A GA-based polynomial FLANN with exploration and incorporation of virtual data points for financial time series forecasting

Subhranginee Das, Sarat Chandra Nayak, Sanjib Kumar Nayak, Biswajit Sahoo

Abstract— Global stock markets across the world continue to see a phenomenal increase in adoption and interest over the years, largely because of rapid expansion of international financial links, coupled with liberalization in stock markets. Individuals as well as financial institutions across the globe engage in trading stocks and derivatives, attempting to leverage benefits, associated with accurate prediction of the price trends and value. However, non-linear nature of the stock movements and their varied levels of volatility, adds to the challenges of prediction accuracy. While diverse artificial computational models, particularly multilayer perceptron (MLP) is arguably most frequently used forecasting model because of its good approximation and generalization abilities, the model's computational complexities amplify manifold with increase in number of layers. Also, a greater neuron density in each layer and its black-box nature compels researchers to adopt computationally simpler models such as functional link artificial neural networks (FLANN). This article proposes a polynomial functional link artificial neural network (Poly-FLANN) model for stock movement forecasting. The proposed forecasting model is applied to forecast daily closing indices of BSE, NASDAQ, FTSE, TAIEX, and DJIA. Compared to the conventional FLANN, the process of functional expansion of input data is carried out after two data processing methods such as exploration and incorporation of virtual data points (VDP) to the original stock prices and combination of input data. Addition of VDPs to the original financial time series expands the volume of time series, where as taking combinations of data elements increases the dimension of the original financial time series. The performance of the proposed model is compared with trigonometric FLANN (TFLANN), Chebysheb FLANN (CFLANN), Lagurre FLANN (LFLANN), Legendre FLANN (LeFLANN), and a MLP model. All these models are trained by genetic algorithm (GA). Extensive simulation results prove the efficiency of the proposed model in terms of generating less forecasting error signals.

Key-word: stock market forecasting; multilayer perceptron; functional artificial neural network; virtual data position; genetic algorithm.

1. INTRODUCTION

Stock index forecasting is enormously challenging because of its volatile nature. Stock index prediction is the forecasting of a stock index by analyzing the existing stock indices. Stock index forecasting has been considered as an area of research in financial engineering as well as economical mathematics. Stock market is behaving as random walk and statistically the serial correlation is immaterial. Since the stock market movement is immeasurably uncertain it is considered as an exigent task. Prediction of market is pretty difficult not only because of nonlinearities and volatile nature but also be deficient in incessant data, other stock market activities, national politics influences and numerous macro-economical factors and also psychology of human beings [1]. A variety of economical factors such as exchange rate, crude oil price, reserve bank interest rate, stock market indices of global market, economical condition of the country etc. have been used in the study of stock index prediction [11] and are found influencing stock market behavior. Day by day investment in stock market is escalating by the stock brokers, individual investors and farm, therefore it becoming popular. Hence a robust prediction model is always attractive to the investors in stock markets. Lots of research is going on predicting stock market successfully so that the investors made substantial profit. In last few years, numerous methods have been developed for proper forecasting of the stock markets [34, 35, 42]. Being influenced by many factors experimentation in this field carries different theoretical and practical challenges. According to the market hypothesis, the profit from a stock price fluctuation is very uncertain. In a businesslike stock market, it reflects accessible information about stocks and thus any opening of earning excess profit come to standstill in short time interval. So it is assured that no system can consistently beat the market predictability.

Linear models have been used for forecasting in financial engineering since many decades. As financial time series is nonlinear in nature and existence of noise is there, traditional linear models are not so effective. Financial time series uses past indices to predict the future. Here the association among former and later prices is totally nonlinear, so a small change in the past price can affect the future price widely. Different statistical based techniques comprehensively been implemented for stock exchange prediction [30]. Different approaches such as moving averages (MA), auto-regressive integrated moving average (ARIMA), auto-regressive heteroscedastic (ARCH), generalized ARCH (GARCH) are set up to be well-liked amongst statistical models. These models are experimented in diverse engineering and financial applications. These statistical models are not to a large extent resourceful in handling the nonlinear time series. These are

Revised Version Manuscript Received on March 10, 2019.

Subhranginee Das, School of Computer Engineering, KIIT University, Bhubaneswar, India.(E-Mail: subhrangineedas@gmail.com)

Sarat Chandra Nayak, Department of Computer Science and Engineering, CMR College of Engineering & Technology, Hyderabad, Telangana, India.(E-Mail: saratnayak234@gmail.com)

Sanjib Kumar Nayak, Department of Computer Application, Veer Surendra Sai University of Technology, Burla, India.(scansanjib@gmail.com)

Biswajit Sahoo, Telangana, India (E-mail: bsahoofcs@kiit.ac.in)

developed to handle certain type of problems. Autoregressive moving average (ARMA) models used in Box-Jenkins method [5] are comprehensively used in time series forecasting.

Tremendous research is going on soft computing, fuzzy algorithms, artificial neural networks (ANN) and evolutionary computations. This development in computational intelligence helps to build up dynamic, complex and multivariable nonlinear systems. Different soft computing techniques have been implemented successfully for data classification, credit scoring, stock exchange forecasting, risk level evaluation, portfolio management of individual and farms etc. ANN uses previous data to train the neuron to get better accuracy in prediction which is an advantage. ANN models are adaptive in nature and can handle nonlinearity. ANNs can be used as excellent approximator to predict to any desired accuracy when specified a continuous function.

In computational finance, artificial neural networks have gained ample receiving for the reason of improved approximation capabilities and learning abilities. ANNs are effective course of action for depicting input to output that contains exceptions as well as regularities. ANN can handle multifaceted problems having constitutional instability. Artificial neural networks are quite homologous to non-linear, nonparametric regression models. The uniqueness of ANN is, it can imitate non-linear processes with a little prior assumption. Since in financial engineering to determine asset prices, much is presumed and a very little fact is conventional regarding the processes; NN is useful.

