Ga based stochastic optimization for stock price forecasting using fuzzy time series hidden markov model

Article · January 2018

DOI: 10.12732/ijpam.v117i1.13

G. Kavitha
Hindustan University

International Journal of Pure and Applied Mathematics
Volume 117 No. 1 2017, 143-171
ISSN: 1311-8080 (printed version); ISSN: 1314-3395 (on-line version)
url: http://www.ijpam.eu
doi: 10.12732/ijpam.v117i1.13

ijpam.eu

GA BASED STOCHASTIC OPTIMIZATION FOR STOCK PRICE FORECASTING USING FUZZY TIME SERIES HIDDEN MARKOV MODEL

G. Kavitha[1 §], A. Udhayakumar[2]
[1,2]Department of Mathematics
Hindustan Institute of Technology and Science
Tamil Nadu, 603103, INDIA

Abstract: This paper presents GA based stochastic optimization for Hidden Markov Model (HMM) combined with Fuzzy Time Series (FTS) for forecasting stock price with accuracy. The proposed model is adopted to realize the probabilistic state transition involving time evolution in a probabilistic system. The parameters of the HMM model are calculated in the first phase for initialization. In the second phase, the initial parameters are fed into HMM in MATLAB for estimation and they are optimized by the GA optimization technique in MATLAB. The method is tested on the datasets of several stocks of the National Stock Exchange of India Limited (NSE) and New York Stock Exchange (NYSE). The different forecasting errors namely, Mean Square Error (MSE), Root Mean Square Error (RMSE) and Mean Absolute Percentage Error (MAPE) for various stocks of NSE and NYSE are found out. The results of directional accuracy average for various NSE stock indices from January to August 2014 shows the prediction quality with a best MAPE of 1.3% for INFOSYS, 1.71% for TCS, 1.16% for HCLTECH and 1.09% for WIPRO. Furthermore, the performance of NYSE stocks was tested for validation and compared to some of the existing methods. It was found that the model provides a good prediction performance with a best MAPE of 0.7629% for IBM, 1.8% for APPLE, 0.9110% for DELL stocks getting satisfactory quality solutions showing forecasting efficiency.

AMS Subject Classification: 60Gxx
Key Words: hybrid intelligent systems, computational linguistics, parameter estimation, genetic algorithms, forecast uncertainty

Received: 2017-08-04
Revised: 2017-10-30
Published: November 29, 2017

§Correspondence author

1. Introduction

Hybrid intelligent features are employed for designing an efficient architecture in projecting the stock price forecast based on bivariate FTS and HMM. For stock price forecasting, both mathematics and financial background are necessary. FTS as well as HMM work on the concept of solving problems involving time, uncertainty, and reduction in error forecasting. The reason for developing such models is that it deals with money. The idea behind this approach is that history repeats itself to a certain probability. It deals with predicting the next day numerical close price of the stock market. Prediction errors are always expected during quantitative stock analysis. The model proposed here aims at optimizing the variable parameters of HMM and helps in minimizing the error.

FTS models deal with analysis of linguistic features of large fluctuating historical stock market data. They have the ability to react to situations of dynamic, non-linear data of high uncertainty. In addition to FTS, HMM is also a powerful probability tool capable of handling bi-factor forecasting problems using linguistic labels. HMM are developments of Markov models where the observations are the outcomes of unobserved states denoted by linguistic state symbols. Using HMM, a fuzzy relationship matrix can be built utilizing the fully available information, providing high accuracy.

GAs are search algorithms based on the concept of Survival of the fittest. GA optimization is a natural selection of the biological evolutionary process. GAs find global solutions to a problem with population of solutions. GA involves two operational phases, namely, genospace and phenospace to perform its genetic operations and function evaluations on individuals. The proposed model uses GA for HMM parameter optimization.

The main purpose of this study is to develop a hybrid model combining HMM, GA and FTS in an efficient way for stock price prediction for one day ahead with close price of the daily stock indices of the dataset considered. The advantage of the model is that the parameters of the HMM are initially calculated by classical HMM model, randomly using a linguistic fuzzy relationship between the transformed values. The classical HMM parameters are then estimated through HMM MATLAB Algorithms. Finally, the estimated values of the parameter are optimized using GA MATLAB tool for accurate prediction of the next day close price of any dataset of the stock indices resulting in a unique mathematical structure. The significance of the proposed model is that the observations are generated by FTS with a triangular membership function controlled by an optimized GA in HMM.

There are five sections in this paper. Section 2, provides a brief literature survey. Stock price forecasting in mathematical finance based on hybrid FTS and GA optimized HMM is illustrated in Section 3. Experimental results, data description and setup, overall performance evaluation, forecasting accuracy details are discussed in Section 4. A brief conclusion is presented in Section 5.

2. Review of Literature

The Fuzzy set theory was first proposed by Zadeh in 1965. The concept of fuzzy sets employed by Song and Chissom is capable of dealing with noisy data represented as linguistic values under circumstances of uncertainty ([24], [25]). In recent years, FTS has been widely used for forecasting non-linear and noisy data. Many models based on the concept of FTS have been developed by researchers for forecasting stock indices. Yu et al. ([30], [31], [32]) have proposed various FTS models for improvement of forecasting for Taiwan Futures Exchange (TAIEX). Chen et al. ([6], [1], [2], [3], [5], [7]) have developed various forecasting models for TAIEX based on FTS, fuzzy variation groups, fuzzy clustering, fuzzy rule interpolation technique, particle swarm optimization techniques and support vector machines. Wei et al. ([29]) have proposed a hybrid ANFIS based on n-period moving average model to forecast TAIEX stocks. K.H. Huarng et al. ([13]) have proposed a multivariate heuristic model for fuzzy time-series forecasting. Chen et al. ([4]) have proposed a hybrid fuzzy time series model based on granular computing for stock price forecasting. Guo et al. ([8]) have proposed a feature fusion based forecasting model for financial time series. Sullivan and Woodall ([27]) set the forecasting FTS model with a time invariant Markov using linguistic labels with probability distributions, normalizing all values of rows of membership matrix to unity. To overcome the limitation of involvement of one variable, a hybrid bivariate model is developed combining optimized GA with FTS and HMM.

HMM defined by Rabiner ([19], [20]) tackles the situation of forecasting the probability of the state caused by previous states. Hassan et al. ([9], [10], [11], [12]) have proposed analysis of stocks prices and predicting time series phenomena's using HMM, fuzzy model and extended their work with a fusion model of HMM, Artificial Neural Network (ANN) and Genetic Algorithm (GA) for forecasting. This study investigates the effectiveness of a hybrid approach. The concept of entropy to measure the degrees of fuzziness has been developed by Tsaur ([28]). Lee ([15]) has predicted the temperature and forecasted TAIEX on the basis of high - order fuzzy logical relationships and genetic simulated annealing techniques. Stevenson ([26]) has proposed FTS forecasting using percentage change. Ling et al. ([16]) have proposed temperature forecasting and TAIFEX forecasting based on fuzzy relationships. Jyoti Badge ([14]) has predicted stock market prices using the multivariate higher order FTS model. Qisen Cai ([17]) have developed a novel forecasting model based on FTS and a genetic algorithm. Rafal Weron ([21]) has proposed electricity price forecasting: A review of the state-of-the-art with a look into the future. Saerom Park et al. ([22]) have demonstrated prediction of market impact costs using nonparametric machine learning models. ShashankIyer et al. ([23]) have proposed a stock market prediction device using digital signal processing models. Yun et al. ([33]) have introduced stock market trading rules based on the biclustering method. Zhiqiang et al. ([34]) have proposed a stock market forecasting model combining a two-directional two-dimensional principal component analysis and a radial basis function neural network. Qisen Cai ([18]) has developed a new FTS forecasting model combined with ant colony optimization and autoregression.

The existing literature survey reveals different types of techniques like artificial neural network, fuzzy logic, genetic algorithm, particle swarm optimization, ant colony optimization, HMM, FTS which have been used for developing stock market prediction models for various stock indices. These models have their own merits and demerits in solving problems of various domains. Recent advancements in research have provided new insight in the forecasting of financial crisis. According to the present study, a hybrid bivariate model combining FTS with GA optimized HMM using MATLAB for training, testing and validation is used to analyse and forecast the stock price prediction for one day ahead of close value of the daily stock indices for the test dataset. In earlier studies, GA has been used to optimize the initial parameters of HMM choosing the number of states as the number of attributes in the observation vectors. The highlight in the proposed study is the involvement of GA efficiently in optimizing HMM variable parameters choosing different number of states and attributes in the observation vector involved in simulations. The goal is to generate a better parameter population instead of random parameter populations in GA. Some efforts has been made to identify an efficient input population parameter using GA optimization in MATLAB to obtain a desirable output. The main objective is to develop a hybrid model for accurate forecasting with a reduced mean square error giving a better quality solution compared to the other existing models. These features make the model well suited for financial market applications.

