Evolving computationally efficient prediction model for Stock Volatility using CGPANN

Niaz Muhammad
Department of Electrical Engineering
University of Engineering and Technology Peshawar
Peshawar Pakistan

Syed Waqar Shah
Department of Electrical Engineering
University of Engineering and Technology Peshawar
Peshawar Pakistan

Gul Muhammad Khan
Department of Electrical Engineering
University of Engineering and Technology Peshawar
Peshawar Pakistan

2022 2nd International Conference on Artificial Intelligence (ICAI) | 978-1-6654-6896-1/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICAI55435.2022.9773706

***Abstract*—Financial market volatility has become one of the most difficult applications for stock price forecasting in ongoing situations. The current statistical models for stock price forecasting are too rigid and inefficient to appropriately deal with the uncertainty and volatility inherent in stock data. CGPANN-CGP based ANNs and LSTM are the most common methods used these days to predict such dynamics in time series data. In comparison to other methodologies, studies have demonstrated that the application of Cartesian genetic programming evolved Artificial Neural Networks (CGPANNs) to time series forecasting problems produces better results, and LSTM can be competitive at times. CGPANN provides the ability to train both structure, topology, and weights of network to achieve the global optimum solution. The prediction model is trained on the behavior of stock exchange patterns and is based on trends in historical daily stock prices. The proposed CGPANN and LSTM models produced competitive results of 98.86% and 98.52% respectively. However, CGPANN architecture is capable computationally efficient than LSTM and its ability of quick predictions makes it ideal for real-time applications.**

***Index Terms*—Real-time Stock Prediction, CGPANN, Optimum ANN designs, time-series data forecasting**

I. INTRODUCTION

In real-time prediction applications, the second most important factor is computational efficiency. Obtaining global optimum is out of the question if you are manually modifying the architecture of networks and adjusting a few parameters at runtime. CGPANN provides this ability in a much better and computationally cost-effective way, with the previous demonstration of obtaining robust architecture and faster training time [3]. CGPANN evolves all the network parameters obtaining optimum architecture producing the best possible results. The neural system development is based on the characteristics of Cartesian Genetic Programming evolved Artificial Neural Network (CGPANN) is experimented with time series forecasting in past [3] producing the best optimum architecture and competitive accuracy.

Prediction of data sets for stock exchange rates has been exploited for various implementations by ANNs and statistical models. Conventional analytical models are better in terms of performance for the time series forecasting condition while these models have many drawbacks when nonlinear data sets are employed. Other ANN models have been utilized for the problems of time-series forecasting i.e. GARCH (Generalized Auto Regressive Conditional Heteroskedastic) model in [1] seems to perform well in contrast with the non-linear arithmetic models and fixed ANNs models, Smooth Transition Auto-Regressive (STAR) model, the Self-Exciting Threshold Auto-Regressive (SETAR) model, Restricted Vector Auto-Regressive (RVAR) model, Bayesian Vector Auto-Regressive (BVAR) model, the Vector Autoregressive (VAR) model, Auto-Regressive Integrated Moving Average (ARIMA) giving accuracies in the range of 98.39% on average. Generalized Auto-Regressive Conditional Heteroskedasticity (GARCH) model conversely with their linear parts performing incredible with the straight models, in [2] for trade rates expectations the models utilized are Bayesian Vector Error Correction (BVEC) and Vector Error Correction (VEC) model, the MLP (Multilayer Perceptron) in 2007 with the accuracy of 72%, the Volterra with 76% [?] the MLFN (multi-layered feed-forward neural system) in [4] the MNN (Multi-Neural system model) in [4] with an accuracy of 71%, the GRNN (general relapse neural systems) in [4], the AFERFM (ANN unfamiliar conversion scale estimating model) in (Philip (2011)) with 81.2%, and the Hidden Markov Expectation Model for forecasting foreign exchange rate in (Philip (2011)) with 69.9%. The ANNs models mentioned above failed for the dynamic data set scenario of the time series forecasting and performed great for the static data set scenarios of the time series forecasting.

A. Neuro Evolutionary Techniques for Neural Networks

There are several ways in neural networks to find the efficient solution for different substantial problems i.e., designing controllers for robotics agents. The ANN provides effective structure blocks for creating complex operations in AI and computational neuro-science situations [5]. The neural evolutionary algorithms are appealing since they just require an evaluation of the scenario's top-level realization, as opposed to supervised techniques that require time steps per time step error signal. These strategies can achieve the network's parameters and topology, as well as the ability to simultaneously optimize numerous competing objectives (Miettinen and Preface By-Neittaanmaki (1999)). Many approaches for neuroevolution have been proposed, including the direct application of real-valued evolutionary algorithms and the evolution of construction programs whose instructions are interpreted to create neural networks [5]; [6]. The new neural system architecture was introduced in [7] which could add new neurons incrementally in the proposed computationally efficient network architecture. This new system architecture has efficient utilization to solve complex static problems. Classic machine learning techniques like reinforcement learning and back-propagation are useful in simple settings, but they fail to tackle complicated scenarios with arbitrary topologies. Other methods of evolving neural networks include those inspired by computational neuroscience, such as importing synaptic models into developed neural networks. Some early efforts based on simple static neuro evolutionary rules were proposed in (Di Paolo (2000), but more recent works including heterosynaptic primitives were published in (Floreano et al. (1996); Urzelai and Floreano (2001); Niv et al. (2002); Floreano (2002)). The proposed computationally efficient network architecture suggested a reward for the alternative branches. When an optimal behavior is obtained according to the reward, the neuromodulation allowed stopping the synapse changes potentially. The synaptic attribute, i.e., weights, is fixed by the evolutionary algorithm in the majority of the aforementioned circumstances, and hence cannot be modified throughout the development process. Such approaches generate static systems whereas algorithms containing developmental agents that can adapt to the changing work environment are more effective and convenient.

