A Novel Approach of Genetic Algorithm in prediction of Time Series Data

Satyendra Nath Mandal
Kalyani Govt. Eng. College, Kalyani, W.B., India

Arghya Ghosh
Kalyani Govt. Eng. College, Kalyani, W.B., India

Subhojit Roy
Kalyani Govt. Eng. College, Kalyani, W.B., India

J. Pal Choudhury
Kalyani Govt. Eng. College, Kalyani, W.B., India

S. R. Bhadra Chaudhuri
BESU, Shibpur, W.B., India

ABSTRACT
The statistical and soft computing methods have been used to predict time series data from different fields. The different methods have been applied on same or different time series data. One method has been selected among the applied methods based error analysis. In this paper, an algorithm has been proposed to forecast the time series data based on Genetic Algorithm. The error has been calculated between original and predicted data. A comparison has been made among the predicted result by Genetic Algorithm and the other applied methods like fuzzy system, artificial neural network and statistical methods on same data set. It has been proved that the Genetic Algorithm method is given better result compare to other applied methods.

Keywords
Statistical Method, Soft Computing Method, Fuzzy system, Artificial Neural Network, Genetic Algorithm and Time Series Data.

1. INTORDUCTION
Stephen D. Sloan, Raymond W. saw, James J.Sluss [1] have described genetic algorithm to forecast the long term quarterly sales of product in telecommunication technology sector. This has been used widely available economic indicators such as disposable personal income and new housing starts as independent variables. Authors have used individual chromosomes to indicate inclusion and dis-inclusion of specific economic variable as well as operational rules for combining the variables. In their proposed method, several features beyond those of canonical GA were also incorporated, including evolution of individual in distinct ecosystem with a specific level of intermarriage between ecosystems, the capability for a single gene in an individual's chromosome to indicate a subroutine call to the complete chromosome of an individual from a previous generation, and hill-climbing applied to improve the most fit offspring produced by generation.

Sultan H. Aljahdali and Mohammed E. El-Telbany [2] have used genetic algorithm to predict software reliability. Software reliability models are very useful to estimate the probability of the software fail along the time. They have used Genetic Algorithms (GA) as an alternative approach to derive proposed models. GA is a powerful machine learning technique and optimization techniques to estimate the parameters of well-known reliably growth models. They choose GA for this task is its capability of estimating optimal parameters through learning from historical data. They have conducted this experiment to confirm these hypotheses by evaluating the predictive capability of the developed ensemble of models and the results were compared with traditional models.

A Genetic Algorithm for Conformational Analysis of DNA by C. B. Lucasius, M. J. J. Blommers, L. M. C. Buydens, and G. Kateman[3]. It is a development of a genetic algorithm for determining the structure of a sample of DNA based on spectrometric data about the sample. An interesting "cascaded" evaluation technique that greatly enhances the efficiency of their evaluation function has been used. The authors have used the bit strings to encode molecular structures. Their evaluation function have been measured the degree to decode structure conforms to the data that have been collected about the sample. The genetic algorithm evolves a description of molecular structure in agreement with the data collected.

Bahaa Ibraheem Kazem, Ali Ibrahim Mahdi, Ali Talib Oudah [4] have proposed genetic algorithm to optimize the point-to-point trajectory planning for a 3-link robot arm. The objective function of their proposed method has been minimizing the traveling time and space, while not exceeding a maximum pre-defined torque, without collision with any obstacle in the robot workspace.

Application of genetic algorithm in software testing has been introduced by Praveen Ranjan Srivastava and Tai-hoon Kim [5]. They do this by developing variable length genetic algorithm that optimize and select the software path clusters are weighted in accordance with the criticality of the path. Generally software testing is rarely possible because it becomes intractable for even medium sized software. Typically only parts of a program can be tested for this reason they are focusing only those parts that are most critical so that these paths can be tested.