ANNs can successively handle non-linear and noisy data and have the ability to use different kinds of input. Multilayer perceptrons (MLPs) which are used for stock market prediction have resemblance with other intricate nonlinear models based on exponential GARCH processes [21, 37, 38]. Modular neural networks [38] are used in advance for finding the relation among different market factors. They applied it on Tokyo stock exchange; several other forecasting models and learning algorithms have already been used for prediction of Tokyo stock market. Compared to multiple regression models these models are having much higher correlation coefficients. The researcher produced different composite rules [31] by combining different networks using logical operators and suggested that their hybrid synthesized rule prohibits higher gain than previous methods. ANNs have been predominately applied in financial forecasting [13, 26, 19]. Cao et al. [6] had done research on Shanghai stock exchange; used ANNs for prediction of stock value of different firms. In that the predictability of linear models are compared with that of single variable and multi variable neural network models. All the analysis shows that neural based models outperformed over other. The results obtained are significant; indicating ANN imitations are excellent for stock exchange prediction. ANN and linear regression models were used by Leigh et al. [19] on New York Stock Exchange Composite Index. The results obtained were stout and informative for the function of trading volume. Chen et al. [7] used probabilistic neural model to predict the market index of the Taiwan stock exchange. Then with Kalman filter the results were compared to the generalized methods of moments (GMM).

In the field of non linear prediction, neuro-genetic hybrid algorithmic networks gained ample applications because of its large adaptive and learning capacity [34, 35, 42]. Since last few decades back propagation neural network (BPNN) was the most commonly used model. It has drawbacks such as large computation time, slow learning rate, stuck to local minima. RBF neural networks are moreover popular and used in stock market prediction. It has superior approximation and expanding abilities. It can also handle stronger nonlinearity [46]. In terms of learning and accuracy the hybrid iterative evolutionary algorithms are more effectual compared to conventional algorithms [20].

Till today multilayer perceptron (MLP) is the far and wide used model for stock prediction. An MLP as the name indicates consist of multiple hidden layers with each layer having more than one neuron. The specified input is applied on the input layer and then it passes through layers. Throughout this forward propagation the synaptic weights of the networks stay unchanging. In backward propagation period the weights get updated according to the learning rule. This backward learning algorithm is popularly known as back propagation algorithm. It has two major drawbacks such as slow convergence and sticking to local minima. Different research [8] on MLP claimed that the foremost lacuna of standard MLP is sticking to local minima. Also, there is no such specific way to decide a suitable MLP network for a given prediction or classification problem [40]. The local minima problem can be solved by adding up more number of neurons to the hidden layers. The more number of nodes and hidden layers increases the computational complexity. Hence, there is lack of ideal methods for finding an absolute structure of MLP. Getting a perfect MLP requires long computational time, since it requires iterative testing of the MLP with various architectures and finally adopting the successful network architecture with acceptable complexity.

In an order to overcome the computational complexities associated with MLP, different researchers preferred single layer architecture i.e. FLANN, which has been proven to be superior in financial forecasting. Several basis functions have been used as functional expansion of input vectors for the network. These basis functions are sine and cosine trigonometric functions, Chebysheb function, Lagurre, Legendre, and power polynomial functions.

It is observed from the literature that along with an improved accuracy, adaptive methods with reasonable computational cost are trending areas of research in this field. For this reason different evolutionary computational models have been proposed [33 - 36]. However, forecasting to a desired accuracy is still a major concern. This paves a path to develop nonlinear adaptive structures which are computationally efficient and better forecasting ability. These issues motivated us to design an efficient model for stock index forecasting.

Hence, we propose a nonlinear adaptive neural model called as polynomial FLANN for prediction of index behavior of some well-known stock market. We added VDPs to the actual time series to increase the volume of training examples [41].

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

In the second step we considered the combination of two closing prices which is of degree two. These combinations then go through functional expansion which amplifies the dimension of the input pattern. Several polynomial and trigonometric basis functions have been used for functional expansion, rather using only trigonometric basis functions.

The paper is organized in five sections. Section 1 discussed about introduction to stock market problem and literature. The FLANN based models for financial forecasting is described in Section 2. Section 3 presents the model development. Proposed polynomial based FLANN forecasting is described by Section 4. A short description about the GA training algorithm is discussed in Section 5. Section 6 presented and discussed about the experimental results. Concluding remarks are given by Section 7 followed by a references list.

2. FLANN BASED FORECASTING

It is found that an MLP with only a single hidden layer is enough to approximate functions of any order. But in many realistic applications researchers are employed MLP with more than one hidden layer to accomplish superior generalization. MLP suffers from two well known lacunas such as high computational complexities and slow convergence. Also, to prevail over the problem of local minima additional processing nodes are required to the hidden layers. In this way, increase in layers and number of neuron to the network increases the complexity and hence increases the computational burden to the network. In MLP the succeeding layers perform a series of mapping until an ultimate representation found and preferred separation is possible. A FLANN based forecasting model is shown by Figure 1.