The paper is organized in a standard format where the model and problem are introduced in section I. Section II is about the brief literature of the algorithm and problem. Section III has the methodology, details about datasets, and models. While section IV discussed the results obtained model architecture and equations which evaluation metrics. Section V concludes the paper.

II. Literature Review

To achieve the most accurate and efficient forecasting model for the significant time-series forecasting problems through ANNs. We are using Cartesian Genetic Programming and Artificial Neural Network (CGPANN) for producing efficient techniques for time series forecasting problems. CGPANN model can adjust to the varying environment to achieve the robust architecture of the developmental neural network. To resolve the impact of outside factors on the stock market prices and attain the robust techniques for the prediction model, then this network manages those factors in real-time.

The researchers proposed a deep-learning-based model to predict foreign currency exchange rate, the model (deep learning-based) used for the prediction of currency exchange shows better results than normally employed different prediction techniques. This model is applied over three currency exchange rates which include Pakistani rupee to USD, British pound sterling to USD, and Hong Kong dollar to USD. The results that the deep-learning-based model for prediction is more accurate than the linear model and the support regression model when sentiment was considered in the model [8]. The researchers proposed the artificial neural network (ANN) together with the vector regressive model (VAR), vector error corrective model (VECM), and post-processing to improve the forecasting accuracy of the EUR/USD exchange rates. The results show that RMSE decreased by 32.5% and by 19.3% of the best economic model. Thus, the post-processing model increases the forecast accuracy as compared to other models [9]. When high forecasting accuracy is needed, the researcher proposed Zhang's hybrid model of artificial neural networks can be an effective way to improve forecasting accuracy. The results show that the MAE of the autoregressive integrated moving average model is 20.16% and the MSE is 33.57%. The MAE of the Artificial neural network is 20.06% and the MSE is 33.50%. Similarly, the MAE of Zhang's hybrid model is 13.80% and the MSE is 21.03% [10]. The deep learning models proposed by the researcher include the Long Short-term Memory (LSTM), Support Vector Regression (SVR), Artificial Neural Network (ANN), and neural network with hidden layers. These models predict currency exchange rates between the top traded currencies in the world for which they collected data of 39 years till December 2018. The results show that deep learning models to predict multi-currencies exchange rates exceed 99% average accuracy [11]. In this paper, the researcher proposed the model to optimize the performance estimation of the na¨ıve bays model using a genetic algorithm for automatic feature selection. The dataset for the model was EUR/USD of the currency exchange market by which the whole model was tested. The results of the proposed model improved the un-optimized system from 0, 43% to 10, 29% [12]. The powerful data modeling tool proposed by the researcher is a Neural network (NNs) that can represent input and output relationships. The main aim of the modeling tool neural network (NNs) was used to predict the volatility of the Kenyan exchange rate forecasting against the four foreign currencies exchange rates forecasting. The researcher proposed the neural network (NNs) and multilayer perceptron (MLP) with backpropagation learning algorithm will be applied to predict the most volatile Kenyan exchange rates against the four major foreign currencies. The data set was collected from 2005 to 2017 over the trading period of ten years from the central bank of Kenya (CBK). The results show that KES to JPY exchange rates the MSE, MAE, and MAPE of the neural network (NNs) model and multilayer perceptron (MLP) with backpropagation learning algorithm is 0.001889, 0.022, and 2.22, and the predicted value is 55 which is a much better result than any of the other four foreign currencies [13].

The researchers proposed the trading tool to enhance the decision-making process by using the artificial neural network and genetic algorithms. Both the genetic algorithms (GAs) and Artificial neural networks (ANNs) combine the two signals generated by the user of the trading tool and output the set of the signals to decide whether to buy, hold or sell in the current situation of the stock market. The system is based on the data sets collected by the researchers over the last 5 years period which spans ten currencies. So, the results of this optimized system are significantly more profitable as compared to an unoptimized system. The optimal system is more profitable than an un-optimized system up to a 5% on an average level when looking per currency and when looking per period is signed up to 1% and over the latest three periods, the average level is significant up to 0.05% [14].

