chattogram, Bangladesh.
Email: furkan.ict31@gmail.com

Ishtiaq Ahammad is a Lecturer at the Department of Computer Science and Engineering (CSE), Prime University, Dhaka, Bangladesh. He has completed his graduation and post-graduation from Department of Information and Communication Engineering (ICE), Noakhali Science and Technology University (NSTU), Chattogram, Bangladesh. He is the University Rank Holder (Merit Position: 1st) at both Undergraduate and Postgraduate levels. He has multiple publications in international journals including reputed publishers like Elsevier, Springer, and IGI Global. His Area of interest includes: Internet of Things (IoT), Software Defined Networking (SDN), Fog Computing, and Edge Computing.
Email: ahammadishtiaq27@gmail.com

Ratul Prosad has completed his graduation and post-graduation from Department of Information and Communication Engineering (ICE), Noakhali Science and Technology University (NSTU), Chattogram, Bangladesh. He is very much attentive to details, accurate and a keen observer. He has the ability to write reports and proposals. He has a strong computational knowledge as well as different kinds of programing language indeed, simulation process. His Area of interest includes: Genetic Algorithm, Biomedical Engineering, Greedy algorithm.
Email: ratulprosad36@gmail.com

Zayed-Us-Salehin is an Assistant Professor at the Department of Information and Communication Engineering, Noakhali Science and Technology University. He has been working in Noakhali Science and Technology University since 2013. His research interests include Computer Networks and Wireless Area Networks.
Email: salehin.fahad@nstu.edu.bd

Tanvir Zaman Khan is currently working as Assistant Professor in Information and Communication Engineering department in Noakhali Science and Technology University, Noakhali-3814, Bangladesh. His research interest includes antenna design, machine learning, pattern recognition, neural networks, computer networking, VLSI system design, image processing, embedded system design, web security & application, telecommunication and feature selection.
Email: tzkhan19@nstu.edu.bd

Md. Sabbir Ejaz is a Lecturer at the Department of Information and Communication Engineering, Noakhali Science and Technology University, Bangladesh. He has been working at Noakhali Science and Technology University since January, 2022. Before he has been working as a Senior Lecturer at Bangladesh Army University of Engineering and Technology, Bangladesh more than 5 years. Md. Sabbir Ejaz received his Bachelor degree in 2016 and a Master of Science in 2022 from Rajshahi University of Engineering and Technology, Rajshahi, Bangladesh. His research interests include Digital Image Processing, Deep Learning, Artificial Intelligence, and Machine Learning. He attended a different conference and workshop. Md. Sabbir Ejaz is performing a role as a reviewer and editorial board member in a journal.
Email: sabbirejaz.ice@nstu.edu.bd

Main Uddin is a lecturer at the Department of Information and Communication Engineering, Noakhali Science and Technology University, Bangladesh. He has obtained the degree B.Sc in Computer Science and Telecommunication Engineering from the same university. Formerly, he would work in Z.H. Sikder University of Science and Technology, Bangladesh as lecturer. His research interest includes Fog Computing, Distributed Network and Network Science.
Email: mainuddin.ice@nstu.edu.bd

Appendix I

S. No.	Abbreviation	Description
1	ANNs	Artificial Neural Networks
2	AI	Artificial Intelligence
3	BiLSTM	Bidirectional LSTM
4	BR	Bayesian Regularization
5	CG	Conjugate Gradient
6	CNN	Convolutional Neural Network
7	DJIA	Dow Jones Industrial Average
8	DL	Deep Learning
9	DNN	Deep Neural Network
10	DSE	Dhaka Stock Exchange
11	DT	Decision Tree
12	EMA	Exponential Moving Average
13	FFNN	Feed-Forward Neural Network
14	FX	Foreign Exchange
15	IT	Information Technology
16	LM	Levenberg Marquardt
17	LSTM	Long Short-Term Memory
18	MAPE	Mean Absolute Percentage Error
19	ML	Machine Learning
20	MSE	Mean Squared Error
21	NNs	Neural Networks
22	QN	Quasi Newton
23	R2	R-Square
24	RMSE	Root-MSE
25	RNN	Recurrent Neural Network
26	SCG	Scaled Conjugate Gradient
27	SLR	Systematic Literature Review
28	SOSE	Sum-of-Squared-Errors
29	SVM	Support Vector Machine