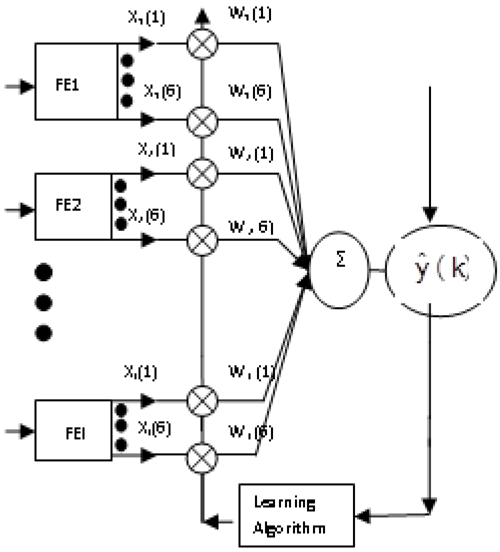

Figure 1 FLANN based forecasting

FLANN architectures are proposed by Pao [43 - 45]. This is a only one layer network with capacity of forming multifaceted decision boundaries. It belongs to the class of higher order neural networks (HONN) which used superior combination of inputs [45]. Several HONN architectures devoid of hidden layers are introduced in literature. The performance of FLANN based models are experimented and evaluated in several research works. The single layer in FLANN provides huge reduction in computation necessity and hence possesses elevated convergence rate. The initial depiction of a pattern in FLANN is described in a room of amplified dimensions. The enhanced representation of original pattern is achieved by functionally expanding each component of input vector. The functions used for expansion of input vector are subset of orthonormal functions straddling over a multidimensional representation space. In FLANN functional linkage generates several functions which are nonlinearly dependent. The input after expansion causes an augment in input pattern dimensionality which helps it in solving intricate classification task through generation of non-linear margins [2]. Intelligent models with lower complexity is proposed in [15] for the capacitor pressure sensor. Three polynomials i.e., Chebysheb, Legendre and power series are used in FLANN. They observed that it offered computational enhancement over MLP. There are quite a lot uses of FLANN starting from pattern recognition and classification [22, 26, 27, 9, 10, 39], system identification and control [17, 31], and financial forecasting [32,39] to digital channel equalization [15].

There found many applications of FLANN in financial forecasting domain. Its computational efficiency has been established [16]. Short as well as long term forecasting with FLANN using LMS and RLS has been proposed in the research works [26, 27]. Their findings are in favor of FLANN. FLANN based models also have been employed for exchange rate prediction in [27]. They compared FLANN, cascaded FLANN and LMS based models, and cascaded FLANN found to be superior to other. Trigonometric based functions are used in many research works and shown improved performance [15 -17]. The research work in [2] used Wilcoxon Artificial Neural Networks and Wilcoxon Functional Link Artificial Neural Networks (WFLANN) for prediction of currency rates. Comparative results revealed approximately indistinguishable performance. However the later implicated lower computation cost and was preferred. An improved PSO is employed to train a FLANN (ISO-FLANN) for the purpose of classification [36]. The work compared the effectiveness of the proposed model with support vector machine, MLP, FLANN with gradient descent, and fuzzy swarm net (FSN) and found to be superior.

The above discussion can be summarized as follows.

- MLP model has been employed frequently for the purpose of stock market forecasting.
- Computational complexities and black-box nature of MLP paves path towards designing single layer, less complex FLANN models for stock market forecasting.
- Combining evolutionary training algorithms with FLANN become popular in attempt to achieve better accuracy.
- All research works adopted conventional basis functions for input vector expansion.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

However, with the aim of achieving better forecasting accuracy, the key factors motivated us are:

- Importance to model design with a reduction in structural architecture and minimal computational load.
- Improving forecasting accuracy by searching the model parameters with derivative free optimization techniques.
- Using polynomial and trigonometric basis functions rather conventional basis functions alone for functional expansions.
- Inclusion of virtual data positions to actual dataset for enhanced forecasting accuracy.

3. MODEL DEVELOPMENT

This section describes about the neural based forecasting models. The models include a MLP, FLANN based models such as TFLANN, CFLANN, LFLANN, LeFLANN and the proposed polynomial FLANN. All the forecasting models are trained by GA.

3.1. Multi-layer perceptron

MLPs are the most widely implemented neural networks for stock market prediction. We considered here a MLP with one hidden layer and only one output. The input layer use linear transfer, the hidden and output layer use sigmoid activation. The first layer is the input layer and contains one node for each input signal. So the no. of neurons in this layer is same as the size of input vector. The second layer also called as the hidden layer try to capture the non-linear relationships among data points in the financial time series. The output neuron calculates the model estimation and compared with the actual output. The difference is termed as error value produced by the model. Less the error value better is the model. The root mean squared error obtained from the training samples is then used to propagate back to previous layers in order to train the MLP. The weight and other model parameters are updated based on the principle of gradient descent rule. The input layer neurons use linear activations and hidden as well as output layer neurons use sigmoid activation as in Eq. (1).

$$y_{\mathrm{output}} = \frac{1}{1 + e^{-\lambda y_{\mathrm{input}}}} \qquad (1)$$

Where, λ is sigmoid gain. The output z from a neuron j in the hidden layer is computed as follows.

$$z = f\left(B_j + \sum_{i=1}^{n} V_{ij} X_i\right) \qquad (2)$$

Where, X_i is the i^{th} input vector, V_{ij} is synaptic weight connecting i^{th} input and j^{th} hidden neuron. B_j is the bias. The output y from the output neuron is computed as:

$$y = f\left(B_0 + \sum_{j=1}^{m} W_j * z\right) \qquad (3)$$

where, W_j is the synaptic weight connecting j^th hidden neuron and output layer neuron, z is the weighted sum from previous layer, and B_0 is bias to output neuron. The error is calculated on comparing model output y and actual output. Weight and other model parameter values are fine tuned for minimal error signal generation. In this experimental work, GA is used to search such optimal parameters for the MLP considered.

Back propagation learning is follows gradient descent rule and suffers from limitations like sluggish convergence, attentive to local minima. It affects the forecasting accuracy of the model. On contrast, GA performs exploration over entire resolution space, reaches near optimal solution fairly without difficulty, and does not necessitate incessant differentiable objective function. The task of searching optimal parameters for MLP could be seen as a search methodology into the space of all potential solutions. Here weight set for input-hidden layer, weights for hidden-output layers and bias values are the model parameters. This search can be done by applying GA. The individual representation for GA is shown in Figure 2.