The researcher proposed the architecture of recurrent deep learning to predict financial time series forecasting. For the efficiency and accuracy of the trading model predictions, the researchers compare the deep recurrent neural network and feed-forward network to long short-term memory and gated recurrent networks. The results show the suitability for the exchange rate forecasting of deep recurrent neural networks as compared to other trading models [15].

The researcher proposed the problem domain of the stock market prices which is a challenging task affected by some factors such as social media sentiments, company performance, investor sentiment. The researcher used the multilayer perceptron (MLP) algorithm of artificial neural network (ANN) to predict efficient future stock market prices with better accuracy. The stock market dataset has been taken from Ghana Stock Exchange (GSE) between January 2010 and September 2019 and predicted the stock price value for a different number of days. Specificity, Sensitivity, RMSE, MAPE and Accuracy 90-days ahead of predicting stock value based on web news 0.43, 0.62, 0.9135, 0.6663 and 55.81%, based on Google trends 0.50, 0.61, 0.0050, 0.5123 and 52.95%, based on Twitter 0.51, 0.69, 0.0047, 0.4874 and 60.05%, based on forum post 0.41, 0.49, 1.0177, 0.7647 and 41.77%, and based on combined data-set 0.69, 0.85, 0.0034, 0.3624 and 77.12% [16].

The researcher used many models such as the LSTM model, regression model, support vector machine (SVM) model, ANN model, etc. for prediction by taking the previous year's dataset. The researcher collected results from different models, but only the SVM model has shown better accuracy in results by predicting stock market prices as compared to other models. The dataset has been taken from yahoo finance and uses the SVM model to train the dataset for predicting the stock market prices for both companies Reliance Ltd. And Tesla Inc. of yahoo finance [17].

The researcher proposed the machine learning (ML) model to predict the variation of 100 indexes Borsa Istanbul (BIST). The researcher used the model that was performed by two algorithms such as multilayer perceptron–particle swarm optimization (MLP–PSO) and multilayer perceptron–genetic algorithms (MLP–GA) by considering two cases Gaussian function and tanh(x) as the output function. The financial time series data has been utilized by the researcher is from 1996 to 2020. After testing the model provided higher accuracy for MLP–GA with a population size of 50, followed by MLP–PSO with a population size of 125. The results showed the MAPE of 28.16%, 29.09%, RMSE of 0.732583, 0.733063, and correlation coefficient of 0.694, 0.695 respectively. After the results, the prediction accuracy was successfully improved by the use of the hybrid ML method [18].

The researcher proposed a machine learning (ML) model to predict the stock market prices with higher accuracy. The main problem is that the pattern of the data is difficult to identify in high-dimensional data. The variations in the stock market prices make forecasting difficult. This research will tell you the technique to identify the hidden pattern within the dataset and provides the model to forecast the future data of the stock market prices with the best accuracy. The accuracy of the proposed model is 80% because the dataset of the stock as input to the model is affected by some other factors [19].

The stock market prediction accuracy is a more interested and challenging task for the investors in the stock market because of globalization. For this, the researcher proposed an Artificial neural network (ANN) model to predict the stock market Prices prediction with better accuracy. The most widely used ANN models in stock market predictions are MLP and BP. There are different strategies used to improve the performance of the model and also these strategies may be combined to improve the accuracy [20].

There are many techniques like time series forecasting, fundamental analysis, technical analysis, etc. used to predict the share price in the share market but all the methods were not capable to predict share price with better accuracy. There are two ways to predict share market price by using the ANN model, one is a prediction based on previous data and the other is a training session. The researcher used a multilayer feedforward network for predicting share price and for the training session he used the back-propagation algorithm. The historical data has been taken from the ACI pharmaceutical company; the researcher tested this data first for 2 input datasets to predict the share price for the future 8 days of November month. The error ranges between (1.74% -5.65%) for 2 input datasets. The researcher tested this data first for 5 input datasets to predict the share price for the future 8 days of November month. The error ranging between (0.58% - 3.23%) for 2 input data sets [21].

A good number of researches were cited but when it comes to the current advancements in the field of time series prediction in the stock market. In the paper [22], the researchers used an LSTM on 10 years of data. The researchers claim accuracy of 63.59%, 56.25%, and 57.95% on the datasets of HDFC, Yes Bank, and SBI, respectively. In another research [23], the authors propose convolution neural networks on NASDAQ and NYSE data and all the results claimed are below 90%. Besides low accuracies, the other benefit of the proposed model is developed on evolution approaches. Therefore, the model designed was very small and only have that neuron which is necessary for the scenarios.

III. Methodology

Statistical and neural network models have explored the time series forecasting domain for several implementations and applications. When using linear data sets, traditional statistical models perform better for time series forecasting, but when using non-linear data sets, these models have several limitations (Refenes et al. (1993); Kadilar et al. (2011)). According to a survey conducted by Azoff (1994), 127 business applications based on ANNs were published in the first year, followed by 86 additional applications in the second year. There are already over 20 commercially available ANN-based solutions that have been effectively created for use in financial time series scenarios. The experiment was done on time series forecasting also demonstrated that none of these models could perform well when a dynamic scenario was presented, proving the high rate of applications of ANNs in the time series forecasting scenario. As a result, the computationally efficient network architecture using the CGPANN technique has been proposed in this paper to be applied to a time series forecasting scenario, such as stock market forecasting, to produce a robust system that can handle a dynamic environment.