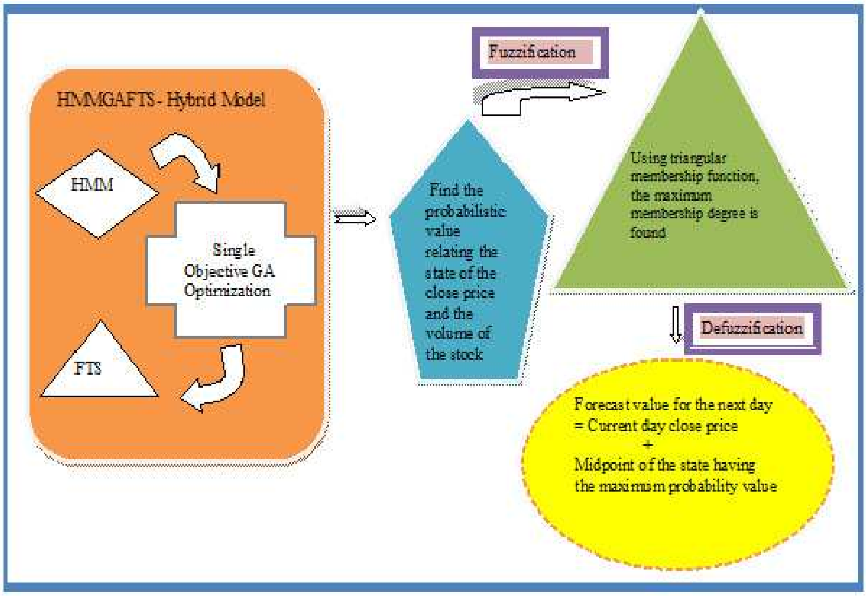

Figure 1: Block diagram of the proposed hybrid model.

3. The Stock Price Forecasting in Mathematical Finance Based on Single Objective GA Optimization for Hybrid FTS and HMM

To forecast various stock price indices efficiently, a hybrid bivariate model of FTS, HMM and GA is developed here in the proposed study. Since it is a bivariate model HMM combines with FTS to perform on a better scale. The performance mainly depends on the classical HMM variable parameter initialization. In the proposed hybrid model, GA is employed in MATLAB to fit the initial HMM parameters by optimization algorithm procedure which is described below in detail. FTS is employed for its linguistic nature, which effectively handles the bivariate historical noisy data. This hybrid model uses rules of triangular membership function for fuzzification process. Then calculate and defuzzify the desired outputs of forecast selecting the midpoint of the corresponding interval having the maximum membership degree. This value is added to the current day close price. The obtained value is the forecast value of close price for the next day. Fig. 1 shows the block diagram of the proposed hybrid model. The technical details of the various phases involved in the proposed study are described in detail below.

3.1. Single Objective Optimization Tuning of GA with HMM

Classical HMM is characterized by the following elements:

★ N_s, the number of states in the model. The state at time t is denoted as $n_s(t)$.

★ M_{osm}, the number of distinct observation symbols in each state. $m_{osm}(t)$ denote the observation symbol at time t.

★ L_{osq}, the length of the observation sequence. The state sequence can be written as $\{n_{s_1}, n_{s_2}, \ldots, n_{s_{L_{osq}}}\}$ and the observation sequence would be $\{m_{osm_1}, m_{osm_2}, \ldots, m_{osm_{L_{osq}}}\}$.

★ A set of transition probability $TPM = \{tpm_{ij}\}$, where $tpm_{ij} = P[n_{s_{t+1}} = j/n_{s_t} = i]$, $1 \leqslant i, j \leqslant N_s$. Also $\{tpm_{ij}\}$ subjects to the probability constraint equations $tpm_{ij} \geq 0$, for all $1 \leqslant i, j \leqslant N_s$ and $\sum_{j=1}^{N_s} tpm_{ij} = 1$, for all $1 \leqslant i \leqslant N_s$.

★ The observation symbol probability, also called emission probability distribution in state 'i' is given by $EPM = \{epm_i(m_{osm_m})\}$, where

$$epm_i(m_{osm_m}) = P[m_{osm_m}(t)/n_s(t) = i], 1 \leqslant i \leqslant N_s, 1 \leqslant m_{osm} \leqslant M_{osm}$$

and m_{osm_m} is the m^{th} symbol in the observation vector.

★ The initial state distribution $\pi = \pi(i)$, where $\pi_i = P(n_{s_1} = i)$, $1 \leqslant i \leqslant N_s$.

From the definitions above, it is clear that a complete specification of a HMM involves three model parameters (N_s, M_{osm} and L_{osq}) and three sets of probability parameters (TPM, EPM and π). For convenience, a compact notation λ = (TPM, EPM and π) is used to represent the complete set of parameters of the classical HMM model.

The proposed study of stock market forecasting is based mainly on the analysis of the bivariate attributes, namely, the daily close price and volume of the stock. The terminologies of the bivariate attributes considered are defined as below:

— Close price - This is the last price that the security traded during the day.

— Stock Volume - This represents the number of shares that were traded during the period.

Let 'N_c' denote the number of days of the close price data taken for training, validation or testing phase. Let 'N_v' denote the number of days of the volume traded data taken for training, validation or testing phase. Let 'O_{diff_c}' denote the one day difference in close price data for the dataset considered. The height for the close price data is calculated using the formula $h_c = \frac{u_c - l_c}{n_c}$, where 'u_c' denotes the upper bound of all close price values, 'l_c' denotes the lower bound of all close price values, 'n_c' denotes the number of subintervals of the close price value. The close price data's are divided into 'n_c' subintervals using the height denoted by 'h_c'. The first subinterval of the close price data is denoted by 'n_{c_1}'. The second subinterval of the close price data value is denoted by 'n_{c_2}'. The n^{th} subinterval of the close price data is denoted by 'n_{c_n}'. For 'n_{c_n}' number of subinterval partitions, there are 'n_{cs_n}' number of corresponding states. The states are given linguistic description. Each State has a midpoint denoted by a vector $\{m_{cs1}, m_{cs2}, \ldots, m_{cs_n}\}$. The midpoints obtained are considered as linguistic hidden states. Forecast accuracy is sensitive to selected interval partitions. The parameters are calculated using classical HMM.

The initial values of the parameters namely, Initial Transition Probability Matrix (ITPM), Initial Emission Probability Matrix (IEPM) and steady state probability ($I\pi$) are determined using classical HMM in phase I described above. While classical HMM is implemented the difficulty lies in choosing the number of states, number of observation symbols, number of input features to be considered making it difficult for the model to perform. The study involves the basic classical HMM model to calculate the input parameters. The parameter estimation is done by HMM Algorithm in MATLAB using MATLAB code in phase II.

Phase II Algorithm:

$[ITPM, IEPM]$ = hmmestimate (seq, states)
Seq1 = hmmgenerate (100, $ITPM$, $IEPM$);
Seq 2 = hmmgenerate (200, $ITPM$, $IEPM$);
Seqs = Seq 1, Seq 2;
$[ETPM,\ EEPM]$ = hmmtrain (Seqs, $ITPM$, $IEPM$,
'Maxiterations', 1000)

The proposed model deals with the issue of parameter optimization. The model aims at showing how GA can be employed for improving the performance and the efficiency of computerized trading systems to get the best HMM architecture. The trained HMM is then passed as an initial matrix for optimizing the HMM parameters during the simulation process in GA. The user defined parameters are taken as initial population to be fed to GA during simulation. This is an advantage of this model over considering a random population for the optimization process. The model gives better prediction compared to earlier models, when the initial population parameter values are user defined classical HMM concept. The number of variables for optimization depends upon the number of subintervals which is once again user defined, confined to the problem environment. Usually the number of variables is taken according to the number of attributes as in Hassan ([11]). Here the proposed model employs a different dimension for variables in comparison with the attributes. For smaller number of intervals, the convergence rate is also low, avoiding further complexity and high computational time. This is the marked difference between the existing model in ([11]) and the proposed model. Here the objective function in GA optimization is a minimization function which reduces the MAPE, MSE, RMSE errors subject to the defined constraints used in simulation in MATLAB. MATLAB codes were developed for simulating data analysis and prediction. The determined matrix in classical HMM is an advantage of the proposed model during the initialization process. The first phase is used for determining the parameters which are used in initialization. The size of the chromosome is equal to the dimension of parameters which is to be optimized. For a 'n_{cs_n}' state HMM architecture, the chromosome size will be '$n_{cs_n} \times n_{cs_n}$'. The variable size for GA optimization is $n^2_{cs_n}$. GA is executed till the resulting fitness value converges. Initial Population $= (x_1, x_2, \ldots, x_{n^2_{cs_n}})$. The variables $x_1, x_2, \ldots, x_{n^2_{cs_n}}$ involved in the objective function are called decision variable. All variables $x_1, x_2, \ldots, x_{n^2_{cs_n}}$ are non-negative, $0 \leq x_i \leq 1$. Selection, Crossover and Mutations are the three operators employed by GA. The significance of the proposed model lies in the initial population generation calculated through classical HMM.

An algorithm was implemented in MATLAB for the above mentioned process, a high performance scientific computational tool consisting of the following steps.