Weight								Bias	
Input - Hidden				Hidden - Output				Hidden	Output
V_1	V_{12}	.	V_{nm}	W_1	W_2		W_m	B_1 … ….	B_0

Figure 2 Individual representation of MLP

In this work, a chromosome of GA acts as a potential weight and bias of MLP. Set of such chromosome forms a matting pool. An input sample and a chromosome value is fed to MLP model. The fitness of an individual is evaluated as difference between the actual output and model estimation. The less the error of an individual, the better is its fitness. We represent weight values linked with input-hidden layer as V_11toV_nm and that of hidden-output layer as W_1toW_m. The biases for hidden-output layer are B_1 and B_0 respectively. The different steps of training process are:

1. Genotypes are considered as potential candidate for MLP consist of synaptic weight and bias values.
2. Model fed with finest no. of inputs. Phenotypes are extracted from optimized genotypes. Each input is weighted with the phenotypes.
3. At convergence point the immediate sample is considered as test data and fed to the model computed error

Step 1-3 are repeated for all training and testing samples and average percentage of error has been calculated.

3.2 Functional link artificial link neural network

Using four basis functions for expansion of an input value, we develop four FLANN based models for forecasting purpose. We termed these as TFLANN, LFLANN, CFLANN, and LeFLANN.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

3.2.1 Trigonometric FLANN (TFLANN)

Here we used seven trigonometric sine and cosine functions as basis functions for expansion of an input from lower to higher dimension. An input x_i is expanded to seven terms as follows:

$c_1(x_i) = (x_i, \; c_2(x_i) = \sin(x_i), c_3(x_i) = \cos(x_i), c_4(x_i) = \sin(\pi x_i), c_5(x_i) = \cos(\pi x_i), \; c_6(x_i) = \sin(2\pi x_i), \; c_7(x_i) = \cos(2\pi x_i)$

3.2.2 Chebysheb FLANN (CFLANN)

The base function here is Chebysheb polynomial hence we termed it as CFLANN. An input value x_i after expansion is shown as follows.

$c_1(x_i) = (x_i), \; c_2(x_i) = 2xi^2 - 1, \; c_3(x_i) = 4xi^3 - 3xi, \; c_4(x_i) = 8xi^4 - 8xi^2 + 1$

Higher order polynomials can be created using the recursive formula $c_{n+1} = 2 * x_i * c_n(x_i) - c_{n-1}(x_i)$

3.2.3 Lagurre FLANN (LFLANN)

The Lagurre polynomial is used to enhance an input data. We termed the model as LFLANN. An input x_i is expanded into several terms as $c_1(x_i) = -x + 1, c_2(x_i) = \frac{x^2}{2} - x + 1$. Higher order polynomials are generated recursively as $c_{n+1}(x_i) = \frac{1}{n+1} * ((2n+1)c_n(x_i) - nc_{n-1}(x_i))$.

3.2.4 Legendre FLANN (LeFLANN)

We termed this model as LeFLANN. The base functions used are Legendre polynomials. An input x_i is expanded as $c_1(x_i) = x, c_2(x_i) = \frac{(3x^2-1)}{2}, c_3(x_i) = \frac{(5x^3-3x)}{2}, c_4(x_i) = \frac{(35x^4 - 30x^2+3)}{8}$

Higher order Legendre polynomials are generated recursively as follows:

$$c_{n+1}(x_i) = \frac{1}{n+1} * ((2n+1) * x_i * c_n(x_i) - nc_{n-1}(x_i))$$

4. PROPOSED POLY-FLANN MODEL

FLANN architecture applies a set of base functions on each input for expansion. The functional linkage acts on complete pattern through generating a set of functions which are linearly independent. The functional expansion block of FLANN causes amplification in the input space dimensionality. This makes FLANN more capable to unravel complex problems in terms of generating non-linear decision margins.

Keeping objective of achieving more forecasting accuracies, we propose a new method which incorporates VDPs to the original dataset and considering the combination of two closing prices for functional expansion. In this way we achieve more forecasting accuracy by the proposed model in comparison to the traditional FLANN model. The proposed model consists of five basic steps. The following Figure 3 shows the proposed Poly-FLANN based forecasting.

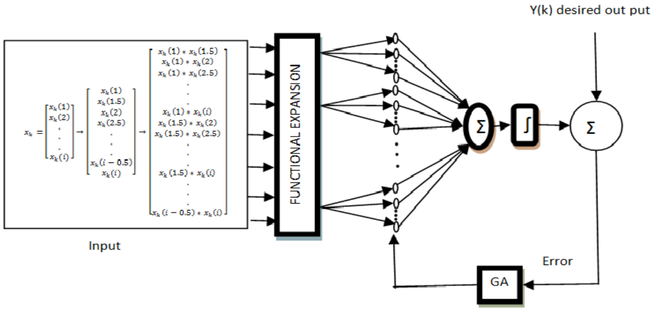

Figure 3 Poly-FLANN based forecasting model

a) Virtual data positions exploration and addition

This section describes few approaches to explore VDPs for original financial time series. The daily closing prices of a stock market is represented as a financial time series as follows.

$$p(1), p(2), p(3), \dots \dots \dots \dots \dots \dots, p(n)$$

The process of modeling and forecasting a future price is based on analysis of present and past data points represented as follows:

$$p(n+k) = f(p(n-k), \dots, p(n))$$

Here p (n + k) is desired closing price. We represent the search space and respective targets as follows.