A. Experimental Setup

To be used efficiently for experiment assessment, the data is first normalized in the range of [0, 1] (using the usual normalization formula). The normalized process is used to convert data from several stock markets with varying ranges into identical ranges for use in a generalized experimental setup. CGPANN uses the fittest mode technique to encode and update network properties like topology, weights, and functions. The evolved neural network consists of processing units called “neurons” that are used to encode the network parameters i.e., input connections, weights, and activation function in the network. The ultimate system input(s) or output(s) from the preceding nodes can be used as input(s) to the neurons. The switch attribute is linked to the inputs, which turn on or off their connection depending on the state of the switch attribute. The range of connection weights randomly generated ranging from -1 to +1 in the system. The neurons’ output(s) can be the ultimate system input(s) or any previous node’s output(s). Every node is linked with a weighted input connection and the result of the node is the output of each node which is forwarded to the summation function node and the output of the summation function is further forwarded to the activation function to generate the resulting node output. CGPANN genotypes are evolved through mutation from one generation to the next until the target result is obtained.

B. The Genotype and Phenotype of the CGPANN Approach

The genotype is the numerical representation like strings of numbers of the neural network while phenotype is the physical network representation in terms of neurons and input/output connections. Figure 2 represents both the genotype and phenotype of CGPANN. The CGPANN structure in Figure 2 represents the number of inputs (Input0, Input1, Input2, · · · , Input9), node outputs (Outpt0, Output1, Output2, · · · , Output9),connection weights (weight0, weight1, weight2, · · · , weight17), connected nodes that are active (0, 1, 2, 5, 7, 8, 9),nodes that are in-active(3,6) and arity is 2. As we can see the genotype of the CGPANN network is in terms of string numbers in Figure 2.

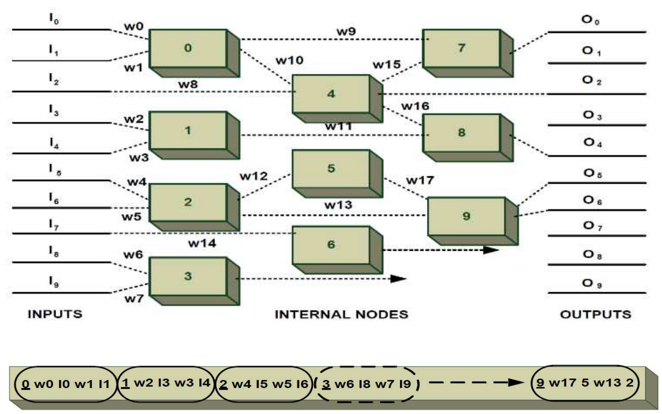

Fig. 1. CGPANN genotype and phenotype

1) Data selection, Acquisition, and Formulation: There are two sections in the experimental setup of the CGPANN model, one is training and the other is the testing section. The stock exchange rate prediction network is trained on 500 days of historical data from the Wilshire US Large-Cap Growth Total Market Index (WILLLRGCAPGR). The US market index shows the daily market value at market closing time. The closing time of the market is 4 PM daily and it closes early on the holidays. The dataset of Wilshire US Large-Cap Growth Total Market Index has been obtained from Federal Reserve Economic Data (FRED), Federal Reserve Bank of St. Louis. The data is acquired in the form of average daily market prices from June 1, 2017, until the next 500 days. The rate of the stock market is taken at 4 PM daily concerning other stock market prices and some values were taken earlier in the holidays. The price rate of the market index is taken daily at the end of the day. The model is tested using data of fifteen stock markets obtained from the Federal Reserve Bank of St. Louis' Federal Reserve Economic Data (FRED). The data sets are obtained for testing in terms of average daily market index values from 1st June 2017 to the next 1000 days. To be used efficiently for experiment assessment, the data is first normalized in the range of [0, 1] (using the usual normalization formula). The normalizing process is used to convert data from several stock prices with varying ranges into equivalent ranges for use in a generalized experimental setup.