Step 1: Initialize the $n^2_{cs_n}$ parameter populations using classical HMM

Step 2: Estimate the $n^2_{cs_n}$ initial population in HMM MATLAB tool

Step 3: Optimize the $n^2_{cs_n}$ parameters using the GA MATLAB tool using Step 4 to Step 9

Step 4: Specify the lower bound and the upper bound for the $n^2_{cs_n}$ parameters to be optimized

Step 5: Set options as shown in Table 1 with gaoptimset in MATLAB during the simulation process to create genetic algorithm options structure

Step 6: Fitness Function is the MAPE, MSE, RMSE error minimization function which is defined and called from the main algorithm

Step 7: Constraint Equations are specified for the algorithm developed. Since the parameters are probability values, it takes values within the range 0 to 1

Step 8: Syntax for minimizing the function : [x, fval, exitflag, output] = ga (Fitness Function, number of variables, lower bound, upper bound, Constraint Function, Options) where x is the best point that GA located during its iterations, fval is the fitness function evaluated at x, exitflag is the integer giving the reason why GA stopped iterating, output is the structure containing the output from each generation giving information about algorithm performance

Step 9: Simulation of GA is terminated when the fitness value remains the same for successive generations

The Activity diagram of the GA-based HMM Algorithm is given below in Fig. 2

The proposed model is a single objective GA optimization finding the best optimal solution for HMM parameters in the optimization problem considered.

3.2. Tuning of FTS with GA Optimized HMM

The state of the one day difference in close value is probabilistically related to the state of the stock volume data. The relationship between the two variates namely, close price and volume are linked by FTS. Now, the height for the second variate (i.e.,) the volume traded data is calculated using the formula $h_v = \frac{u_v - l_v}{n_v}$, where '$u_v$' denotes the upper bound of all traded volume, 'l_v' denotes the lower bound of all traded volume, 'n_v' denotes the number of subintervals of the traded volume. The volume of traded data for the stock indices taken for prediction are divided into 'n_v' subintervals using the height 'h_v'. The first subinterval of the volume traded data is denoted by 'n_{v_1}'. The second subinterval is denoted by 'n_{v_2}'. The n^{th} subinterval is denoted by 'n_{v_n}'. For 'n_{v_n}' number of subintervals, we have 'n_{vs_n}' number of corresponding states. Each State has a midpoint denoted by a vector $\{m_{vs_1}, m_{vs_2}, \ldots, m_{vs_n}\}$. For, the multivariate attributes of the test dataset namely close price and stock volume, there will be 'n_{clhs_n}' and 'n_{vlhs_n}' linguistic hidden states for each stock considered. Consider the observation vector $\{v_1, v_2, \ldots, v_n\}$ of the volume dataset. Check if 'v_1' belongs to any one of the volume traded subintervals defined (i.e.,) if $v_1 \in \{n_{v_1}(or)n_{v_2}(or)\ldots(or)n_{v_n}\}$ then the corresponding subinterval will have the maximum membership degree. The position of the corresponding subinterval where maximum membership degree occurs is determined for further computations. Furthermore, at this stage, the exact membership value at each and every subinterval can also be found out. There are various membership functions available for finding the exact values at all positions of the subintervals. For simplicity and generalization, the triangular membership function is employed in the proposed prediction model. The fuzzy sets are fuzzified by the fuzzifying rules. The Rules for fuzzification using triangular membership function are given below.

Rule 1: If the historical stock volume data v_t at time t is less than m_{vs_1},

Table 1: List of Options set with gaoptimset in MATLAB

Option	Description	Values
CrossoverFcn	Handle to the function that the algorithm uses to create crossover children	@crossoverintermediate
CrossoverFraction	The fraction of the population at the next generation, not including elite children, that is created by the crossover function	Positive scalar |{0.8}
FitnessLimit	Scalar. The algorithm halts when the fitness function attains this FitnessLimit	Scalar |{-Inf}
FitnessScalingFcn	Handle to the function that scales the values of the fitness fuction	@fitscalingrank
Generations	Positive integer specifying the maximum number of iterations before the algorithm halts	Positive integer |{100}
InitialPopulation	Initial population used to seed the genetic algorithm ; can be partial	Matrix from IEPM |{[]}
MigrationFraction	Scalar between 0 and 1 specifying the fraction of individuals in each subpopulation that migrates to a different subpopulation	Scalar |{0.2}
MutationFcn	Handle to the function that produces mutation children	@mutationadaptfeasible
OutputFcns	Functions that ga calls at each iteration	Function handle or cell array of function handles |{[]}
PlotFcns	Array of handles to functions that plot data computed by the algorithm	@gaplotbestf | @gaplotbestindiv | @gaplotdistance | @gaplotexpectation | @gaplotrange | @gaplotselection | @gaplotscores | @gaplotstopping |
PopulationType	String describing the data type of the population	{'doubleVector '}
SelectionFcn	Handle to the function that selects parents of crossover and mutation children	@selectionstochunif
StallGenLimit	The algorithm stops if the weighted average relative change in the best fitness function value over StallGenLimit generations is less than or equal to TolFun	Positive integer |{50}
TolFun	The algorithm stops if the weighted average relative change in the best fitness function value over StallGenLimit generations is less than or equal to TolFun	Positive scalar |{1e -6}

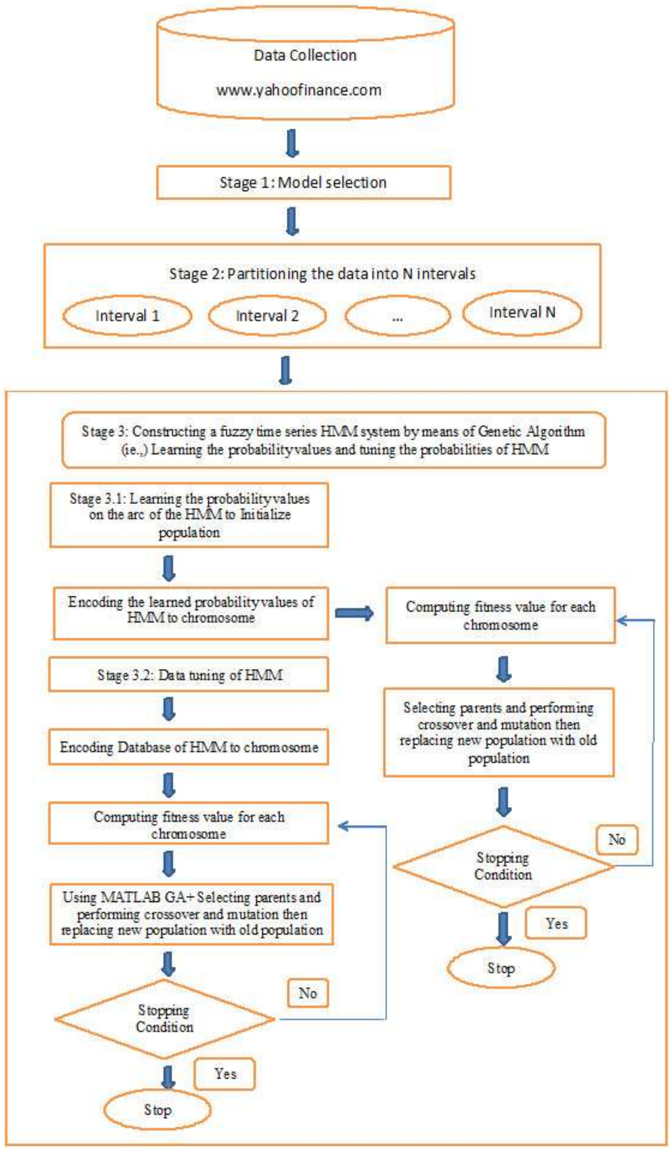

Figure 2: Activity diagram of the GA-based HMM Algorithm.

where m_{vs_1} is the midpoint of the first partitioned subinterval n_{v_1}, then the membership degree is 1 of n_{vs_1}, otherwise zero.

Rule 2: If the historical stock volume data v_t at time t is greater than m_{vs_n}, where m_{vs_n} is the midpoint of the last partitioned subinterval n_{v_n}, then the membership degree is 1 of n_{v_n}, otherwise zero.

Rule 3: If the historical stock volume data v_t at time t is within the range of m_{vs_i} and $m_{vs_{i+1}}$, $i = 1, 2, \ldots, n-1$, then the membership degree is $\frac{m_{vs_{i+1}} - v_t}{m_{vs_{i+1}} - m_{vs_i}}$ for n_{vs_i}, and $\frac{v_t - m_{vs_i}}{m_{vs_{i+1}} - m_{vs_i}}$ for $n_{vs_{i+1}}$, where m_{vs_i} and $m_{vs_{i+1}}$ are the midpoints of the partitioned subinterval n_{v_i} and $n_{v_{i+1}}$, respectively.

The corresponding column in the Optimized Emission probability matrix (OEPM) is chosen on the basis of this position, transposed and multiplied with the initial steady state probability ($I\pi$). The resultant matrix is a row vector whose elements are the states of the close price value. The State of the close price which has the maximum probability value is found out. Then defuzzification is done by adding the corresponding midpoint of the state which has the maximum membership value. Now, the predicted close value is calculated using the formula, $Predicted_{cp}$ = the midpoint of the corresponding state of the close price + the previous day close price data.