$$\begin{array}{cccc|c} p_i(1) & p_i(2) & \cdots & p_i(m) & p_i(m+k) \\ p_i(2) & p_i(3) & \vdots & p_i(m+k) & p_i(m+k+1) \\ \cdots & \cdots & & \cdots & \cdots \\ p_i(n-k-m) & p_i(n-k-m+1) & \cdots & p_i(n-k) & p_i(n) \end{array}$$

Here n, m, and k represent the total number of closing prices in the dataset, window size, and forecast step respectively. The window is moving one step to right over the series and the number of data points incorporated within it is considered as one input sample. We introduced VDPs to the original series with the hope of achieving higher accuracy. We followed linear interpolation techniques as in Eq. 6 for exploration of such VDPs [41].

$$p(i+0.5) = p(i) + \frac{1}{2}\big(p(i+1) - p(i)\big) \qquad (6)$$

The $p(i+0.5)^{th}$ VDP is calculated considering two adjoining data points on the time series. The time series after VDP incorporation is as follows:

$$p(1), p(1+0.5), p(2), p(2+0.5) \dots \dots, p(n-0.5), p(n)$$

B) Combination of inputs values

Suppose the moving window size has been selected as n, i.e. there are n numbers of closing prices constitute one training sample. Suppose the i^{th} training example is represented as follows.

$$p_i(1), p_i(2), \dots \dots \dots \dots \dots .., p_i(n)$$

Taking combination of two closing prices, the training example converted to another expanded vector which can be represented as follows.

$$p_i(1) * p_i(2), p_i(1) * p_i(3), \dots, p_i(1) * p_i(n), p_i(2) * p_i(3), p_i(2) * p(4), \dots, p_i(2) * p_i(n), p(n-1) * p_i(n)$$

Hence, if the window size is n, the number of combinations m can be represented as follows.

$$m = \frac{n*(n-1)}{2}$$

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

c) Functional expansion of input elements

The functional expansion units expand input signals in a nonlinear fashion to create several values. Expansion component produces a set of functions which are linearly independent. This effect of nonlinearity added to input pattern, trim down the need of hidden layers and minimizes computational involvedness in consequences. The expanded value of an input $x_i x_j$ (after combination of x_i and x_j) using polynomial expansion can be represented as:

$x_i x_j$, $(x_i x_j)^2$, $\sin(x_i x_j)$, $\sin(\pi\,(x_i x_j))$, $\sin(2\pi(x_i x_j))$, $\cos(x_i x_j)$, $\cos(\pi(x_i x_j))$, $\cos(2\pi(x_i x_j))$

d) The estimation process

Responding to the nonlinear input elements, the adaptive forecasting model computes the output, which we termed as estimated closing prices. By comparing this estimated closing price with the corresponding desired closing price, an error signal is generated.

e) Adaptive process

This process involves finding an optimal connection weight and bias set whose anticipated value possibly will come close to the Wiener solution W_0 [4]. We used GA to search out the best possible weight and bias set.

The foremost steps of poly-FLANN model can be summarized as follows.

1. Setting training data.
2. Exploration and incorporation of VDPs.
3. Form all combination of input features.
4. Inflate lower input pattern space to higher dimension.

/* *Training Phase* */

5. Initialization of search spaces randomly
6. While (termination criteria = false)

For each chromosome evaluate weighted sum
Feed to output neuron
Supply target, compute error
Evaluate fitness
End
Apply crossover operation
Apply mutation operation
Select healthier chromosomes
End

/* *Testing Phase* */

7. Present the test data, calculate error signal

Do again the steps 1-7 for all training and testing patterns, and record sum error

5. GENETIC ALGORITHM

Genetic algorithms are worldwide search technique works on a inhabitants of probable solutions. The individual of GA is represented as chromosomes. The optimal solution can be obtained through a course of artificial evolution. It applies genetic operations such as estimation, selection, crossover, and mutation. For application of GA in financial prediction prospective readers may follow articles in literature [34, 35, 39]. The basic steps are as follows:

1. Search space initialization
2. Fitness evaluation
3. Selection of better fit individual.
4. Crossover operation.
5. Mutation operation.

The FLANN model used exhibits distinctiveness of both ANN and GA, hence forming a hybrid model. First, we identified a set of connections with a fixed quantity of input and a only one output as shown in Figure 1. GA is used in the learning phase then to search the optimal parameters.

6. EXPERIMENTAL DETAILS AND RESULTS

a) Preparation of input data

We collected the closing prices of five stock exchanges from https://in.finance.yahoo.com/quote/. The stock indices are summarized in Table 1.

Table 1: Stock market datasets

Stock Index	Type	Period
BSE	Daily closing price	2nd Jan - 2017 to 31st Dec - 2017
DJIA	Daily closing price	3rd Jan - 2017 to 31st Dec -2017
NASDAQ	Daily closing price	3rd Jan - 2017 to 31st Dec -2017
FTSE	Daily closing price	3rd Jan - 2017 to 31st Dec - 2017
TAIEX	Daily closing price	2rd Jan - 2017 to 28th Dec - 2017

For robust learning of ANN models, we normalized original data by means of sigmoid normalization method as follows:

$$\hat{a} = \frac{1}{1+e^{a}} \qquad (7)$$

b) Experimental setting

In this section we show the forecasting results generated by models. The MAPE is used as the evaluation metric to compare the efficiency of the models under experimentation which is given as follows:

$$\mathrm{MAPE} = \frac{1}{N}\sum_{i=1}^{N}\frac{|\mathrm{actual}_i - \mathrm{estimated}_i|}{\mathrm{actual}_i} \times 100\% \qquad (8)$$

Let the k^{th} input sample is $X_i = [x_i(1), x_i(2), \dots, x_i(m)]$. The virtual data positions are calculated by linear interpolation of two consecutive feature values and added to the original input. The expanded input pattern is represented $X_i' = [x_i(1), x_i(1.5), x_i(2), x_i(2.5), \dots, x_i(m-0.5), x_i(m)]$. Combination two data from the expanded input sample X_i' , a new list is generated as $X_i'' = [x_i(1) * x_i(1.5), x_i(1) * x_i(2), \dots, x_i(1) * x_i(m), \dots\dots, x_i(m-0.5) * x_i(m)]$.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