IV. Results and Analysis

In this area of research, we will explain the results and analysis of the proposed CGPANN model for the scenario of stock exchange forecasting. We have used CGPANN to Produce three different Network Architectures to evolve the neural networks for time series analysis. In the evolution process of these network architectures, the parent genotype of CGPANN is generated initially which are further mutated by the process of mutation to developed more genotypes. CGPANN's network evolution method results in an optimized network with the optimal number of neurons, inputs, and outputs. Figure 2,3, and 4 demonstrates the construction of a typical CGPANN neuron, which includes inputs (I0, I1, I2,...., I7), a node function (F), weights associated with the input connections shown in the three different network architectures, a summation function, and outputs. The processing unit/neurons are the functional nodes of the neural network. The number of inputs and outputs and the number of days prediction are the same for each architecture except the input per node which is different for each architecture. The computationally efficient network architectures as shown in Figures 2,3, and 4 carry different numbers of inputs per node connection for the three architectures as needed for the network to generate optimal results. The nodes (Junk nodes) may be disabled from participating in the network's ultimate output or permitted to actively participate in developing the system output throughout the evolution process.1, 2, and 3 demonstrates the mathematical representations of the three different architectures, where the weights are connected to the adjacent input connections. F8, F9, F10, F11, F12, F13, and F14 are the output of the node for the three architectures. The ultimate system output has been obtained by averaging all the system outputs. During the process, the network takes the eight historical values and predicts the eight future values of stock market prices.

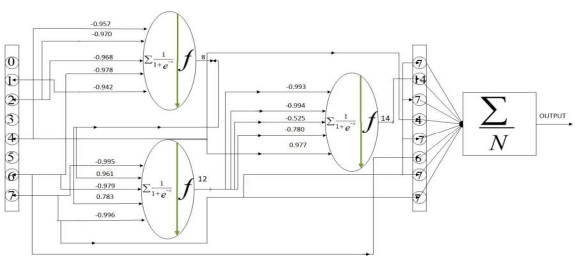

Fig. 2. 3 Nodes Feed-forward-CGPANN Architecture

$$
\begin{aligned}
F8 &= [\frac{1}{1+e^{-x}}] \\
X &= -0.957I4 + (-0.970I2) \\
&\quad + (-0.968I2) + (-0.978I6) + (-0.942I1) \\
F12 &= [\frac{1}{1+e^{-x}}] \\
X &= -0.995I7 + (0.961I8) \\
&\quad + (-0.979I6) + (0.783I8) + (-0.996I7) \\
F14 &= [\frac{1}{1+e^{-x}}] \\
X &= -0.993I12 + (-0.994I12) \\
&\quad + (-0.525I12) + (-0.780I12) + (0.977I4)
\end{aligned} \tag{1}
$$

$$
\begin{aligned}
F8 &= [\frac{1}{1+e^{-x}}] \\
X &= -0.989I2 + (-0.901I2) \\
&\quad + (0.416I0) + (-0.967I3) \\
&\quad + (-0.877I2) + (-0.949I3) \\
F10 &= [\frac{1}{1+e^{-x}}] \\
X &= -0.988I8 + (-0.998I8) \\
&\quad + (-0.964I8) + (-0.991I8) \\
&\quad + (-0.920I8) + (-0.984I8) \\
F11 &= [\frac{1}{1+e^{-x}}] \\
X &= 0.930I3 + (-0.961I8) \\
&\quad + (-0.999I8) + (0.796I3) \\
&\quad + (-0.989I8) + (-0.990I3) \\
F13 &= [\frac{1}{1+e^{-x}}] \\
X &= -0.982I8 + (-0.931I8) \\
&\quad + (-0.994I8) + (-0.993I8) \\
&\quad + (-0.999I8) + (0.886I3)
\end{aligned} \tag{2}
$$