4. Experimental Results

4.1. Data Description and Setup

The complete set of data for the proposed study has been taken from http://www.finance.yahoo.comwww.finance.yahoo.com. The daily close price and stock volume data for various NYSE stocks, namely, IBM Corporation, Apple Computer Inc., Dell Inc. for a period from 10^{th} February 2003 to 10^{th} September 2004 are taken as the training data period and the period from 13^{th} September 2004 to 21^{st} January 2005 is taken as the testing data period. In this new model FTS and HMM are executed simultaneously, for daily close price and volume of the stock traded on a particular day. IBM stock price indices are taken as the sample dataset for describing the proposed method. The One day difference in close price 'O_{diff_c}' for the entire dataset 'N_c' are partitioned into 'n_{c_n}' equal subintervals using the height h_c. For 'n_{c_n}' number of subintervals equally partitioned, we have 'm_{cs_n}' equal number of midpoints. Choosing 'n_{c_n}' depends on the expert's opinion. Since the model is a fuzzy time series model, it gets easily over fitted for more number of intervals. It is better to choose a small number of intervals for stronger generalizability. Therefore, 'n_{c_n}' is chosen as six for convenience for the above mentioned reason. The subinterval partition vector is given by $\{n_{c_1}, n_{c_2}, n_{c_3}, n_{c_4}, n_{c_5}, n_{c_6}\}$. The state vectors of the corresponding subinterval partition are $\{n_{cs_1}, n_{cs_2}, n_{cs_3}, n_{cs_4}, n_{cs_5}, n_{cs_6}\}$. The state vectors of the first

Table 2: Fuzzy linguistic states, observing symbols, midpoints showing equal subinterval partition for one day difference in close value (O_{diff_c})

Equal subinterval partition for O_{diff_c}		Fuzzy linguistic states		Fuzzy linguistic observing symbols		Midpoint of the corresponding subinterval	
n_c level	Corresponding subinterval	n_{cs} level	Corresponding fuzzy linguistic description	m_{osm} level	Corresponding fuzzy linguistic description	m_{cs} level	Corresponding midpoint value
1	[-1.8 , -0.925]	1	Very low	1	Large Decrease	1	-1.3625
2	(-0.925 , -0.05]	2	Low	2	Decrease	2	-0.4875
3	(-0.05 , 0.825]	3	Moderate low	3	Moderate Decrease	3	0.3875
4	(0.825 , 1.7]	4	Moderate high	4	Moderate Increase	4	1.2625
5	(1.7 , 2.575]	5	High	5	Increase	5	2.1375
6	(2.575, 3.45]	6	Very high	6	Large Increase	6	3.0125

variant are given fuzzy linguistic descriptions. The descriptions are given by the experts according to the number of states chosen. The fuzzy linguistic observing symbols are denoted as $\{m_{osm_1}, m_{osm_2}, m_{osm_3}, m_{osm_4}, m_{osm_5}, m_{osm_6}\}$. The descriptions made for the symbols are shown in Table 2. The midpoint vector of the subinterval partition is $\{m_{cs_1}, m_{cs_2}, m_{cs_3}, m_{cs_4}, m_{cs_5}, m_{cs_6}\}$.

HMM is used as a learning tool for analysis and estimation of the parameters. The initial values of the parameters of the HMM are obtained from Table 3, using the fuzzy states and observing symbols which are defined linguistically enabling the model to learn the probability values on the arc. ITPM, IEPM and $I\pi$ of classical HMM parameter matrix are given below in Table 4. Table 5 given below shows the Estimated Transition Probability Matrix (ETPM), Estimated Emission Probability Matrix (EEPM) of IBM stock index using HMM in MATLAB. Table 5 infer that in ETPM, the maximum probability occur when the transition was made from n_{cs} level 1 to 2 in the first row. Similarly the other levels of maximum transitions can be noticed from Table 5. The Optimized Emission Probability Matrix (OEPM) for IBM, APPLE, DELL stock indices using GA for estimated HMM variables in MATLAB are given below in Table 6, which clearly shows the probability level of maximum transition from one state to another state. These are the 36 OEPM HMM variables optimized by GA in MATLAB for NYSE stocks.

Fig. 3- 5 gives the results of IBM, APPLE, DELL HMM variable optimization indicating eight criteria decribed in Table 1, namely, current best individual, average distance between individuals, best, worst and mean scores, best value and mean value, selection function, fitness of each individual, score histogram and stopping criteria obtained through GA in MATLAB using the Plot Function. The plot function plots all the criteria mentioned at every generation till the percentage of stopping criteria is satisfied. The crossover and migration fraction was taken by default positive scalar with 0.8 and 0.2 respec-

tively as mentioned in Table 1. The maximum number of iterations set was 100, since for each and every dataset the iterations converged at almost 52 iterations. The algorithm stops if the weighted average relative change in the best fitness function value over stall limit generations is less than or equal to TolFun. For the proposed fitness function these default settings yield the best optimized result. In Fig. 3a, the current best individual among all the 36 variables is the 32^{nd} variable which has the maximum probability value 0.9989. The results in the Fig. 3f, shows the best MAPE of 0.7629 obtained from simulations for IBM stocks. Fig. 3g shows that the 36 variables take value between 0 and 1.

Now, the entire dataset of the second variate N_v are partitioned into n_{v_n} equal subintervals using the height 'h_v'. For 'n_{v_n}' number of subintervals equally partitioned, we have 'm_{vs_n}' equal number of midpoints. n_{v_n} is chosen as six for convenience. The subinterval partition vector is given by

$$\{n_{v_1}, n_{v_2}, n_{v_3}, n_{v_4}, n_{v_5}, n_{v_6}\}.$$

Let

$$\{m_{osm_1}, m_{osm_2}, m_{osm_3}, m_{osm_4}, m_{osm_5}, m_{osm_6}\}$$

be the six fuzzy linguistic observing symbols for the second factor considered. The descriptions made for the symbols are shown in Table 7. The midpoint vector of the subinterval partition is $\{m_{vs1}, m_{vs2}, m_{vs3}, m_{vs4}, m_{vs5}, m_{vs6}\}$.

Table 8 shows the result for the level of triangular membership degree obtained using fuzzification rules as described in Section III(B) for the stock volume data. For instance, the stock volume data 4801400 for IBM stock has the maximum membership degree at m_{osm_2} from Table 8. The fourth column of the OEPM matrix formed in Table 6 is transposed and multiplied with the transpose of the steady state probability $I\pi$ from Table 4. The forecast result is given by $\begin{pmatrix}0 & 0.056 & 0 & 0 & 0 & 0\end{pmatrix}$. The row vector obtained above infer that the maximum steady state probability occurs for the second element in the long run. Finally, defuzzification is done for the forecasting output, considering the corresponding midpoint of the state. The maximum probability occurs at the fuzzy linguistic state n_{cs_2}. Midpoint of the corresponding subinterval is -1.18. Forecasting close price for the next day = close price of the previous day-1.18 = 76.73. Forecasting error = Actual value − Forecasting value = 0.66.

4.2. Overall Performance Evaluation Results

Results of the performance improvement and forecast accuracy details of the proposed model are given in Table 9, below. The forecasting accuracy is

Table 3: The IBM stock index sample test dataset showing the close price, O_{diff_c}, level of the fuzzy linguistic state and observing symbol for a period from 13^{th} September 2004 to 21^{st} January 2005

Close price in dollars	O_{diff_c}	Fuzzy linguistic state 'n_{cs} level' for O_{diff_c}	Fuzzy linguistic Observing symbol 'm_{osm} level'	Close price in dollars	O_{diff_c}	Fuzzy linguistic state 'n_{cs} level' for O_{diff_c}	Fuzzy linguistic Observing symbol 'm_{osm} level'
86.49				94.89	-1.03	1	1
86.72	0.23	3	2	95.46	0.57	3	2
86.37	-0.35	2	3	95.1	-0.36	2	3
86.12	-0.25	2	3	94.45	-0.65	2	3
85.74	-0.38	2	3	95.11	0.66	3	2
85.7	-0.04	3	2	95.28	0.17	3	2
85.72	0.02	3	2	95.46	0.18	3	2
84.31	-1.41	1	1	94.72	-0.74	2	3
83.88	-0.43	2	3	95.5	0.78	3	2
84.43	0.55	3	2	94.24	-1.26	1	1
84.16	-0.27	2	3	95.88	1.64	4	5
84.48	0.32	3	2	95.76	-0.12	2	3
84.98	0.5	3	2	97.08	1.32	4	5
85.74	0.76	3	2	97.67	0.59	3	2
86.72	0.98	4	5	96.1	-1.57	1	1
87.16	0.44	3	2	96.65	0.55	3	2
87.32	0.16	3	2	97.51	0.86	4	5
88.04	0.72	3	2	96.67	-0.84	2	3
87.42	-0.62	2	3	96.45	-0.22	2	3
86.71	-0.71	2	3	97.31	0.86	4	5
86.63	-0.08	2	3	97.33	0.02	3	2
86	-0.63	2	3	97.45	0.12	3	2
84.98	-1.02	1	1	96.2	-1.25	1	1
84.78	-0.2	2	3	96.55	0.35	3	2
84.85	0.07	3	2	97.02	0.47	3	2
85.92	1.07	4	5	97.61	0.59	3	2
89.37	3.45	6	6	97.72	0.11	3	2
88.82	-0.55	2	3	97.5	-0.22	2	3
88.1	-0.72	2	3	98.3	0.8	3	2
87.39	-0.71	2	3	98.18	-0.12	2	3
88.43	1.04	4	5	98.3	0.12	3	2
89	0.57	3	2	98.58	0.28	3	2
90	1	4	5	97.75	-0.83	2	3
89.5	-0.5	2	3	96.7	-1.05	1	1
89.75	0.25	3	2	96.5	-0.2	2	3
90.11	0.36	3	2	96.2	-0.3	2	3
90.47	0.36	3	2	95.78	-0.42	2	3
91.2	0.73	3	2	95.68	-0.1	2	3
92.38	1.18	4	5	95	-0.68	2	3
93.28	0.9	4	5	95.21	0.21	3	2
93.37	0.09	3	2	94.45	-0.76	2	3
93.37	0	3	2	94.1	-0.35	2	3
93.61	0.24	3	2	94.9	0.8	3	2
94.79	1.18	4	5	93.1	-1.8	1	1
95.32	0.53	3	2	93	-0.1	2	3
95.92	0.6	3	2	92.38	-0.62	2	3