Each element of X_i'' is then applied to the functional expansion block. Different polynomial functions are applied as base function and further expanded vector is represented as $\bar{X}_i^T$. When each element $x_k(i)$is applied for functional expansion, then the expanded values can be represented by

$$\emptyset(x_k(i), x_k(i+1)) = \{x_k(i) * x_k(i+1), (x_k(i) * x_k(i+1))^2, \sin(n\pi x_k(i) * x_k(i+1)), \cos(n\pi x_k(i) * x_k(i+1))\}.$$

Given $\overline{X_k}^T$as input, the model estimates $\hat{y}$ (k). The model output is computed as follows:

$$y'^{(i)} = \bar{X}_i^T * \text{Weight}(i) + \text{bias} \quad (9)$$

This output is passed through sigmoid activation to produce $\hat{y}(i)$.

$$\hat{y}(i) = \frac{1}{1 + e^{-\lambda y'i}} \quad (10)$$

The error signal $err(i)$ is computed as:

$$err(i) = y(i) - \hat{y}(i) \quad (11)$$

The input vector and error function are used by the learning algorithm to explore the best possible weight and bias vector. We used a genetic algorithm to learn the search space efficiently as well as overcome the limitations of back propagation learning. During the training process, training samples and weight values are repeatedly presented to the FLANN model. The model parameters are adjusted by GA till appropriate input-output mapping. The sum of error values from all training patterns is calculated as in Eq. 12. The objective is to minimize this error function.

$$E(i) = \sum_{i=1}^{N} err(i) \quad (12)$$

As GA works on encoded parameter, we used binary encoding here. We considered gene size of 17 bits and a population of 40-60 genotypes is randomly initialized. The maximum generations in a range of 200-300 are considered for different models. For selection of better fit chromosomes we choose an elitism method (10% best parent + binary tournament selection). We considered uniform crossover of parents followed by operation of mutation. The crossover and mutation probability are in the range of 0.5-0.6 and 0.01 - 0.05 respectively. The offspring generated replaces the current population. The process of updating population continues till convergence. In successive generation, suitability of the best and average chromosomes increased towards global optimum. The parameters simulated during experiment for different models are summarized in Table 2.

Table 2 Simulated model parameters

Parameters	Forecasting model					
	MLP	TFLANN	CFLANN	LFLANN	LeFLANN	POLY-FLANN
Population size	60	40	40	40	40	50
Gene Size (bit)	15	15	15	15	15	15
Cross. Prob(Cp)	0.6	0.5	0.5	0.5	0.5	0.5
Mutation Probability (Mc)	0.02	0.01	0.02	0.01	0.01	0.05
Selection method	Elitism	Elitism	Elitism	Elitism	Elitism	Elitism
Maximum number of generation	300	200	200	200	200	250

The experiments are carried out in MATLAB-2015 environment, with Intel ® core ¬TM i3 CPU, 2.27 GHz processing and 2.42 GB memory size.

c) Result analysis

We discuss the outcome from the experimental work here. To avoid the biasness in model performance, we fed same training and test patterns to all models. To stay away from the stochastic nature of the ANN models, we measured the mean of ten simulations. The MAPE values generated by the models are presenter in Table 3.

It is clearly seen that proposed Poly-FLANN forecasting model generates lowest MAPE values for all datasets. The MLP model produced an average MAPE of 0.396221 over all five datasets. The TFLANN generated 0.067341 MAPE considering all datasets. Similarly, CFLANN generated 0.112414, LFLANN generated 0.056382, and LeFLANN generated 0.06264 average MAPE values over five datasets. However, the proposed Poly-FLANN generated average MAPE of 0.037323 over all datasets which is quite better than other models. To find out the accurate advantage of using Poly-FLANN over others, we calculate the MAPE gain as follows.

$$\text{MAPE gain} = \frac{(\text{MAPE of existing model} - \text{MAPE of proposed model})}{\text{MAPE of existing model}} \times 100\%$$

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

The MAPE gain of the proposed approach over MLP, TFLANN, CFLANN, LFLANN, and LeFLANN considering five datasets are shown in Figure 4.

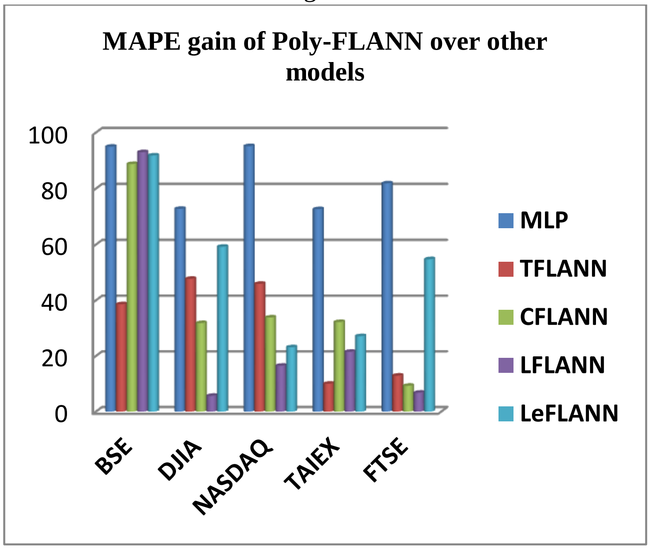

Figure 4 MAPE gain by Poly-FLANN over other models in five stock datasets

Further, to show the advantages of the proposed model, we conducted a statistical significance test, i.e. Deibold-Mariano (DM) test [24]. DM Test is a pair wise comparison test of time series models. The computed DM statistics are presented in the Table 4. It is clear that DM statistics are laying outer surface of critical range which supports rejection of null hypothesis.