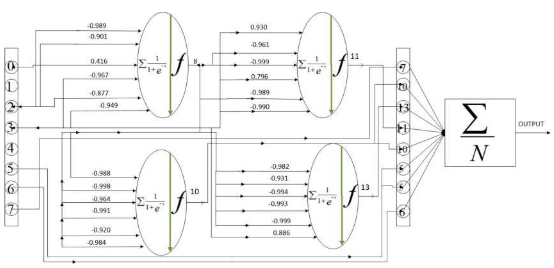

Fig. 3. 4 Nodes Feed-forward-CGPANN Architecture

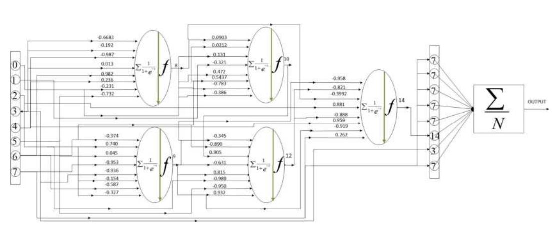

Fig. 4. 5 Nodes Feed-forward-CGPANN Architecture

$$
\begin{aligned}
F8 &= [\frac{1}{1+e^{-x}}] \\
X &= -0.6683I3 + (-0.192I3) \\
&\quad + (-0.987I4) + (0.013I3) \\
&\quad + (0.982I6) + (0.236I4) \\
&\quad + (-0.231I0) + (-0.732I5) \\
F9 &= [\frac{1}{1+e^{-x}}] \\
X &= -0.974I6 + (0.740I2) \\
&\quad + (0.045I5) + (-0.953I7) \\
&\quad + (-0.936I3) + (-0.154I2) \\
&\quad + (-0.587I6) + (-0.327I2) \\
F10 &= [\frac{1}{1+e^{-x}}] \\
X &= 0.0903I8 + (0.0212I6) \\
&\quad + (0.131I0) + (-0.321I1) \\
&\quad + (0.472I0) + (0.5437I6) \\
&\quad + (-0.783I0) + (-0.386I0) \\
F12 &= [\frac{1}{1+e^{-x}}] \\
X &= -0.345I9 + (-0.890I6) \\
&\quad + (0.905I10) + (-0.631I9) \\
&\quad + (0.815I9) + (-0.980I2) \\
&\quad + (-0.950I5) + (0.932I9) \\
F14 &= [\frac{1}{1+e^{-x}}] \\
X &= -0.958I9 + (-0.821I12) \\
&\quad + (-0.3992I8) + (0.881I2) \\
&\quad + (-0.888I12) + (0.959I4) \\
&\quad + (-0.919I9) + (0.262I7)
\end{aligned} \tag{3}
$$

The performance of the proposed prediction model is evaluated from the comparison of actual value and predicted value of stock market prices. MAE is used to measure the performance of the proposed model and also used to evaluate the performance of the network. In the training phase, the proposed model is initially trained on the historical data of Wilshire US Large-Cap Total Market Index for 500 days and after that, the trained network is further tested on fourteen datasets for 1000 days of data of stock exchange markets. The proposed CGPANN model is based on the Feedforward architecture in both the training and testing phase that generates excellent results for both phases. The performance of Feedforward architecture is helpful for the evaluation of the system. The system is not provided with the output values of the Stock prices in the testing session because the system has already learned the behavior of the time series data after the training session of the network architecture. The sliding window concept has been used in the code of CGPANN model for the prediction process. The proposed CGPANN model results in the testing section are illustrated in Table 1. The network MAE or fitness value received as low as 1.14%. The network is run on 50 nodes and achieved the best results with the lowest fitness or MAE value. This process is repeated for 15 different datasets of stock market prices. The MAE and MAPE or fitness values achieved for the CGPANN model were as low as 1.73% in the testing phase. The values of historical data oscillate highly as they can be recognized from different fitness values for multiple days shown in Table 2. It is an outstanding performance with 98.86% accuracy that y has been retrieved by the CGPANN model for such types of datasets. The performance of the CGPANN model for different datasets showed in Table 2.

TABLE I
PERFORMANCE OF CGPANN NETWORK

Stock Markets	MAE	MAPE	ACCURACY
WILLLRGCAPGR	0.0114	1.14	98.86
DJIA	0.0411	4.11	95.89
NASDAQCOM	0.0173	1.73	98.27
NASDAQ100	0.1102	11.02	88.98
SP500	0.0242	2.42	97.58
WILLLRGCAP	0.0206	2.06	97.94
WILL5000PR	0.0216	2.16	97.84
WILL5000INDFC	0.0204	2.04	97.96
WILL5000IND	0.0208	2.08	97.92
WILLMIDCAP	0.0254	2.54	97.46
WILLREITIND	0.0516	5.16	94.84
WILLRESIPR	0.1020	10.20	89.8
WILLSMLCAP	0.1177	11.77	88.23
WILLLRGCAPVAL	0.0256	2.56	97.44
WILLMICROCAP	0.0195	1.95	98.05

1) *Wilshire US Large-Cap Growth Total Market Index:* In Figure 3, the horizontal axis represents the number of days and the vertical axis shows the normalized price values of the Wilshire US Large-Cap Growth Total Market Index (WILLLRGCAPGR) stock market. The MAPE for this market index in the training phase is 1.14% and the accuracy for the market in the training phase is 98.86%. The MAE and MAPE or fitness value achieved for the CGPANN model as low as 1.73% for in the testing phase.

Fig. 5. DJIA Est Vs Target

2) *NASDAQ COMPOSITE INDEX:* In figure 4, the horizontal axis represents the number of days and the vertical axis shows the normalized prices of the NASDAQCOM stock market index. The MAPE for this market index is 1.73% and the

accuracy for the market is 98.27% during the testing phase. The standard deviation for the NASDAQCOM is 0.119.

Fig. 6. NASDAQCOM Est Vs Target

3) Wilshire US Micro-Cap Total Market Index: In Figure 5, the horizontal axis represents the number of days and the vertical axis shows the normalized price values of the WILLMICROCAP stock index. The MAPE for this market index in the case of the testing phase is 1.95% and the accuracy for the market index in the testing phase is 98.05%. The standard deviation for the WILLMICROCAP is 0.089.

Fig. 7. WILLMICROCAP Est Vs Target

4) Robustness of CGPANN: The proposed CGPANN prediction model's robustness may be examined in a developing scenario. By changing the prediction criteria, the proposed CGPANN model is tested in a developmental scenario to retrieve the results with high accuracy. The proposed CGPANN model is trained to predict the ninth day's data based on the historical eight days data. The evaluation scenario is altered by predicting data for multiple days rather than a single day based on the preceding eight days' data. Table 2 evaluates the proposed CGPANN performance for the case of multiple days. The performance of CGPANN for several days predictions is shown in Table 2 which indicates the CGPANN performance achieving MAPE value as low as 2.668%. This is a collection of results demonstrating the CGPANN model's robustness while the learning scenario for multiple days forecasting is constantly changing.