measured in terms of MSE, RMSE, MAPE. The following is the procedure to calculate the MSE, RMSE and MAPE.

$$MSE = \frac{\sum_{i=1}^{n} (Forecastingvalue_i - Actualvalue_i)^2}{n},$$

Table 4: The Initial Transition Probability Matrix (ITPM), Initial Emission Probability Matrix (IEPM), Initial Steady State Probability ($I\pi$) of a sample IBM stock index using classical HMM

(a)

ITPM	n_{cs_1}	n_{cs_2}	n_{cs_3}	n_{cs_4}	n_{cs_5}	n_{cs_6}
n_{cs_1}	0	0.5	0.375	0.125	0	0
n_{cs_2}	0.0645	0.4839	0.3548	0.0968	0	0
n_{cs_3}	0.1538	0.2308	0.4615	0.1538	0	0
n_{cs_4}	0	0.2727	0.5455	0.0909	0	0.0909
n_{cs_5}	0	0	0	0	0	0
n_{cs_6}	0	1	0	0	0	0

(b)

IEPM	m_{osm_1}	m_{osm_2}	m_{osm_3}	m_{osm_4}	m_{osm_5}	m_{osm_6}
n_{cs_1}	0	0.5	0.375	0.125	0	0
n_{cs_2}	0	0.4428	0.5572	0	0	0
n_{cs_3}	0.5961	0	0.0002	0.4036	0	0
n_{cs_4}	0	0.3306	0.4819	0.1375	0	0.0500
n_{cs_5}	0	0	0	0	0	0
n_{cs_6}	0	1	0	0	0	0

(c)

$I\pi =$	(0.089	0.347	0.433	0.122	0	0.011)

$$RMSE = \sqrt{MSE},$$

$$MAPE = \frac{100}{n} \sum_{i=1}^{n} \frac{(Forecastingvalue_i - Actualvalue_i)}{Actualvalue_i},$$

where 'n' denotes the total number of data sequence; $Actualvalue_i$ indicates the actual stock price on day 'i'; $Forecastingvalue_i$ indicates the forecast stock price on day 'i'.

For the proposed hybrid model, the different errors MSE, RMSE, MAPE calculated for IBM, APPLE, DELL stock indices are given below in Table 9. The results show that the function count for IBM, APPLE, DELL MATLAB

Table 5: The Estimated Transition Probability Matrix (ETPM), Estimated Emission Probability Matrix (EEPM) of IBM stock index using HMM in MATLAB

(a)

ETPM	n_{cs_1}	n_{cs_2}	n_{cs_3}	n_{cs_4}	n_{cs_5}	n_{cs_6}
n_{cs_1}	0	1	0	0	0	0
n_{cs_2}	0.0001	0.8272	0	0.1727	0	0
n_{cs_3}	0.7574	0	0.0502	0.1924	0	0
n_{cs_4}	0	0	1	0	0	0
n_{cs_5}	0	0	0	0	0	0
n_{cs_6}	0	1	0	0	0	0

(b)

EEPM	m_{osm_1}	m_{osm_2}	m_{osm_3}	m_{osm_4}	m_{osm_5}	m_{osm_6}
n_{cs_1}	0	0.5789	0.3705	0.0506	0	0
n_{cs_2}	0	0.4428	0.5572	0	0	0
n_{cs_3}	0.5961	0	0.0002	0.4036	0	0
n_{cs_4}	0	0.3306	0.4819	0.1375	0	0.0500
n_{cs_5}	0	0	0	0	0	0
n_{cs_6}	0	1	0	0	0	0

simulation gave 67840, 54680, 60460 respectively with 52 generations each plotting the best MAPE for NYSE stocks in Fig. 3-5.

4.3. The Performance Improvement and Directional Accuracy of the Proposed Model

The performance of the model proposed is measured in terms of MSE, RMSE, and MAPE. The directional accuracy of forecasting is more important than MSE, RMSE, and MAPE when generating the strategic trading rules for investment. Table 11 given below presents the experiment results for each of the different other four stocks of NSE considered. Fig. 6 gives the Graphical Representation for INFOSYS, TCS, HCLTECH, WIPRO stock data. It is interesting to note the improved performance of the FTS, GA optimized HMM model. To validate the proposed model, a comparative result in terms of MSE,

Table 6: The Optimized Emission Probability Matrix (OEPM) of IBM, APPLE, DELL stock indices using GA for HMM variables in MATLAB

(a)

$$\text{OEPM for IBM} = \begin{pmatrix} 0 & 0.4999 & 0.3750 & 0.1249 & 0.0002 & 0 \\ 0.0331 & 0.4786 & 0.3445 & 0.0890 & 0 & 0.0547 \\ 0.1539 & 0.2306 & 0.4613 & 0.1540 & 0.0001 & 0.0001 \\ 0 & 0.2721 & 0.5456 & 0.0916 & 0.0001 & 0.0906 \\ 0.2497 & 0.0036 & 0 & 0.0160 & 0.0001 & 0.7306 \\ 0 & 0.9989 & 0 & 0.0002 & 0.0009 & 0 \end{pmatrix}$$

(b)

$$\text{OEPM for APPLE} = \begin{pmatrix} 0.0001 & 0 & 0.5000 & 0.4999 & 0 & 0 \\ 0.0001 & 0.2501 & 0.5001 & 0.1249 & 0.1249 & 0 \\ 0.0167 & 0.0500 & 0.7167 & 0.1499 & 0.0334 & 0.0333 \\ 0 & 0.1540 & 0.5380 & 0.1540 & 0.0770 & 0.0770 \\ 0.2501 & 0.2498 & 0.5001 & 0.0001 & 0 & 0 \\ 0.0002 & 0.0001 & 0.9996 & 0 & 0 & 0 \end{pmatrix}$$

(c)

$$\text{OEPM for DELL} = \begin{pmatrix} 0 & 0.0005 & 0.9994 & 0 & 0 & 0.0001 \\ 0.3744 & 0.2927 & 0 & 0.3328 & 0 & 0 \\ 0.0001 & 0.6467 & 0.3415 & 0.0001 & 0.0009 & 0.0107 \\ 0.1288 & 0.0791 & 0.0154 & 0.2501 & 0.5263 & 0.0003 \\ 0.7646 & 0 & 0.0001 & 0.0477 & 0.1875 & 0.0001 \\ 0.2494 & 0.0035 & 0.6652 & 0.0123 & 0.0664 & 0.0032 \end{pmatrix}$$

RMSE and MAPE for all the four stocks of NSE is calculated for a period of 8 months to know the directional accuracy of the proposed model. The result shows the prediction quality with a best Average MAPE of 1.3% for INFOSYS, 1.71% for TCS, 1.16% for HCLTECH and 1.09% for WIPRO. These stock indices also on an average given a minimized error and an accurate prediction, analysing the directional accuracy average during the eight months period taken for study. Table 11 investigates on the performance on the average MAPE for eight months. The MAPE error was high showing 2.06% during the month of May for INFOSYS stock and the same was much reduced showing 0.44% during the month of february. But on an average the MAPE error was 1.3% for INFOSYS. A similar analysis was done for other stocks such as TCS, HCLTECH

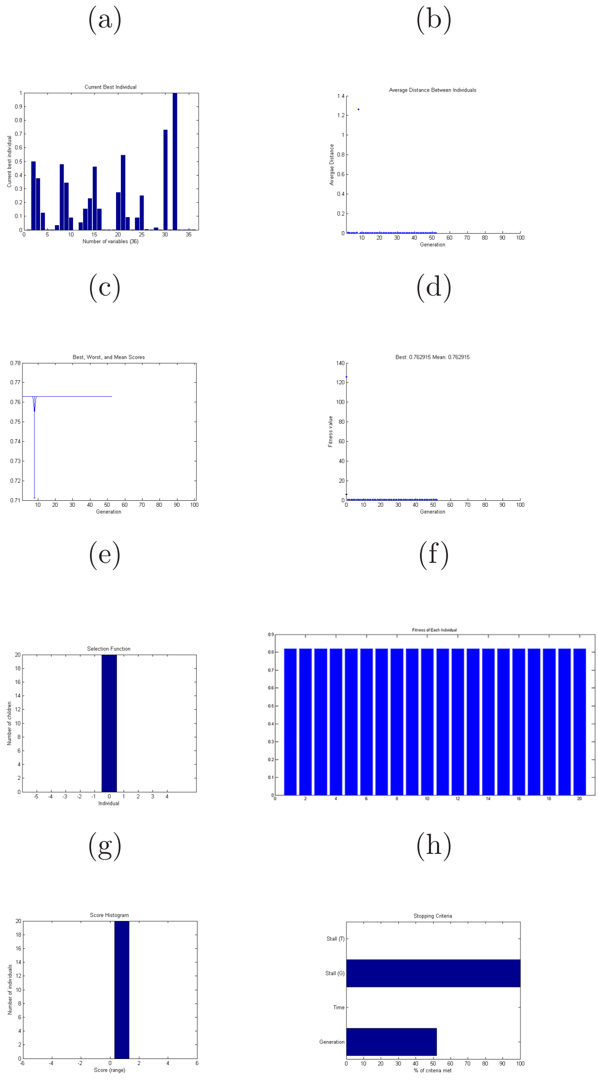

Figure 3: Results of IBM HMM variable optimization

and WIPRO to figure out the maximum, minimum and average error that had occured during prediction. The stocks of NSE were taken only for analysing the deviations in the error performance analysis over a certain period during the year. As a result, the best possible model is built for each of the stock indices

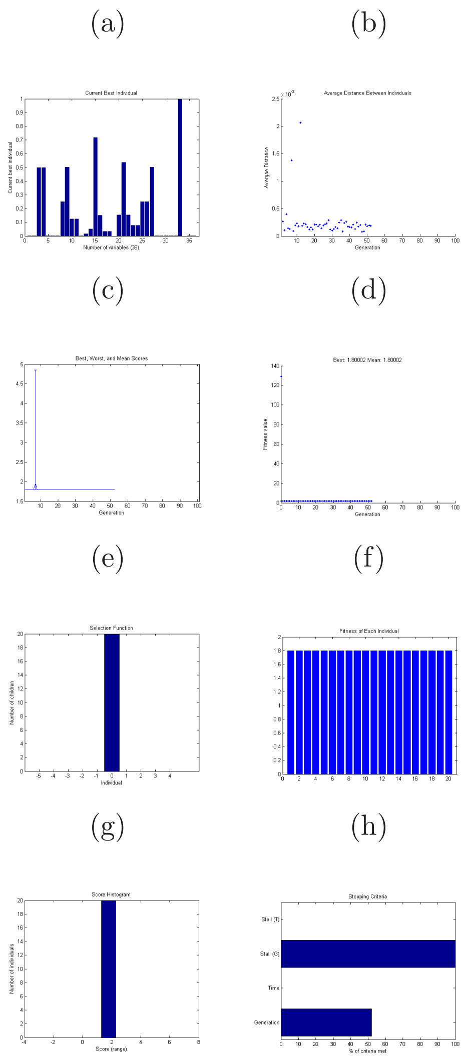

Figure 4: Results of APPLE HMM variable optimization

in this experiment.

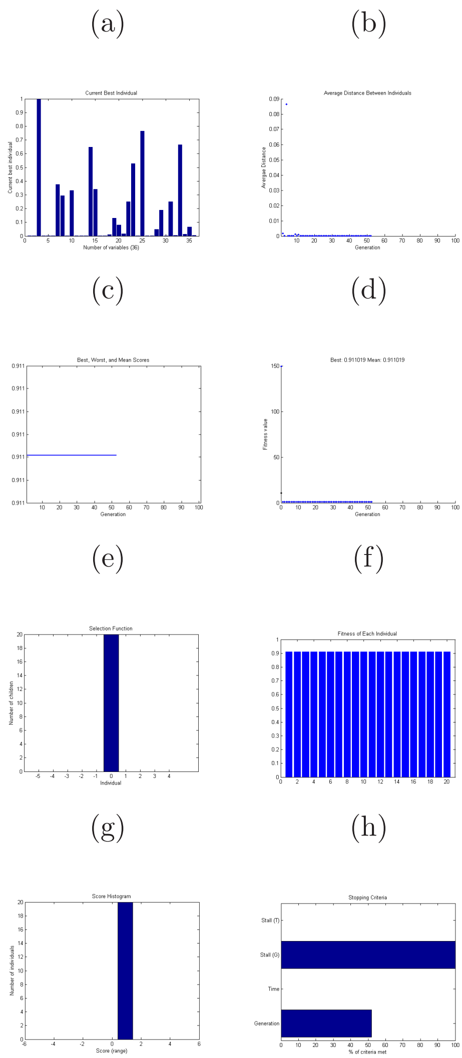

Figure 5: Results of DELL HMM variable optimization

4.4. Forecasting Accuracy Detail and Discussions

The dataset was divided into training set, validation set and testing set for a better study and analysis. The purpose of the simulation experiment is mainly to minimize the prediction error. Table 1 demonstrates the options that are

Table 7: Fuzzy linguistic set with equal interval partition for stock volume data of IBM sample dataset

Equal subinterval partition for stock volume dataset		Fuzzy linguistic observing symbols		Midpoint of the corresponding subinterval	
n_v level	Corresponding subinterval	m_{osm} level	Corresponding fuzzy linguistic description	m_{vs} level	Corresponding midpoint value
1	[2204300 , 4118950]	1	Large Decrease	1	3161625
2	(4118950, 6033600]	2	Decrease	2	5076275
3	(6033600 , 7948250]	3	Moderate Decrease	3	6990925
4	(7948250 , 9862900]	4	Moderate Increase	4	8905575
5	(9862900 , 11777550]	5	Increase	5	10820225
6	(11777550 , 13692200]	6	Large Increase	6	12734875

used to set gaoptimset parameters in MATLAB, so that the parameters are better optimized. Table 3 gives the level of the hidden sequence of fuzzy linguistic states and the corresponding levels of observing symbols. These sequence level helps in manipulating the probability values of ITPM, IEPM, $I\pi$ given in Table 4. It can be observed that in ITPM, transition from state 1 to state 2 has the maximum probability of 0.5, state 2 to state 2 has a maximum transition probability of 0.4839, state 3 to state 3 has a maximum transition probability of 0.4615, state 4 to state 3 has a maximum transition probability of 0.5455, there is no transition from state 5 to any of the other states and state 6 to state 2 has a maximum transition probability of 1. Similarly the transitions can be seen for IEPM. The $I\pi$ investigates the long run performance of the states. It is found that the state 3 has the maximum probability value of 0.433. Analysis indicate that, State 3 will be key state during the prediction process. In Table 5, ETPM, EEPM was found to have the same transition states with maximum probability as Table 4. Table 6 gives the iterated final optimized state transitions maximum probability, which is considered for prediction. This OEPM obtained through GA is considered as the best matrix which is taken for further analysis. Fig. 3- 5 shows the performance of the model for Table 1 options and values. Fig. 6 demonstrates the directional accuracy for NSE stocks indicating the average MAPE error. Fig. 7 gives the comparison of the predicted value and the actual value of the sample IBM stock data when the test data is used as input to the proposed model. The experimental results show that the model can greatly reduce the MSE, RMSE, MAPE and improve forecasting accuracy. The efficacy of the model proposed is evaluated by comparing the

Table 8: The dataset of IBM stock volume for a period from 13th September 2004 to 21st January 2005

Stock Volume dataset	triangular membership degree, 'tr_{md}' level						Corresponding m_{osm} level where maximum degree is attained	Stock Volume dataset	triangular membership degree, 'tr_{md}' level						Corresponding m_{osm} level where maximum degree is attained
	1	2	3	4	5	6			1	2	3	4	5	6	
4801400	0.14	0.86	0	0	0	0	2	5684100	0	0.68	0.32	0	0	0	2
3953500	0.59	0.41	0	0	0	0	1	6353200	0	0.33	0.67	0	0	0	3
4631200	0.23	0.77	0	0	0	0	2	4655900	0.22	0.78	0	0	0	0	2
3623000	0.76	0.24	0	0	0	0	1	5679100	0	0.69	0.31	0	0	0	2
6198700	0	0.41	0.59	0	0	0	3	5814100	0	0.61	0.39	0	0	0	2
4380400	0.36	0.64	0	0	0	0	2	5529900	0	0.76	0.24	0	0	0	2
4049700	0.54	0.46	0	0	0	0	1	3750600	0.69	0.31	0	0	0	0	1
5037100	0.02	0.98	0	0	0	0	2	2204300	1	0	0	0	0	0	1
4801200	0.14	0.86	0	0	0	0	2	5699800	0	0.67	0.33	0	0	0	2
4899500	0.09	0.91	0	0	0	0	2	5870300	0	0.59	0.41	0	0	0	2
4650300	0.22	0.78	0	0	0	0	2	5664500	0	0.69	0.31	0	0	0	2
3874200	0.63	0.37	0	0	0	0	1	5152300	0	0.96	0.04	0	0	0	2
4204500	0.46	0.54	0	0	0	0	2	7026800	0	0	0.98	0.02	0	0	3
5198000	0	0.94	0.06	0	0	0	2	526300	0	0.90	0.10	0	0	0	2
4538000	0.28	0.72	0	0	0	0	2	6477100	0	0.27	0.73	0	0	0	3
5001400	0.04	0.96	0	0	0	0	2	5310700	0	0.88	0.12	0	0	0	2
5150700	0	0.96	0.04	0	0	0	2	5713700	0	0.67	0.33	0	0	0	2
3984400	0.57	0.43	0	0	0	0	1	4188300	0.46	0.54	0	0	0	0	2
3076900	1	0	0	0	0	0	1	4799500	0.14	0.86	0.41	0	0	0	2
4090000	0.52	0.48	0	0	0	0	1	4493200	0.30	0.70	0.41	0	0	0	2
3016300	1	0	0	0	0	0	1	3914500	0.61	0.39	0	0	0	0	1
4626600	0.23	0.77	0	0	0	0	2	5660100	0	0.70	0.30	0	0	0	2
6651400	0	0.18	0.82	0	0	0	3	8853100	0	0	0.03	0.97	0	0	4
4233700	0.44	0.56	0	0	0	0	2	4769900	0.16	0.84	0	0	0	0	2
5928500	0	0.55	0.45	0	0	0	2	4841800	0.12	0.88	0	0	0	0	2
7182600	0	0	0.90	0.10	0	0	3	4950100	0.07	0.93	0	0	0	0	2
13692200	0	0	0	0	0	0	6	3590600	0.78	0.22	0.41	0	0	0	1
6926800	0	0.03	0.97	0	0	0	3	3262900	0.95	0.05	0	0	0	0	1
6137500	0	0.45	0.55	0	0	0	3	4336400	0.39	0.61	0	0	0	0	2
5988700	0	0.52	0.48	0	0	0	2	3296300	0.93	0.07	0	0	0	0	1
5774500	0	0.64	0.36	0	0	0	2	3812400	0.66	0.34	0	0	0	0	1
7335800	0	0	0.82	0.18	0	0	3	2793200	1	0	0	0	0	0	1
6035100	0	0.50	0.50	0	0	0	3	5295200	0	0.89	0.11	0	0	0	2
4226500	0.44	0.56	0	0	0	0	2	5711000	0	0.67	0.33	0	0	0	2
4518500	0.29	0.71	0	0	0	0	2	5646700	0	0.70	0.30	0	0	0	2
5160600	0	0.96	0.04	0	0	0	2	4561700	0.27	0.73	0	0	0	0	2
5388700	0	0.84	0.16	0	0	0	2	6200700	0	0.41	0.59	0	0	0	3
6553300	0	0.23	0.77	0	0	0	3	4625100	0.24	0.76	0	0	0	0	2
6951600	0	0.02	0.98	0	0	0	3	4746400	0.17	0.83	0	0	0	0	2
6708400	0	0.15	0.85	0	0	0	3	5828600	0	0.61	0.39	0	0	0	2
4907300	0.09	0.91	0	0	0	0	2	5339400	0	0.86	0.14	0	0	0	2
4513100	0.29	0.71	0	0	0	0	2	5520800	0	0.77	0.23	0	0	0	2
6258400	0	0.38	0.62	0	0	0	3	8492100	0	0	0.22	0.78	0	0	4
7453400	0	0	0.76	0.24	0	0	3	7352700	0	0	0.81	0.19	0	0	3
4958400	0.06	0.94	0	0	0	0	2	5708600	0	0.67	0.33	0	0	0	2
4887600	0.10	0.90	0	0	0	0	2	7002600	0	0	0.99	0.01	0	0	3

obtained forecast accuracy with that of the popular statistical forecasting tool. The comparison of the errors with other existing methods are given below in Table 10. Table 10 infers that the accuracy of the forecast value of the proposed hybrid model is as good as that of other existing models in ([11]) for IBM and APPLE with best MAPE of 0.7629% and 1.8000% respectively. The forecast accuracies for DELL are better than HiMMI in ([11]), but still it can be improved compared to other models in ([11]). The Comparison shows the

Table 9: The performance improvement and forecast accuracy in percentage for 36 GA optimized variables in MATLAB of the proposed model

Stock Name	MSE	RMSE	MAPE	funccount	No. of Generations
IBM	0.8430	0.9181	0.7629	67840	52
APPLE	2.5611	1.6003	1.8000	54680	52
DELL	0.2782	0.5274	0.9110	60460	52

Table 10: Comparing the proposed method with existing methods

Stock Name	Mean absolute percentage error (MAPE) in forecast for test dataset				
	ARIMA	HiMMI	ANN-GA-HMM Interpolation	Hassan fusion model with weighted average	The proposed model
IBM	0.9723	1.2186	1.0555	0.8487	0.7629
APPLE	1.8009	2.8373	2.1650	1.9247	1.8000
DELL	0.6604	1.0117	0.8446	0.6992	0.9110

Note: Train dataset collected from 10.2.2003 to 10.9.2004
Test dataset collected from 13.9.2004 to 21.1.2005

forecasting efficiency of the proposed model is better than that of other models. Parameters are also optimized in the process which when not optimized cause considerable fall in the prediction performance, rendering them important to the proposed prediction model.

5. Conclusion

The proposed model focuses on business performance analysis based on two-factor fuzzy time series, Hidden Markov model and single objective GA optimization. The model uses GA optimization for HMM parameter optimization in stock price prediction to improve the forecasting accuracy rate of NYSE stocks namely IBM, APPLE, DELL INC. and NSE stocks namely INFOSYS, TCS, HCLTECH, WIPRO. The proposed fusion model has been implemented

Table 11: The directional accuracy in terms of different forecasting error for various stocks of NSE for the months of year 2014

Stock symbol	Month	MSE	RMSE	MAPE
INFOSYS	January	3846.56	62.02	1.77
	February	570.77	23.89	0.44
	March	8725.60	93.41	1.48
	April	1578.72	39.73	0.99
	May	8168.57	90.38	2.06
	June	2789.52	52.82	1.27
	July	3109.75	55.77	1.32
	August	2992.69	54.71	1.09
	Average	3972.77	59.09	1.30
TCS	January	12018.16	109.63	3.64
	February	4412.35	66.43	2.61
	March	2532.06	50.32	1.70
	April	2901.66	53.87	1.96
	May	186.64	13.66	0.50
	June	1324.38	36.40	1.23
	July	2434.7	49.34	1.34
	August	530.31	23.03	0.67
	Average	3292.43	50.34	1.71
HCLTECH	January	485.97	22.04	1.15
	February	53.88	7.34	0.40
	March	389.81	19.74	1.13
	April	531.85	23.06	1.29
	May	806.90	28.41	1.25
	June	624.10	24.98	1.12
	July	1199.11	34.63	1.91
	August	399.46	19.99	1.04
	Average	561.39	22.52	1.16
WIPRO	January	82.87	9.10	1.25
	February	41.89	6.47	0.79
	March	65.88	8.12	1.12
	April	98.69	9.93	1.29
	May	48.24	6.95	1.07
	June	38.48	6.20	0.88
	July	168.75	12.99	1.82
	August	17.51	4.18	0.52
	Average	70.29	7.99	1.09

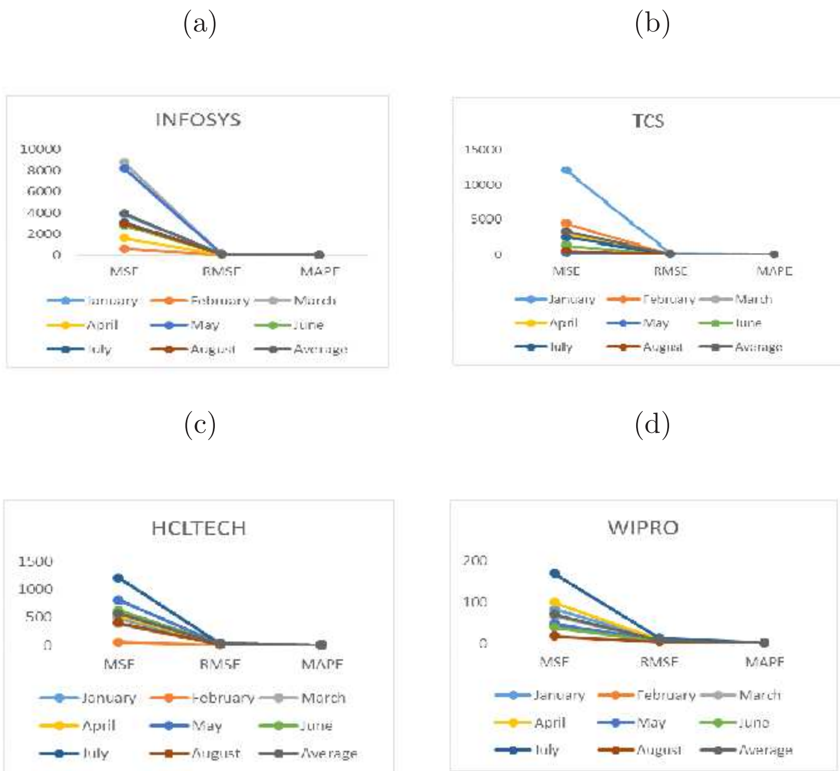

Figure 6: Graphical Representation for INFOSYS, TCS, HCLTECH, WIPRO stock data

choosing the number of states different from the number of attributes in the HMM architecture. Inaccurate forecasts result in poor stock market trading decisions. The stock market deals with huge money. So, the loss is very high for every wrong decision made. It is worth the effort to make sure that the forecasts are as accurate as possible since it involves money and risks. As a result, the performance of the proposed model is better than that of the other basic and fusion models for both training phase and the testing phase during simulation process. Therefore in this direction, the proposed model predicts the trend of the future direction also. In the future, the plan is to develop an online stock price prediction system using fuzzy time series with multi objective GA for HMM parameter optimization. An improvement in performance can be

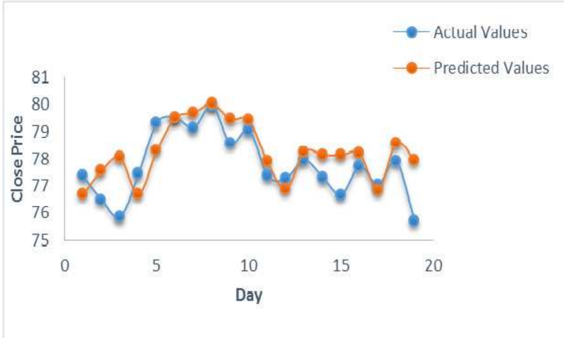

Figure 7: Comparison of the predicted value and the actual value of the IBM sample stock index dataset.

achieved by taking minute-by-minute prediction of online stock value prices.

References

[1] S. M. Chen, Forecasting enrolments based on high-order fuzzy time series, *Cybernetics System: An International Journal*, **33** (2002), 1-16.

[2] S. M. Chen, Y. C. Chang, Multi-variable fuzzy forecasting based on fuzzy clustering and fuzzy rule interpolation techniques, *Information Science*, **180**, No. 24 (2010), 4772-4783.

[3] S. M. Chen, C. D. Chen, TAIEX forecasting based on fuzzy time series and fuzzy variation groups, IEEE Transactions, *Fuzzy Systems*, **19**, No. 1 (2011), 1-12.

[4] M. Y. Chen, B. T. Chen, A hybrid fuzzy time series model based on granular computing for stock price forecasting, *Information Science*, **294**, No. 2 (2015), 227-241.

[5] S. M. Chen, H. P. Chu, T. W. Sheu, TAIEX forecasting using fuzzy time series and automatically generated weights of multiple factors, *IEEE Transactions on System Man Cybernet. Parts A Syst. Hum.*, **42**, No. 6 (2012), 1485-1495.

[6] S. M. Chen, J. R. Hwang, Temperature prediction using fuzzy time series, *IEEE Trans. on Syst., Man. and Cybernetics-Part B: Cybernetics*, **30** (2000), 263-275.

[7] S. M. Chen, P. Y. Kao, TAIEX forecasting based on fuzzy time series, particle swarm optimization techniques and support vector machines, *Information Science*, **247** (2013), 62-71.

[8] Z. Q. Guo, H. Q. Wang, Q. Liu, J. Yang, *A Feature Fusion Based Forecasting Model for Financial Time Series*, PLoS ONE: e101113, **doi:** 10.1371/journal.pone.0101113.

[9] Hassan Rafiul, Baikunth Nath, Stock Market forecasting using hidden Markov model: A new approach, In: *Proceedings 2005 5th International Conference on Intelligent System, Design and Applications*, 0-7695-2286-06/05, IEEE (2005).

[10] Hassan Rafiul, Nath Baikunth, Michael Kirley, HMM based fuzzy model for time series prediction, *IEEE International Conference on Fuzzy Systems* (2006), 2120-2126.

[11] Hassan Rafiul, Baikunth Nath, Michael Kirley, A fusion model of HMM, ANN and GA for stock market forecasting, *Expert Systems with Applications* (2007), 171-180.

[12] R. Hassan, A combination of hidden Markov model and fuzzy model for stock market forecasting, *Neurocomputing*, **72** (2009), 3439-3446.

[13] K. H. Huarng, T. H. K. Yu, Y. W. Hsu, A multivariate heuristic model for fuzzy time-series forecasting, *IEEE Transactions on System, Man Cybernet. B Cybernet*, **37**, No. 4 (2007), 836-846.

[14] Jyoti Badge, Stock Market Forecasting by multivariate higher order fuzzy time series model, *International Journal of Mathematical Finance*, **1**, No. 1 (2013), 1-11.

[15] L. W. Lee, L. H. Wang, S. M. Chen, Temperature prediction and TAIEX forecasting based on high-order fuzzy logical relationships and genetic simulated annealing techniques, *Expert Systems with Applications*, **34** (2008), 328- 336.

[16] Ling-Yuan Hsu, Shi-Jinn Horng, Tzong-Wann Kao, Temperature prediction and TAIFEX forecasting based on fuzzy relationships and MTPSO techniques, *Expert Systems with Applications*, **37** (2010), 2756-2770.

[17] Qisen Cai, Defu Zhang, A novel stock forecasting model based on fuzzy time series and genetic algorithm, *Sciverse Science Direct* (2013).

[18] Qisen Cai, Defu Zhang, Wei Zheng, Stephen, C. H. Leung, A new fuzzy time series forecasting model combined with ant colony optimization and auto-regression, *Knowledge Based Systems*, **74** (2015), 61-68.

[19] L. Rabiner, B. Juang, *Fundamentals of Speech Recognition*, Prentice-Hall, Englewood Cliffs, NJ, 1993.

[20] L. R. Rabiner, A tutorial on HMM and Selected Applications in Speech Recognition, *Proceedings IEEE*, **77**, No. 2 (1993), 267-296.

[21] Rafal Weron, Electricity price forecasting: A review of the state-of-the-art with a look into the future, *International Journal of Forecasting*, **30** (2014), 1030-1081.

[22] Saerom Park, Jaewook Lee, Youngdoo Son, *Predicting Market Impact Costs Using Nonparametric Machine Learning Models*, PLoS ONE: e0150243, **doi:** 10.1371/journal.pone.0150243.

[23] ShashankIyer, R. Nisarg, Kamdar, BaharSoparkar, Stock market prediction using digital signal processing models, *International Journal of Computer Applications* (2015).

[24] Q. Song, B. S. Chissom, Fuzzy time series and its models, *Fuzzy Sets and Systems*, **54** (1993), 269-277.

[25] Q. Song, B. S. Chissom, Forecasting enrolments with fuzzy time series, *Fuzzy Sets and Systems*, **54** (1993), 1-9.

[26] M. Stevenson, E. P. John, Fuzzy time series forecasting using percentage change as the universe of discourse, *World Acadamic Science and Engineering Technology*, **55** (2009), 154-157.

[27] J. Sullivan, W. H. Woodall, A comparison of fuzzy forecasting and Markov modelling, *Fuzzy Sets and Systems*, **64** (1994), 279-293.

[28] R. C. Tsaur, J. C. Yang, H. F. Wang, Fuzzy relation analysis in fuzzy time series model, *Computers and Mathematics with Applications*, **49** (2005), 539-548.

[29] L. Y. Wei, C. H. Cheng, H. H. Wu, A hybrid ANFIS based on n-period moving average model to forecast TAIEX stock, *Applied Soft Computing*, **19** (2014), 86-92.

[30] H. K. Yu, Weighted fuzzy time series models for TAIEX forecasting, *Physica A*, **349** (2005), 609-624.

[31] T. H. K. Yu, K. H. Huarng, A bivariate fuzzy time series model to forecast the TAEIX, *Expert Systems with Applications*, **34**, No. 4 (2010), 2945-2952.

[32] T. H. K. Yu, K. H. Huarng, Corrigendum to A bivariate fuzzy time series model to forecast the TAIEX, *Expert Systems with Applications*, **34**, No. 4 (2010), 2945-2952.

[33] Yun Xue, Zhiwen Liu, Jie Luo, Zhihao Ma, Meizhen Zhang, Xiaohui Hu, Qiuhua Kuang, Stock market trading rules discovery based on biclustering method, *Mathematical Problems in Engineering*, Article ID 849286 (2015).

[34] Zhiqiang Guo, Huaiqing Wang, Jie Yang, David J. Miller, *A Stock Market Forecasting Model Combining Two-Directional Two-Dimensional Principal Component Analysis and Radial Basis Function Neural Network*, PLoS ONE: e0122385, **doi:** 10.1371/journal.pone.0122385.