Reza Entezari-Maleki, Ali Movaghar [6] have used genetic algorithm to increase the throughput of the computational grids. High throughput computing (HTC) is of great importance in grid computing environments. High throughput computing has been aimed at minimizing the total makes span of all of the tasks submitted to the grid environment in long execution of the system. They have proposed a new scheduling algorithm to assign the tasks to the grid resources with goal of minimizing the total make span of the environment. They used genetic algorithm to find the most suitable match between the tasks and grid resources.

Long term energy consumption forecasting using genetic algorithm was introduced by Korhan Karabulut, Ahmet Alkan and Ahmet S. Yilmaz [7]. The most important part of electric utility resource planning is forecasting of the future load demand in the service area. This is achieve by constructing models on the relative information, such as climate and previous load demand data. They have used genetic algorithm to forecast long term electrical power consum ption in the area covered by utility situated in the southeast turkey.

Kathryn A. Dowsland[8] have applied the genetic algorithm in the solution of both classical and practical operational research problems and identifies some of the reasons why they have been slow to find widespread appeal. Then he compared this with other metaheuristic techniques such as simulated annealing and tabu search.

Rahul Kala, Harsh Vazirani, Anupam Shukla and Ritu Tiwari[9] have introduced offline handwriting recognition technique using genetic algorithm. There have enumerable styles in which a character may be written this style can be self-combined to generate more style. In order to prove this they have made a pool of images of characters. They have converted them to graph. Then graph of every character was intermixed to generate style intermediate between the styles of parent character. Character recognition involved the matching of graph generated from the unknown character image with the graphs generated by mixing.

M. K. Deshmukh, C. Balakrishna Moorthy [10] have introduced genetic algorithm to neural network model, namely, feed forward neural network for estimation of wind energy potential at a site. Their proposed model has been used to predict power output of wind conversion system. In this model, real time values of wind speed and variable are taken as input and electric power generated by WECS is computed as output of the model. In this model, neural network with genetic algorithm has been proposed for improvement of the output. The results obtained using this proposed model is compared with those obtained using back propagation algorithm. It is reported that their proposed modified model leads to improved accuracy in prediction of wind energy.

Kristin Bennett, Michael C. Ferris and Yannis E. Ioannidis [11] have described a database query optimization problem and the adaptation of genetic algorithm to this problem. They have presented a method for encoding arbitrary binary trees as chromosomes and describe several crossover operations for such chromosomes. Preliminary computational comparisons with the current best known method for query optimization have been indicated that their proposed method is more promising approach than others. In particular, the output quality and the time needed to produce such solutions are comparable to and in general better than the current method.

Sandeep Singh Rawat, Lakshmi Rajamani [12] have used genetic algorithm in the quest for an optimal class timetable generator. In their paper, they have explained an example usage of genetic algorithm for finding optimal solutions to the problem of class timetable. Their proposed method does not take care of other constraints like unavailability of lecturers, small size of rooms and time required by the lecturer to move from one class to other class, which is to be considered in the future up gradation. This automated class timetable is used at the dept. of computer science & Engineering, Guru Nanak Institute of Technology, Hyderabad

In this paper, an effort has been made to forecast the time series data based on Genetic Algorithm. At first, the search space has been created by random number. Four numbers have been generated corresponding to each original data element. The fitness values have been calculated of these four random numbers based on the original data element. Two random numbers have been selected based on the minimum fitness values among the four random numbers. The two selected numbers have been converted into binary numbers to make chromosomes. The crossover point has been selected randomly and single crossover operator has been applied between two chromosomes. The mutation operators have been applied on newly born chromosomes. Finally, chromosomes have been converted into real numbers. The fitness values again have been calculated between four data elements and the corresponding original data element. The data element with minimum fitness value has been treated as predicted value corresponding the original data elements. This process has been repeated at least 1000 times. The data element with minimum fitness value among 1000 iterations has been treated as predicted value corresponding to the original data elements. The whole process has been repeated for all original data elements. This type of work has not been made earlier; this is the reason for making this effort.

2. THEORY

2.1 Genetic Algorithm

GAs begins with a set of candidate solutions (chromosomes) called population. A new population is created from an old population in hope getting a better population. Solutions which are chosen to form new solutions (offspring) are selected according to r\their fitness. The more suitable the solutions are the bigger chances they have to reproduce. This process is repeated until some termination condition is satisfied

Some basic terminology of genetic algorithms is provided below.