TABLE II

PERFORMANCE OF CGPANN NETWORK

Days	10	30	90	180	360
DJIA	0.0311	0.0399	0.0769	0.0837	0.114
NASDAQCOM	0.0175	0.0278	0.0405	0.0782	0.0746
NASDAQ100	0.0200	0.0393	0.0734	0.0778	0.0769
SP500	0.0246	0.0366	0.0601	0.0827	0.0862
WILLLRGCAP	0.0216	0.0322	0.0454	0.0636	0.0813
WILL5000PR	0.0247	0.0345	0.0453	0.0761	0.0837
WILL5000INDFC	0.0203	0.0407	0.0529	0.0784	0.0767
WILL5000IND	0.0259	0.0368	0.0530	0.0665	0.0830
WILLLRGCAPVAL	0.0236	0.0344	0.0690	0.0761	0.0807
WILLMICROCAP	0.0180	0.0252	0.0374	0.0383	0.0689
WILLMIDCAP	0.0203	0.0283	0.0512	0.0644	0.0786
WILLRETTIND	0.0258	0.0386	0.0465	0.0779	0.0872
WILLRESIPR	0.0206	0.0322	0.0534	0.0765	0.0927
WILLSMLCAP	0.0210	0.0306	0.0549	0.0763	0.0866

TABLE III

PERFORMANCE OF CGPANN NETWORK

MODEL	MAE	ACCURACY
Hidden Markov Model (HMM) (Refenes et al. (1993))	0.0198	98.02
ARIMA (Bidlo (2007))	0.0161	98.39
Multi-Layer Perceptron (MLP) (Kryuchin et al.(2007b))	0.28	72.00
Back Propagation Network (Chen et al.(2008b))	0.377	62.27
Multi Neural Network (MNN) Model (Chen et al. (2008b))	0.331	66.82
Proposed CGPANN	0.0114	98.86
LSTM	0.0148	98.52
Simple RNN	0.0171	98.29

V. DISCUSSION

A. Comparison of CGPANN with other ANN

To have a comparison analysis, the CGPANN model has achieved high accuracy as compared to the performance accuracy of another artificial neural network (ANN) model. The performance accuracy of the time series forecasting model is very high as compared to other models are illustrated in Table 3. As we can observe that the accuracy of the CGPANN model is 98.86% in Table 3, which is very high as compared to other models.

B. CGPANN model comparison with other standards Models

The CGPANN model achieved the best results with low fitness value and high accuracy in the training and testing section. In comparison analysis, the performance of the CGPANN model is highly accurate as compared to other benchmark models. The results of other standard models i.e., Autoregressive–Moving-Average (ARMA) model, Moving Average Convergence/Divergence (MACD), and Naive strategy model are therefore compared with the proposed CGPANN model to have a comparative analysis. Table 4 shows the mean absolute percentage error (MAPE) of the CGPANN model in comparison to other Benchmark models. The fitness value of the CGPANN approach is 1.8% and the fitness value for the Moving Average Convergence/Divergence (MACD) model is 1.95%, so its means that the accuracy of the CGPANN approach is high

TABLE IV
PERFORMANCE OF CGPANN NETWORK

NETWORK	MAE
MACD Model	0.0195
ARIMA Model	0.0346
Naive Model	0.0728
Proposed CGPANN	0.0114

and much more accurate as compared to other benchmark techniques illustrated in Table 4.

VI. CONCLUSION

This paper proposes and uses a neural network based on the representation of Cartesian Genetic Programming (CGPANN) to a critical scenario of time series forecasting, namely Stock Exchange Rates Forecasting. The CGPANN causes the system parameters to be mutated at run time, causing the network to change over time in response to the learning situation. An extra developing gene from the CGPANN output promotes mutation at runtime in the network. When compared to its static counterpart i.e., ANN, the CGPANN produces a more resilient and computationally efficient network architecture. Mutations create each time a brand-new output is received from the CGPANN based architecture, primarily based on the decision module. The stock exchange scenario's results demonstrated that the system is stable and capable of learning in the changing task environment. In the future, we wanted to examine if the CGPANN can be utilized to resolve issues faster and more accurately by repeating its learning experience in the task environment. The ultimate goal is to see how CGPANN performs in a variety of scenarios and see if it can be trained to solve a range of problems without forgetting how to solve previous ones. In any dynamic environment, CGPANN may successfully develop a robust and efficient model for various circumstances, with more efficient and accurate outputs and an optimal network.

ACKNOWLEDGMENT

This research is funded by the National Center of Artificial Intelligence, University of Engineering and Technology Peshawar. All codes were written in python language. All networks were trained on a laptop with GPU specifications of Nvidia GeForce MX150 4 GB.