Table 3: MAPE values of six models from five financial time series

DATA SET	MLP	FLANN				Poly-FLANN
		TFLANN	CFLANN	LFLANN	LeFLANN	
BSE	0.304335	0.056428	0.315736	0.056205	0.084752	0.015375
DJIA	0.056225	0.06608	0.063875	0.038371	0.039651	0.034525
NASDAQ	0.426727	0.070336	0.072455	0.070725	0.052855	0.04794
TAIEX	0.756944	0.056225	0.063502	0.06756	0.056903	0.053032
FTSE	0.436873	0.087635	0.0465	0.049048	0.079037	0.035745

Table 4: Computed DM statistic values

Stock Data		TFLANN	CFLANN	LFLANN	LeFLANN	MLP
BSE	Poly-FLANN	2.3132	1.9847	2.3140	2.0263	2.0545
DJIA		1.9085	2.5330	2.0515	-3.1580	-2.1568
NASDAQ		2.0066	2.4163	-2.4291	3.1392	2.6015
TAIEX		-2.4267	2.2655	1.9972	-2.5823	-3.2413
FTSE		2.6072	2.0575	-2.4832	2.0675	-3.5835

6. CONCLUSIONS AND FUTURE SCOPE

A computationally less complex and novel GA based poly-FLANN forecasting model has been introduced in this paper. The most frequent MLP model and variants of conventional FLANN models such as TFLANN, CFLANN, LFLANN and LeFLANN have been investigated in this experimental work. All these models were employed to forecasting the 1-day-ahead daily closing prices of five stock exchanges. The proposed poly-FLANN model is based on trigonometric and polynomial order of input vectors. The concept of incorporating VDPs to the original financial time-series has been adapted which increases the volume of training patterns. All the forecasting models were trained by a genetic algorithm. The conventional FLANN models were found to be more suitable than the MLP in terms of generating less error signals. However, the proposed poly-FLANN model found to be superior to the MLP as well as the conventional FLANN models. The DM assessment justified the statistical consequence of the model. The work can be extended using other neural networks and search algorithms. Also the proposed model can be used for other time series datasets.

REFERENCES

1. Abdoh, T. H., & Jouhare, H. (1996). The investigation of efficiency of stock price index of TSE. Journal of Financial Research, 13(11-12).
2. Majhi, B., Shalabi, H., & Fathi, M. (2005). FLANN based forecasting of S&P 500 index. Information Technology Journal, 4(3), 289-292..
3. Majhi, B., Rout, M., Majhi, R., Panda, G., & Fleming, P. J. (2012). New robust forecasting models for exchange rates prediction. Expert Systems with Applications, 39(16), 12658-12670..
4. B. Widrow, and S. D. Stearns, " Adaptive Signal Processing," Pearson Education, 2002.
5. Box G.E.P., Jenkins G. M. (1976), Time Series Analysis-Forecasting and Control, Holden-Day Inc., San Francisco.
6. Cao, Q., Leggio, K. B., & Schniederjans, M. J. (2005). A comparison between Fama and French's model and artificial neural networks in predicting the Chinese stock market. Computers & Operations Research, 32(10), 2499-2512.
7. Chen, A. S., Leung, M. T., & Daouk, H. (2003). Application of neural networks to an emerging financial market: forecasting and trading the Taiwan Stock Index. Computers & Operations Research, 30(6), 901-923..
8. Calderon, T. G., & Cheh, J. J. (2002). A roadmap for future neural networks research in auditing and risk assessment. International Journal of Accounting Information Systems, 3(4), 203-236..
9. Dehuri, S., & Cho, S. B. (2010). Evolutionarily optimized features in functional link neural network for classification. Expert Systems with Applications, 37(6), 4379-4391..
10. Dehuri, S., & Cho, S. B. (2010). A hybrid genetic based functional link artificial neural network with a statistical comparison of classifiers over multiple datasets. Neural Computing and Applications, 19(2), 317-328.
11. Huang, C. J., Yang, D. X., & Chuang, Y. T. (2008). Application of wrapper approach and composite classifier to the stock trend prediction. Expert Systems with Applications, 34(4), 2870-2878.
12. Harvey, C. R., Travers, K. E., & Costa, M. J. (2000). Forecasting emerging market returns using neural networks. Emerging Markets Quarterly, 4, 43-54.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