2.1.1 Chromosomes

All living organisms consist of cells. In each cell there is the same set of chromosomes are the string of DNA that serves as a “blueprint” for the whole organism. In genetic algorithm terminology chromosome are the points in the search space and represented by the string of 0’s and 1’s.

2.1.2 Genetic Operator

There are three type of operator in the genetic algorithm, they are

2.1.2.1 Selection

Selection is based on the principle of survival of the fittest. It creates a new generation of chromosomes from the previous generation. Chromosomes with better fitness values increase in number while chromosomes with lesser values decrease in number or may be lost in near future.

2.1.2.2 Crossover

This operator randomly chooses a locus and exchanges the subsequences before and after that locus between two equal-size chromosomes to create two offspring. The position of the locus is determined randomly. The crossover operator roughly mimics biological recombination between two single chromosomes organisms. For example, consider the following parents and a crossover point at position 3:

Parent1	100|0111
Parent2	111|1000
Offspring1	100 1000
Offspring2	111 0111

In this example, offspring1 inherent bits in position 1, 2, and 3 from the left side of the crossover point from parent1 and rest from the right side of the crossover point from parent2. Similarly, offspring2 inherits bits in position 1, 2, and 3 from the left side of parent2 and the rest from the right side of parent1.

2.1.2.2 Mutation

Mutation is simply randomly flips some of the bits in chromosome. In binary code this involves changing a1 to 0 and vice versa. Mutation helps in avoiding the possibility of mistaking a local optimum for a global minimum. Mutation can occur at each bit position in string with some probability, usually very small value. When mutation is used with reproduction and crossover, it improves the global nature of the genetic algorithm search. For example, consider the following chromosome with mutation point at position 2:

Before mutation	1*0*00111
After mutation	1*1*00111

The 0 at position 2 flips to 1 after mutation.

2.1.2.3 Parameter of GA

The basic parameters of the GA are crossover probability and mutation probability.

2.1.2.3.1 Crossover Probability

It specifies how often crossover will be performed. If there is no crossover then children are exact copies of parents. If there is crossover the children are part of both the parents' chromosomes. If crossover probability is 100% then all children are made by crossover. If it is 0% then whole new generation is made from exact copies of chromosomes from old population.

2.1.2.3.2 Mutation probability

It specifies how often parts of chromosomes are mutated. If there is no mutation then children are directly copied after crossover without any change. If mutation is performed then one or more parts of chromosomes are changed. If mutation is 100% then whole chromosomes is changed. If 0% then nothing is changed. In general it should be very small value otherwise GA will change to random search.

2.2 Predicted Error and Average Predicted Error

The Predicted error and average Predicted error are calculated using the formula:

Predicted error = | (Predicted value - actual value)| / (actual value) * 100 %
Average Predicted error = (sum of Predicted errors) / (total no of errors).

2.3 Time Series Data

Quantities represent the values that have been taken by a variable over a period such as a month, quarter, or year. Time series data occurs wherever the same measurements are recorded on a regular basis. Time series data is a series of statistical data that is related to a specific instant or a specific time period as shown in Fig 1.

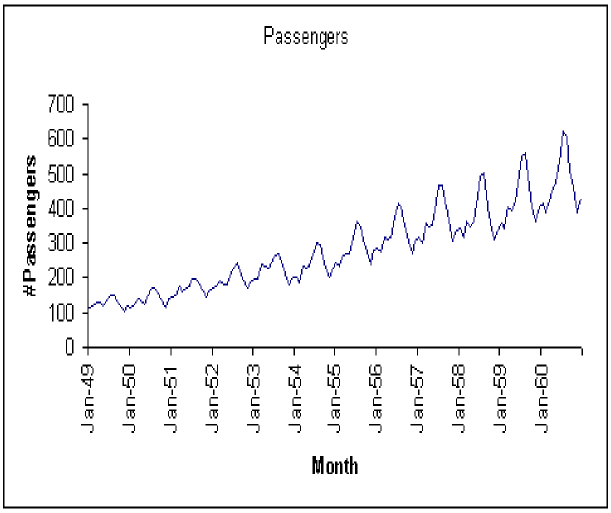

Fig 1: Time Series Data

2.4 Data used in this paper

The data set has been used in this paper collected from the research paper Satyendra Nath Mandal [13]. The data set is related to the shoot length of mustard plant. The shoot length is growing during the time duration so it can be taken as time series data.

3. PROPOSED ALGORITHM BASED ON GENETIC ALGORITHM

Table 1. Proposed Algorithm

Input:	**Number of decision variable (n), Lower and upper bounds of decision variables, Maximum number of iteration.**
Output:	**Best solution (X_{best}).**
Step 1:	Define Fitness Function f(x), f(x) = | Actual Value – Forecasted value |.
Step 2:	Randomly generate first population of n chromosomes, X_0[4] [n],i.e. for each data value 4 new population will be generated. set $X_{best} = X_0$.
Step 3:	Evaluate the fitness f(x) of each chromosome with the current population.
Step 4:	Select two parent chromosomes form current population according to their minimum fitness value.

Step 4.2:	Perform Crossover operation between two parents to form two new children.
Step 4.3:	Randomly select a chromosome of this current population and perform mutation operation on randomly selected locus (position in the chromosome).
Step 4.4:	Find the best chromosomes of this population according to their fitness value.
Step 4.5:	If X_{best} > best chromosome, set X_{best} = best chromosome.
Step 5:	If the termination condition is satisfied, stop, and return the best solution (X_{best}).
Step 6:	Otherwise go to step 3.

4. IMPLEMENTATION

Step 1: The parameters are initialized as follows:
Number of decision variables are 13;
The upper and lower limit of decision variables are: x_{min} = 15 and x_{max} = 70, they are defined as per available data;
Number of iteration = 10000;
The initial solution is generated with the help of a random generator (*rand()*). The random number between x_{min} and x_{max} is generated by the formula:
$P_{i,j} = x_{min}[i] + (x_{max}[i] - x_{min}[i]).rand()$, where i represents i^{th} row and j represents j^{th} column.
Then these numbers are converted into binary form.

Step 2: The error (e) is calculated as follows:

e = | Actual Value – Forecasted value |

Step 3: Select two chromosomes according to their minimum fitness value and make them parent 1 and parent2.

Step4: Randomly chooses a locus and exchanges the subsequences before and after that locus between parents to create two offspring. For example, consider the following parents and a crossover point at position 3:

Parent1	100|0111
Parent2	111|1000
Offspring1	100 1000
Offspring2	111 0111

In this example, offspring1 inherent bits in position 1, 2, and 3 from the left side of the crossover point from parent1 and rest from the right side of the crossover point from parent2. Similarly, offspring2 inherits bits in position 1, 2, and 3 from the left side of parent2 and the rest from the right side of parent1.

Step5: Randomly select a chromosome from the current population and perform mutation operation in randomly selected locus. For example, consider the following chromosome with mutation point at position 2:

Before mutation	1*0*00111
After mutation	1*1*00111

The 0 at position 2 flips to 1 after mutation.

Step6: The next move is to proceed to a better solution in the next population. If the better solution of the next population is superior to the current global best solution then it will be recorded as the best solution.

Step7: After successful iterations, i.e., after the termination criteria are satisfied, the best fitness value is obtained. The predictions are done with these obtained values.

Step8: The predicted values of the shoot length along with error percentage are calculated, from the best fitness value
The predicted values for the shoot length are calculated which are furnished in Table 2.

Table 2. Predicted values of shoot length

Actual Value	Predicted Value	Predicted Error (%)
19	-	-
24	24	0
28	28	0
33	32	3.03
37	36	2.70
41	41	0
45	45	0
49	48	2.04
54	55	1.85
57	57	0
59	59	0
63	63	0
66	66	0

Average Error: 0.740%

5. RESULT

The average error of different statistical models with soft computing models which have been applied on shoot length furnished in Table 3 below.

Table 3. Average Error of different models

Model	Average Error
Least Square Technique based on Linear Equation	2.88%
Least Square Technique based on Exponential Equation	8.80 %
Least Square Technique based on Logarithmic Equation	4.39 %
Least Square Technique based on Asymptotic Equation	3.134 %
Fuzzy Time Series	4.73
Artificial Neural network with fuzzy input	1.68
Genetic Algorithm	0.740

6. CONCLUSION AND FUTURE WORKS

From the Table 3, it has been observed that the Genetic Algorithm has been given minimum error. In this data set, the performance of Genetic Algorithm is better than others. In this paper, only one data set has been tested but in future more data set will be tested to predict the suitable membership function in prediction of time series data.

7. ACKNOWLEDGMENTS

The authors would like to thank to the All India Council for Technical Education (F.No-1-51/RID/CA/28/2009-10) for funding this research work.

8. REFERENCES

[1] Stephen D. Sloan, Raymond W. saw, James J.Sluss, “Genetic Algorithm forecasting for telecommunication products”, school of electrical and computer engineering, The University of Oklahoma, Norman, Oklahoma.

[2] Sultan H. Aljahdali and Mohammed E. El-Telbany, “Genetic Algorithm for optimizing of models in software reliability prediction”, ICGST-AIML Journal, vol 8,Issue I,2008.

[3] C. B. Lucasius, M. J. J. Blommers, L. M. C. Buydens, and G. Kateman, “A Genetic Algorithm for Conformational Analysis of DNA”, Cathelic University of Nijmegen Loboratory for Analytical Chemistry Toernooiveld , Netherland.

[4] Bahaa Ibraheem Kazem, Ali Ibrahim Mahdi, Ali Talib Oudah, “Motion planning for a robot arm by using genetic algorithm”, Jordan journal of mechanical and industrial Engineering, vol. 2, sep 2008,pp 131-136.

[5] Praveen Ranjan Srivastava and Tai-hoon Kim, “ Application of Genetic Algorithm in Software Testing”, International Journal of Software Engineering and its Applications, vol. 3, No. 4, October 2009.

[6] Reza Entezari-Maleki, Ali Movaghar, “A Genetic to Increase the Throughput of the Computational Grids”, International Journal of Grid and Distributed Computing, Vol. 4, No. 2, June 2011.

[7] Korhan Karabulut, Ahmet Alkan and Ahmet S. Yilmaz, “Long Term Energy Consumption Forecasting Using Genetic Programming”, Mathematical and Computational Applications, Vol. 13, No. 2, pp 71-80, 2008.

[8] Kathryn A. Dowsland, “Genetic Algorithm-A Tool for OR”, The Journal of the Operation Research Society, Vol. 47, No. 4, pp 550-561, Apr. 1996.

[9] Rahul Kala, Harsh Vazirani, Anupam Shukla and Ritu Tiwari, “Offline Handwriting Recognition using Genetic Algorithm”, IJCSI International Journal of Computer Science Issues, Vol. 7, No. 1, pp 16-25, March 2010.

[10] M. K. Deshmukh, C. Balakrishna Moorthy, “Application of Genetic Algorithm to Neural Network Model for Estimation of Power Potential”, Journal of Engineering Science and Management Education, Vol. 2, pp 42-48, 2010.

[11] Kristin Bennett, Michael C. Ferris and Yannis E. Ioannidis, “A Genetic Algorithm for Database Query Optimization”, Computer Science Department, University of Wisconsin.

[12] Sandeep Singh Rawat, Lakshmi Rajamani, “A Timetable Prediction for Technical Educational System using Genetic Algorithm”, Journal of Theoretical and Applied Information Technology, 2005.

[13] Satyendra Nath Mandal, J. Pal Choudhury, Dilip De and S. R. Bhadrachaudhuri , “A framework for development of Suitable Method to find Shoot Length at Maturity of Mustard Plant Using Soft Computing Model” , International journal of Computer Science and Engineering Vol. 2, No. 3, pp113-120

.