REFERENCES

[1] D. G. McMillan and A. E. Speight, "Daily volatility forecasts: Reassessing the performance of GARCH models," Journal of Forecasting, vol. 23, no. 6, pp. 449-460, 2004.

[2] J. C. Cuaresma and J. Hlouskova, "Beating the random walk in central and eastern Europe," Journal of Forecasting, vol. 24, no. 3, pp. 189-201, 2005.

[3] G. M. Khan, and A. M. Ahmad, "Breaking the stereotypical dogma of artificial neural networks with cartesian genetic programming", Inspired by nature, Springer, pp. 213-233, 2018.

[4] J. Chen and Z. Chen, "Extended Bayesian information criteria for model selection with large model spaces," Biometrika, vol. 95, no. 3, pp. 759771, 2008.

[5] X. Yao, "Evolving artificial neural networks," Proceedings of the IEEE, vol. 87, no. 9, pp. 1423-1447, 1999.

[6] D. Floreano, P. Durr, and C. Mattiussi, "Neuroevolution: from architect- ¨ tures to learning," Evolutionary Intelligence, vol. 1, no. 1, pp. 47-62, 2008.

[7] N. Kasabov, "Evolving fuzzy neural networks-algorithms, applications, and biological motivation," Methodologies for the conception, design, and application of soft computing, World Scientific, vol. 1, pp. 271274, 1998. (Alife XI), 2008, no. CONF: MIT Press, pp. 569–576.

[8] M. Yasir et al., "An intelligent event-sentiment-based daily foreign exchange rate forecasting system," Applied Sciences, vol. 9, no. 15, p. 2980, 2019.

[9] A. Parot, K. Michell, and W. D. Kristjanpoller, "Using Artificial Neural Networks to forecast Exchange Rate, including VAR-VECM residual analysis and prediction linear combination," Intelligent Systems in Accounting, Finance, and Management, vol. 26, no. 1, pp. 3-15, 2019.

[10] M. Khashei and M. Bijari, "An artificial neural network (p, d, q) model for time series forecasting," Expert Systems with Applications, vol. 37, no. 1, pp. 479-489, 2010.

[11] A. R. Nagpure, "Prediction of multi-currency exchange rates using deep learning," International Journal of Innovative Technology and Exploring Engineering, vol. 8, no. 6, pp. 316-322, 2019.

[12] G. Abreu, R. Neves, and N. Horta, "Currency exchange prediction using machine learning, genetic algorithms, and technical analysis," arXiv preprint arXiv:1805.11232, 2018.

[13] G. Kipruto, J. K. Mung'atu, G. O. Orwa, and N. Wairimu, "Application of Artificial Neural Network (Ann) in Modeling Foreign Currency Exchange Rates:(A case of Kenyan Shilling against four world's major currency)," Intl. J. Scie. Rese. Manag., vol. 6, no. 10, pp. M-2018.

[14] H. Thinyane and J. Millin, "An investigation into the use of intelligent systems for currency trading," Computational Economics, vol. 37, no. 4, pp. 363-374, 2011.

[15] A. J. Dautel, W. K. Hardle, S. Lessmann, and H.-V. Seow, "Forex¨ exchange rate forecasting using deep recurrent neural networks," Digital Finance, vol. 2, no. 1, pp. 69-96, 2020.

[16] I. K. Nti, A. F. Adekoya, and B. A. Weyori, "Predicting Stock Market Price Movement Using Sentiment Analysis: Evidence From Ghana," Appl. Comput. Syst., vol. 25, no. 1, pp. 33-42, 2020.

[17] P. Werawithayaset and S. Tritilanunt, "Stock Closing Price Prediction Using Machine Learning," in 2019 17th International Conference on ICT and Knowledge Engineering (ICT &KE), 2019: IEEE, pp. 1-8.

[18] F. Ecer, S. Ardabili, S. S. Band, and A. Mosavi, "Training multilayer perceptron with genetic algorithms and particle swarm optimization for modeling stock price index prediction," Entropy, vol. 22, no. 11, p. 1239, 2020.

[19] R. Arevalo, J. Garc´ ´ıa, F. Guijarro, and A. Peris, "A dynamic trading rule based on filtered flag pattern recognition for stock market price forecasting," Expert Systems with Applications, vol. 81, pp. 177-192, 2017.

[20] S. Simon and A. Root, "Accuracy drove artificial neural networks in stock market prediction," International journal on soft computing, vol. 3, no. 2, p. 35, 2012.

[21] Z. H. Khan, T. S. Alin, and M. A. Hussain, "Price prediction of share market using artificial neural network (ANN)," International Journal of Computer Applications, vol. 22, no. 2, pp. 42-47, 2011.

[22] Agrawal, M., Shukla, P. K., Nair, R., Nayyar, A., & Masud, M. (2022). Stock prediction based on technical indicators using a deep learning model. CMC-Comput Mater Cont, 70(1), 287-304.

[23] Chandar, S.K. Convolutional neural network for stock trading using technical indicators. Autom Softw Eng 29, 16 (2022). https://doi.org/10.1007/s10515-021-00303-z