13. Nayak, S. C., Misra, B. B., & Behera, H. S. (2018). ACFLN: artificial chemical functional link network for prediction of stock market index. Evolving Systems, 1-26.
14. Patra, J. C., & Van den Bos, A. (2000). Modeling of an intelligent pressure sensor using functional link artificial neural networks. ISA transactions, 39(1), 15-27.
15. Patra, J. C., Pal, R. N., Baliarsingh, R., & Panda, G. (1999). Nonlinear channel equalization for QAM signal constellation using artificial neural networks. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 29(2), 262-271.
16. J. C. Patra, W. Kim, P. K. Meher, E. L. Ang, "Financial prediction of major indices using computational efficient artificial neural networks", IJCNN, Vancouver, Canada, pp. 2114-2120, 2006.
17. J. C. Patra, R.N. Pal, B.N. Chatterji, G. Panda, Identification of nonlinear dynamic systems using functional link artificial neural networks, IEEE Trans. Syst. Man Cybern. B 29 (2) (1999) 254–262.
18. Kumar, K., & Bhattacharya, S. (2006). Artificial neural network vs. linear discriminant analysis in credit ratings forecast. Review of Accounting and Finance, 5, 216–227.
19. Leigh, W., Hightower, R., & Modani, N. (2005), "Forecasting the New York stock exchange composite index with past price and interest rate on condition of volume spike," Expert Systems with Applications, 28, 1–8.
20. L. Yu, and Y. Q. Zhang, "Evolutionary Fuzzy Neural Networks for Hybrid Financial Prediction," IEEE trans. on Systems, Man, and Cybernetics—Part C: Applications and Reviews, vol. 35, no. 2, May 2005.
21. Liu, H.-C., Lee, Y.-H., Lee, M.-C., 2009, "Forecasting china stock markets volatility via GARCH models under skewed-GED distribution," Journal Money Investment Banking, 5–14.
22. Mishra, B.B., Dehuri, S., 2007. Functional link artificial neural network for classification task in data mining. Journal of Computer Science 3, 948–955.
23. Mishra, B.B., Dehuri, S., Panda, G., Dash, P.K., 2008. Fuzzy swarm net (FSN) for classification in data mining. The CSI Journal of Computer Science and Engineering 5 (2&4(b)), 1–8. McGrath, C. (2002). Terminator portfolio. Kiplinger's Personal Finance, 56, 56–57.
24. S. C. Nayak, Misra, B. B. (2018), 'Estimating stock closing indices with A GA-Weighted polynomial neural network', Financial Innovation.
25. Patra, J. C., Panda, G., & Baliarsingh, R. (1994), "Artificial neural network based nonlinearity estimation of pressure sensors," IEEE Transactions on Instrumentation and Measurement, 43(6), 874–881.
26. R. Majhi, G. Panda and G. Sahoo, "Development and performance evaluation of FLANN based model for forecasting of stock markets", Expert Systems with Applications , vol. 36, pp. 6800–6808, 2009
27. R. Majhi, G. Panda and G. Sahoo,"Efficient prediction of exchange rates with low complexity artificial neural network models," Expert Systems with Applications, vol. 36, pp. 181-189, 2009.
28. R. Majhi, B. Majhi, and G. Panda, "Development and performance evaluation of neural network classifiers for Indian internet shoppers," Expert Systems with Applications, vol. 39, pp. 2112-2118, 2012.
29. R. R. Trippi, and D. DeSieno, "Trading equity index futures with a neural network," Journal of Portfolio Manangment, vol. 19, pp. 27-33, 1992.
30. Ravichandran K. S., Thirunavukarasu P., Nallaswamy R., Babu R. (2007), "Estimation on return on investment in share market through ANN", Journal of Theoretical and Applied Information Technology, vol. 3, pp. 44-54.
31. S. Purwar, I. N. Kar, and A. N. Jha, "On-line system identification of complex systems using Chebyshev neural networks," Applied Soft Computing, vol. 7, pp. 364-372, 2007.
32. S. C. Nayak, Misra, B. B., and Behera, H. S. (2017), 'Efficient financial time series prediction with evolutionary virtual data position exploration', Neural Computing and Applications, https://doi.org/10.1007/s00521-017-3061-1,.
33. S. chakravarty, and P. K. Dash, "A PSO based integrated functional link net and interval type-2 fuzzy logic system for predicting stock market indices," Applied Soft Computing, Vol. 12, pp. 931-941, 2012.
34. S. C. Nayak, B. B. Misra, and H. S. Behera, "Index Prediction Using Neuro-Genetic Hybrid Networks: A Comparative Analysis of Perfermance," International Conference on Computing Communication and Application, IEEE Xplore, 22-24 Feb.2012, doi: 10.1109 / ICCCA.2012.6179215.
35. S. C. Nayak, B. B. Misra, and H. S. Behera, "Stock Index Prediction With Neuro-Genetic Hybrid Techniques," Intenational Journal of Computer Science and Informatics, IJCSI, vol. 2, Issue-3, pp. 27-34, 2012.
36. S. Dehuri, R. Roy, S. B. Cho, and A. Ghosh, "An improved swarm optimized functional link artificial neural network (ISO-FLANN) for classification," The Journal of Systems and Software, Vol. 85, pp. 1333-1345, 2012.
37. Swicegood, P., and Clark, J. (2001). Off-site monitoring systems for prediction bank underperformance: A comparison of neural networks, discriminant analysis, and professional human judgment. International Journal of Intelligent Systems in Accounting, Finance and Management, 10, 169–186.
38. T. Kimoto, K. Asakawa, M. Yoda, and M. Takeoka, "Stock market prediction system with modular neural network," Proceedings of the International Joint Conference on Neural Networks, pp. 1-6, San Diego, 1990.
39. S. C. Nayak, B. B. Misra, and H. S. Behera, (2015) 'An adaptive second order neural network with genetic algorithm based training (ASONN-GA) to forecast the closing prices of the stock market, Int. Journal of Applied Metaheuristic Computing.
40. Wang, Y. (2003). Mining stock prices using fuzzy rough set system. Expert System with Applications, 24, 13–23.
41. S. C. Nayak, Misra, B. B., and Behera, H. S. (2014), 'Exploration and Incorporation of Virtual Data Position for Efficient Forecasting of Financial Time Series', Int. Journal of Industrial and Systems Engineering, Vol. 26, No. 1.
42. Y. K. Kwon , B. R. Moon, "A Hybrid Neuro Genetic Approach for Stock Forecasting," IEEE trans. on Neural Networks, vol. 18, no. 3, May 2007.
43. Y. H. Pao, " Adaptive Pattern Recognition and Neural Networks," Addision-Wesley Inc., 1989.
44. Y. H. Pao, "Adaptive Pattern Recognition and Neural Networks, Addison-Wesley, Reading, MA, 1989.
45. Y. H. Pao and Y. Takefuji," Functional-link net computing: thory, system architecture, and functionalities," Computer, Vol. 25, pp. 76-79, 1992.

*Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication*