International Journal of Computing and Digital Systems
ISSN (2210-142X)
Int. J. Com. Dig. Sys. 14, No.1 (Sep-2023)

http://dx.doi.org/10.12785/ijcds/140172

An Efficient Framework For Software Maintenance Cost Estimation Using Genetic Hybrid Algorithm: OOPs Prospective

Mohammad Islam[1], Nafees Akhter Farooqui[2], Mohd. Haleem[3] and Syed Ali Mehdi Zaidi[4]

[1]*Assistant Professor, Department of Computer Science, Era University, Lucknow, UP., India*
[2]*Assistant Professor, School of Computer Applications, BBD University, Lucknow, UP., India*
[3]*Associate Professor, Department of Computer Science, Era University, Lucknow, UP., India*
[4]*Assistant Professor, Department of Computer Application, Shia P.G. College, Lucknow, UP., India*

Received 12 Jan. 2023 Revised 14 Jul. 2023, Accepted 31 Jul. 2023, Published 01 Sep. 2023

Abstract: One of the most significant exercises in software development is the software cost estimation, which leads to improvements in software engineering technique. The objectives of cost estimation, including effort, schedule, and manpower needs, are helpful advice for the establishment and operation of projects. This paper proposes an object-oriented software development framework for maintenance cost estimation using a genetic hybrid algorithm, a novel approach for estimating the maintenance cost of software systems. The framework combines object-oriented software development principles with genetic algorithm techniques to create a hybrid algorithm that can accurately estimate maintenance costs for software projects. The paper begins by discussing the importance of accurately estimating maintenance costs, as software maintenance can account for up to 60% of the total cost of a software system. Then the paper outlines the proposed framework, which consists of several components, including a cost estimation model and a genetic algorithm. The cost estimation model uses a set of parameters to predict maintenance costs, and the genetic algorithm is used to optimize the model's parameters for maximum accuracy using an appropriate data set. The paper then presents the results of an empirical study that was conducted to evaluate the effectiveness of the proposed framework. The study found that the framework was able to accurately estimate maintenance costs for several software projects by reducing root mean square error (RMSE) as well as mean absolute error (MAE). It has been observed that an improved prediction model over the regression model has been developed, resulting in lower RMSE and MAE values of 61.66 and 0.098818, respectively, as compared to the earlier ones from the regression model of 96.31 and 0.1718818, respectively. Overall, the Object-Oriented Software Development Framework for Maintenance Cost Estimation using Genetic Hybrid Algorithm Techniques provides a promising approach for accurately estimating maintenance costs for software systems, which can help organizations better manage their software development projects and budgets.

Keywords: Estimation of Costs, Software Maintenance, OOSD Framework, Genetic Algorithm, Regression Techniques

1. INTRODUCTION

The estimation of costs related to software development is being one of the most significant problems in the software development industry. The broad importance of forming a cost model approach for software is identical to the process for estimating anywhere else cost factor [1], [2], [3], [4]. Conversely, there are several characteristics of the approach that are unique to software prediction [2], [5]. The nature of software as a product drives so many of the distinctive elements of software estimation as well as their issues that arise as a result of the architecture of the assessing techniques [6]. Cost estimation sets the stage for a budget that will provide adequate funding to complete a project on time and in accordance with quality standards [2], [7]. It helps to improve the business software performance while increasing its adaptability to accept changing business environments. Whether you have developed highly secure & functional software for your business, it will showcase bugs and errors after some time. This is where you will need software maintenance services and how much software maintenance costs.

Even some of the challenges that hamper the ongoing development are to credit for the complexity of measuring that expenditure. Recognizing and characterizing the system to somehow be evaluated is one of the initial tasks in any estimate. Estimating the cost of the software is a sophisticated process that requires knowledge of several critical aspects of the project for which the estimate is being created [2], [8], [9].Software maintenance cost is

E-mail: mohdislam3@gmail.com, nafeesf@gmail.com, haleem.hbti@gmail.com, samzaididr@gmail.com

derived from the changes made to software after it has been delivered to the end user. The software does not "wear out," but it will become less useful as it matures and will always have bugs [10]. TCO is 75% software maintenance[9], [11]. Maintenance expenses for software include [12], [13], [14]:

- **Corrective maintenance**: expenses are incurred when software is modified to address issues that emerge after the original deployment. (Typically, 20% of the cost of maintaining software)
- **Adaptive maintenance**: Costs related to upgrading software to keep it useful in a changing business environment (representing 25% of total software maintenance costs)
- **Perfective maintenance**: Expenses associated with developing or refining a system of software to boost its overall efficiency (typically 5% of maintenance costs)
- **Enhancements**: continuous innovation-related costs, which frequently comprise 50% or more of the total expense of maintaining software.

Another study estimated that more than 90% of the entire cost of the software life was related to maintenance and software development management[15], [16].Identifying the cost estimates elements is essential for extending the life of HIS software, it may increase productivity and supply a native model for estimating the system's maintenance cost. In this way, the project manager will always have a reliable estimate of costs. The following categories of software maintenance operations are covered by the term[5], [9], [17]:

- A software product's smaller less than 50% new code components are rebuilt and redeveloped.
- The creation of more compact interface software packages that only require a minimal 20% or less modification to the original software.
- Modifications to the database design, and documentation, for the code of the software.
- The design and creation of significantly more than 20% of the source instructions that make up the current product interface software package that requires the least amount of modification to the current product.
- Data input, database value update, and data processing system operation.

The benchmark characterizes software management into four categories[18], [19]: corrective, preventive, adaptive, and perfect. These categories are distinguished by (a) the change's aim (repair or augmentation), and (b) the change's timing (proactive or reactive). Corrections are defined as corrective and preventive maintenance, whilst improvements are defined as adaptive and perfect maintenance. Several types of maintenance procedure are described in Figure 1.

Changes made to software after it has been made available to the end user are what drive up the cost of software maintenance. Figure 2 shows that maintenance related to a software product's growth or improvement is the biggest single cost factor.

This paper discusses the importance of cost estimation in software maintenance phase. A framework for software maintenance cost estimation is proposed and implemented using genetic algorithm. The proposed framework is as shown in Figure 3 uses combined approach of cost estimation model and genetic algorithm for accurately estimating maintenance cost in software project. The need and importance of cost estimation is discussed in this section.

The remaining section of the paper is described as: Related research works explain in section 1. Proposed framework is described in section 2. Model used is explained in section 3. Framework implementation is done in section 4. Section 5 explains result and discussion. Comparative analysis is discussed in section 6. Conclusion and future research is explained in section 7.

2. RELATED RESEARCH WORKS

There has been a lot of development in the field of software recently [5]. The software development life cycle consists of two stages: development and maintenance. Studies reveal indicate the maintenance part of a software's life cycle accounts for around 90% of the total cost of operation [20], [21]. The cost of software maintenance can be estimated and reduced by reducing the components through extraction and consideration. Estimates indicate a 50% growth over the previous two decades[22]. The proper estimation of the effort required to maintain provided software is aided by the breakdown of the overall effort into the component tasks that make up the complete process.

In a study, Henry Raymond (2007) created a predictive model for predicting software by combining estimation methodologies with the expertise of the development team, project manager, and director [23]. According to this framework, maintenance is essential for the success of IT initiatives. Although the efficient use of technology for time and cost estimation is essential, it is insufficient. The leadership team needs knowledge, knowledge integration, and knowledge exchange in order to estimate the exact time and cost[24], [25]. Traditionally, IT service companies advise their clients to use software maintenance services for the system's improved and consistent operation. According to Robert Glass, author of "Facts and Fallacies of Software Engineering," 60% of the expense of software maintenance is set aside for upkeep, and of that total, 60% is for solution enhancement[9], [20], [21].Several kinds of research that have been addressed and connected to frameworks for

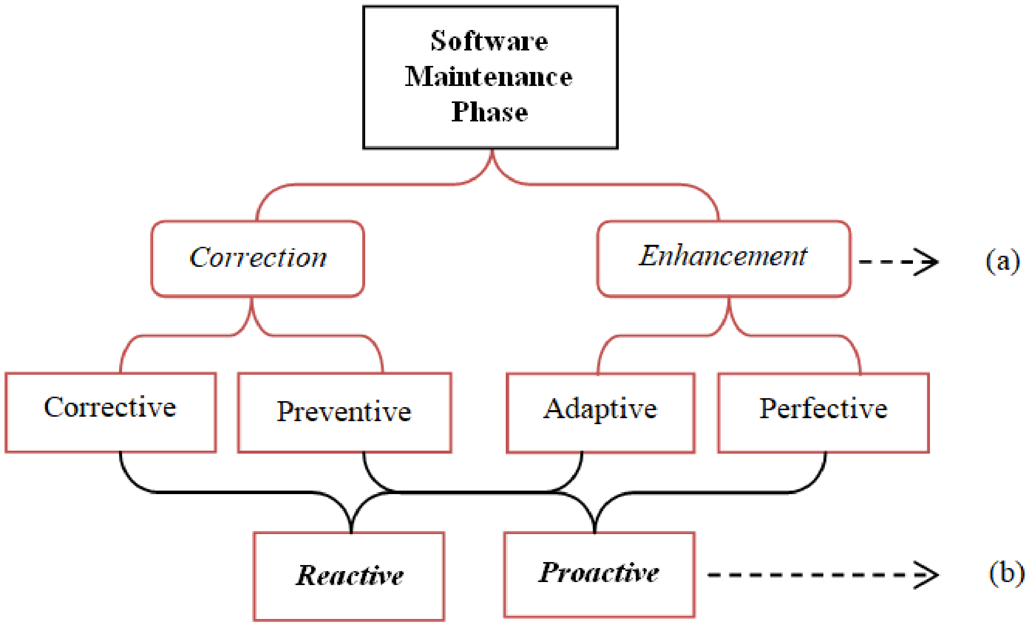

Figure 1. The several types of maintenance procedures

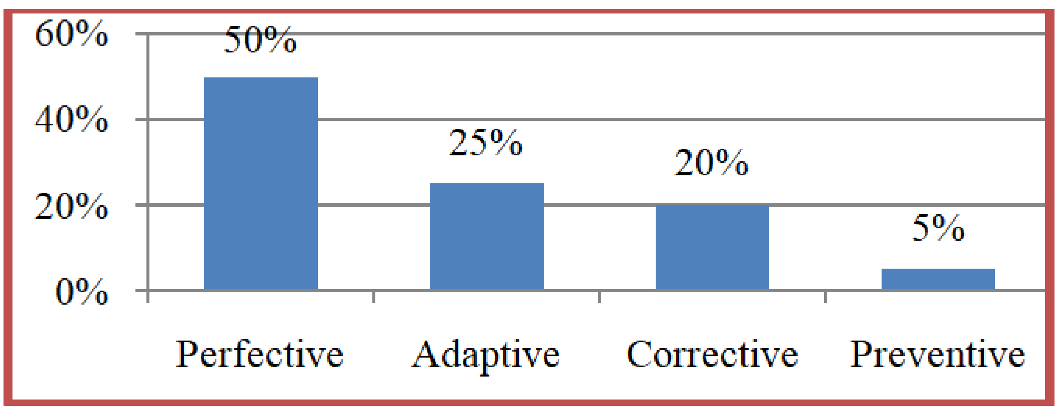

Figure 2. Percentages of each category's overall maintenance work

software maintenance and also cost-estimating approaches have been presented. Studies that have already been done are reviewed at, and their scope and limits are as follows: We stumbled over several comprehensive literature surveys that covered the application of GA in our investigation[26]. This article collected research and offered an explanation of how GA is used. Due to the inclusion of more research, our work is something that differs from this work. Also, we have already undertaken a careful study of modeling GA in that perspective to the design variables, the fitness value, analysis techniques, and used source datasets. Besides this, we have additionally listed some major benefits, and drawbacks as well as recommendations that may be beneficial to future studies trying to fill in the study deficiencies.

The new hybrid genetic k-means method means to determine the worldwide effective way and categorize a set of data into a predetermined number of groups [27]. The recommended GA eliminates the cheap crossover operator that is necessary for generating appropriate child chromosomes from parent chromosomes[27]. Distance-based mutation, a biased mutation operator designed specifically for clustering, was created. It modified the GA by employing the K-means technique, a standard gradient descent clustering approach. The k-means operator has been designed and implemented as a searching operator at the place of crossover inside the genetic K-means algorithm. The authors showed that the planned GKA accumulates to be the global optimum using the finite Markov chain theory. Additionally, it has been found that GKA searching was quicker than a number of other genetic clustering techniques[28].

The selection of cluster sites directly from the data set means faster the fitness assessment method by producing a point table in preparation as well as maintaining the distances between all pairs of data points, as part of a GA-based unmonitored clustering approach suggested [29]. A variable number of clusters are encoded using the binary format as an alternative to string representation, and improved selection, crossover, and mutation operators have been also implemented.

It introduces a novel algorithm that is placed on the conventional genetic algorithm as well as the new method provides certain refinements for the existing GA algorithm [29]. Using selection techniques for genetics lessened the likelihood of becoming caught in global optimization. In comparison to the conventional genetic algorithm, the novel approach enhances the search process and has a lower level of complexity. After reviewing the standard function optimization test reports, it has been determined that the novel method is more effective than the conventional genetic algorithm in terms of optimization accuracy. We also employ this new method for data categorization, and the results of the testing, demonstrate that it works so much better and is more accurate than KNN. It had been demonstrated that a genetic algorithm is much more reliable and simpler than a challenging technique for improving the architecture of an oceanographic experiment than simulated annealing[30]. [31]Determined that for detecting a group of sensors in the seas after they had drifted from their initial deployment point, an evolutionary programming strategy performed more consistently than conventional technologies [32].

3. PROPOSED FRAMEWORK

The dataset will first be well before using the Robust Regression Technique[33] to identify any outliers. This will be accomplished by giving each data point a rating. Multiple times reweighted least squares are a method for dynamically and repetitively weighting data. Each point is given identical values in the initial iteration as well as model parameters have been produced by ordinary least squares. Where, weights are recalculated in succeeding iterations such that locations furthest from predicted results in the preceding phase for assigned a lower weight. Then, generalized least squares are employed to recalculate the parameters of the model. The procedure is repeated until the variable predictions' values converge within a predetermined limitation [34].

To improve the estimation cost of software accuracy, the framework will next undergo further improvement of its parameters by adopting a genetic algorithm modeling technique. The genetic operators have been adopted like as crossover, mutation as well as the selection which operate to provide responses for progressing generations. Accordingly, the efficiency of the responses gets better over time and the procedure is finished after the best answer has been identified. The next step is to examine the model's performance using the RMSE and MAE factors and the proposed broad framework for the concerned current study is shown in Figure 3 below.

4. MODEL USED

A. GA Model

Natural selection & genetics are the cornerstones of genetic algorithms, which are meta-heuristic optimization algorithms. It is a methodology for supervised learning that incorporates an iterative procedure. Ordinary GA deals with problems through fitness statistics and genetic operators including crossover, mutation, as well as the selection which operate to develop outcomes for subsequent generations[35], [36]. As an outcome, the performance of the responses in later generations continues to get better; the procedure is completed when the correct solution is identified. The following are the responsibilities of genetic operators:

Mutation: Adapting innovative solutions in an effort that finds better responses.

Crossover: By merging components from different two populations, it only produces new components for the population.

Selection: Whenever the fitness function is implemented for measuring the existing response, fitness is a similar measure of how effectively a chromosome addresses the issue being addressed.

Fitness Function: The fitness function, also known as the objectives function is the function to improve standard algorithm optimization techniques, one should always establish the fitness function's minimum. The fitness functions should have been written as a document or anonymous method, and they should be passed as a method to handle model parameters based on the genetic algorithm mechanism.

The genetic algorithm applied for a population is subjected to the same set of operators periodically and continuously until a termination condition is achieved. The fitness function has been used in the present GA formulation is as below:

$$Minimize\ \ Y = abs(a - (\alpha + b * x(1) + c * x(2) + d * x(3))) \tag{1}$$

Where,
Y= minimization value for function objective,
abs= absolute value,
a= observed value of the effort estimation,
α= constant value obtained from the regression equation,
b= NEM, c = NSR, d = NOA,
x(1), x(2) & x(3)= values parameter of the regression equation to be obtained by optimization using the GA technique.

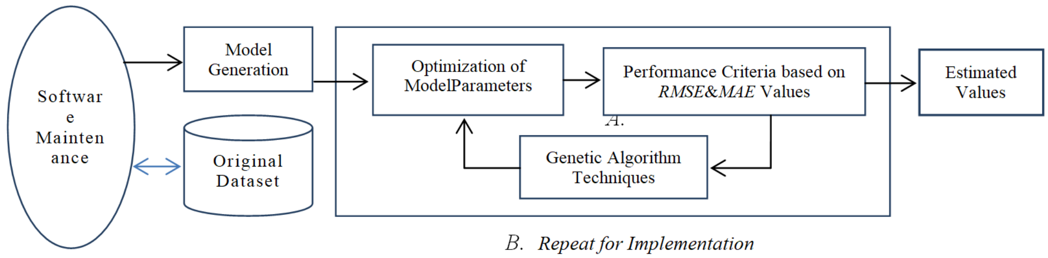

Figure 3. Proposed Framework for GA-Based Software Maintenance Cost Estimation

TABLE I. The implemented elements for developing models are listed

Projects Dataset	Actual Effort	NEM	NSR	NOA
1	287	142	197	170
2	396	409	295	292
3	471	821	567	929
4	1016	975	723	755
5	1261	997	764	1145
6	261	225	181	400
7	993	589	944	402
8	552	262	167	260
9	998	697	929	385
10	180	71	218	77
11	482	368	504	559
12	1083	789	362	682
13	205	79	41	98
14	851	542	392	508
15	840	701	635	770
16	1414	885	701	1087
17	279	97	387	65
18	621	382	654	293
19	601	387	845	484
20	680	347	870	304
21	366	343	264	299
22	947	944	421	637
23	485	409	269	451
24	812	531	401	520
25	685	387	297	812
26	638	373	278	788
27	1803	724	1167	1633
28	369	192	126	177
29	439	169	128	181
30	491	323	195	285
31	484	363	398	444
32	481	431	362	389
33	861	692	653	858
34	417	345	245	389
35	268	218	187	448
36	470	250	512	332
37	436	135	121	193
38	428	227	147	212
39	436	213	183	318
40	356	154	83	147

B. *Optimization Model*

The model used for the optimization of the parameters is obtained by carrying out the Robust Regression [33]analysis of the datasets.

$$The\ model\ is\ observed = \alpha + b * x(1) + c * x(2) + d * x(3) \quad (2)$$

Where, a, b and c are the model parameters.

Keeping these parameter values in the regression model, further optimization of these parameter values is being done using the GA optimization technique so as to be able to improve the model parameters and hence enhance the prediction accuracy. Where, the constant values of 'b', 'c' & 'd' have been upgraded to get a better solution approach. A very minimal root mean square error results from an attempt to obtain the prediction accuracy for the development effort extremely near the observed value.

C. *Data Set Used for Implementation*

Two successive semesters of graduate software engineering courses generated a wide range of the Table I dataset comprising from the examination of 40 software projects, which is adopted from Software Engineering Repository of PROMISE dataset that is open access for researching reason [37]. Initial experimental proof of the efficiency of the category-level technique has been provided by the use of such facts in the test procedure. It is obvious that employing student projects might compromise the experiment's external validity and, by extension, the method's evaluation; more study responses gathered from the industrial sector are required. However, we have made an effort to ensure that the validation procedure is as precise as feasible. Utilizing the Mat Lab platform, the framework was constructed using hybrid algorithm techniques.

Three separate object-oriented variables, each retrieved during the maintenance phase, make up the data set that was processed like the number of services requested (NSR), the number of external methods (NEM), and the number of attributes (NOA) with the help of actual effort as shown in Table I.

5. FRAMEWORK IMPLEMENTATION

For the present problem, the fitness function used to minimize abs $(\sum(_s\text{Measured} -_s \text{Computed}))$; Where s =

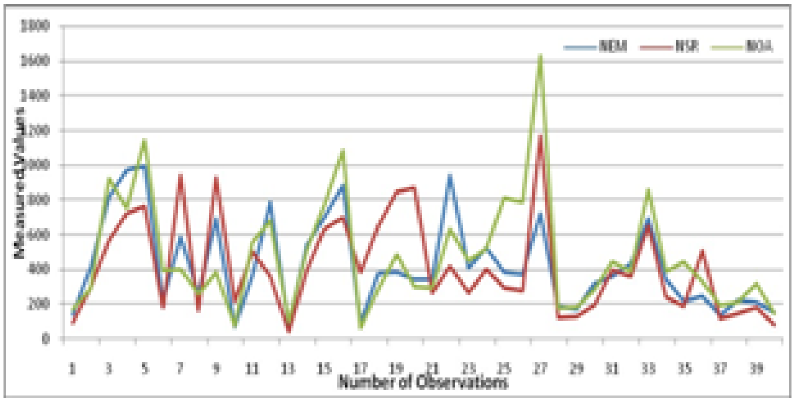

Figure 4. The plot of the various inputs & output variables are shown

software effort estimation value measured in man-months of model parameters for tuning and $_s$Comp has been calculated for value of effort based on the model adopted, and $_s$meas has been adopted for the measured value of effort.

The genetic algorithm is used to adapt the input variables of the system in order to decrease the overall square error as stated above. When recalled in the Mat Lab command window, the code for the requested optimal solution is generated as an M-file in the M-File editor.

Based on the values applied in the linear regression analysis as shown in equation (2) above, the lower and upper constraints of the three elements ”b”, ”c” & ”d” as indicated for the estimation model are specified.

A. Hybrid Genetic Algorithm for Software Cost Estimation Framework

The research shows that population size has a significant impact on the sampling ability of Gas[38]. Local search methods including the mimetic algorithm, Baldwinian, Lamarckian, and local search have been combined with GAs to address this issue. An appropriate balance between intensification and diversity is provided by this integration. Setting parameters is another issue with GA [39]. To make the succeeding generation from the existing population, the genetic algorithm is applied for three basic types of instructions in each process:

- Selection criteria just choose individuals, referring to as parents, who would also produce the population of the coming generation.
- Crossover provisions integrate the offspring of two parents for making a new generation.
- To produce children, mutation regulations expose every parent to unpredictable alterations.

The genetic algorithm’s action has been illustrated in the following phases:

- The technique produces some arbitrary basic population at the beginning.
- After then, the algorithm generates a series of new populations.
- The method creates the succeeding population during every phase utilizing members from the current generation.

Image denoising techniques, chemical process optimization, and many other optimization techniques may readily be used with genetic algorithms to improve efficiency. Improved solution quality, better efficiency, a guarantee of workable solutions, and enhanced control parameters are the key benefits of hybridized GA with other approaches. The algorithm applies several rounds to generating a new population:

1) Evaluates every individual person’s fitness value for assigning them their scoring value.
2) Filters the raw fitness values for providing a better choice of values that can be employed.
3) Choosing members, designated as parents, depending on the specific fitness.
4) The superior members of the present population have been picked from those with lesser fitness levels, and then they are transferred to the succeeding population.
5) Creates offspring from parents who are either established by randomly altering one parent, the mutation by merging the genetic elements of two crossover parents.
6) Consists of children who would also eventually replace the present population.
7) When any one of the ending requirements is achieved, the algorithm terminates.

6. RESULTS & DISCUSSIONS

Hybrid genetic algorithm has been used to implement the proposed framework. The optimized function value and

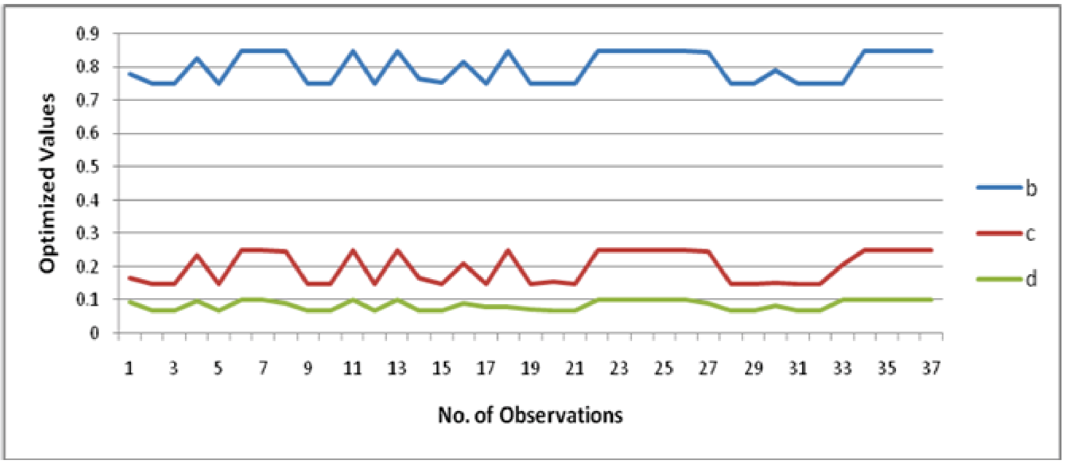

Figure 5. Plot of optimized parameters "b", "c" & "d" for datasets

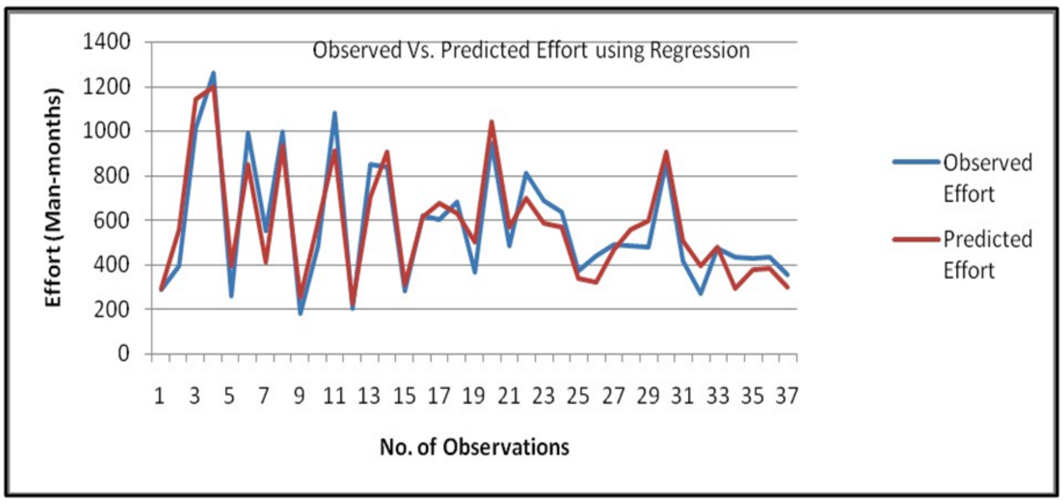

Figure 6. Plot for observed vs.predicted effort using RR

TABLE II. RMSE & MAE values using RR & GA

Prediction Criteria	Regression Techniques	Genetic Algorithm
RMSE	96.31	61.66
MAE	0.17188	0.098818

the optimal parameter values are calculated after optimization of the fitness function using the Mat Lab simulation tool. Software effort (s) is calculated using different parameter options for GA algorithm functions as given eq.3 below:

$$_s\text{Estimated} =_s \text{Observed} = \alpha + b * x(1) + c * x(2) + d * x(3) \quad (3)$$

Where: α=143.654790554716, is the constant term
b=0.802930099303848
c=0.212133114686023,
d=0.0811574301282941,
The lower and upper bound values used as follows:
LB = [0.75 0.15 0.07];
UB = [0.85 0.25 0.10];
On further analysis of the above equations (2) & (3), it has been observed that equation (3) for the model has been identified as the most advanced approach, resulting in a minimal RMSE score of 61.66 as compared to that of the regression model for the same datasets having RMSE value of 96.31. Also, MAE values for RR and GA-based models are 0.17188 and 0.098818 respectively, which again demonstrates the superiority of GA over other techniques (Table II and Figure 4 & 5).

Further, it clearly demonstrates that genetic algorithm optimization techniques have been successful in developing a better prediction model by lowering the RMSE value. Figure 5 below shows optimized parameter values of all the datasets using GA optimization.

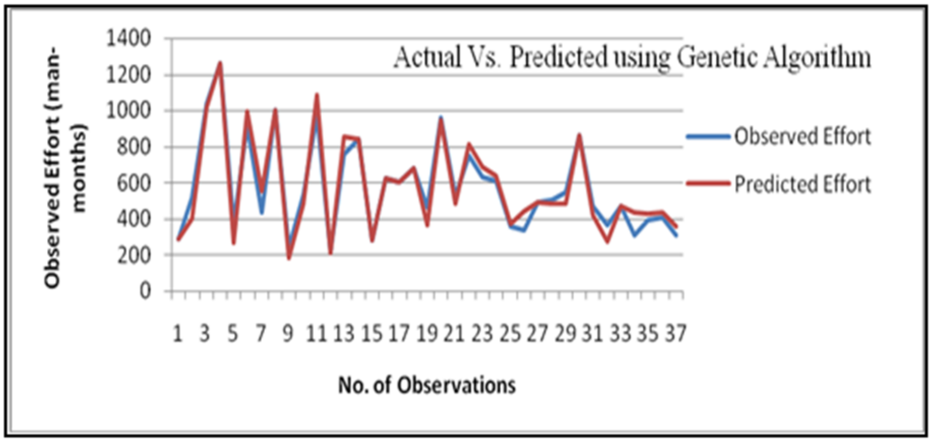

Figure 7. Plot for observed vs.predicted effort using GA

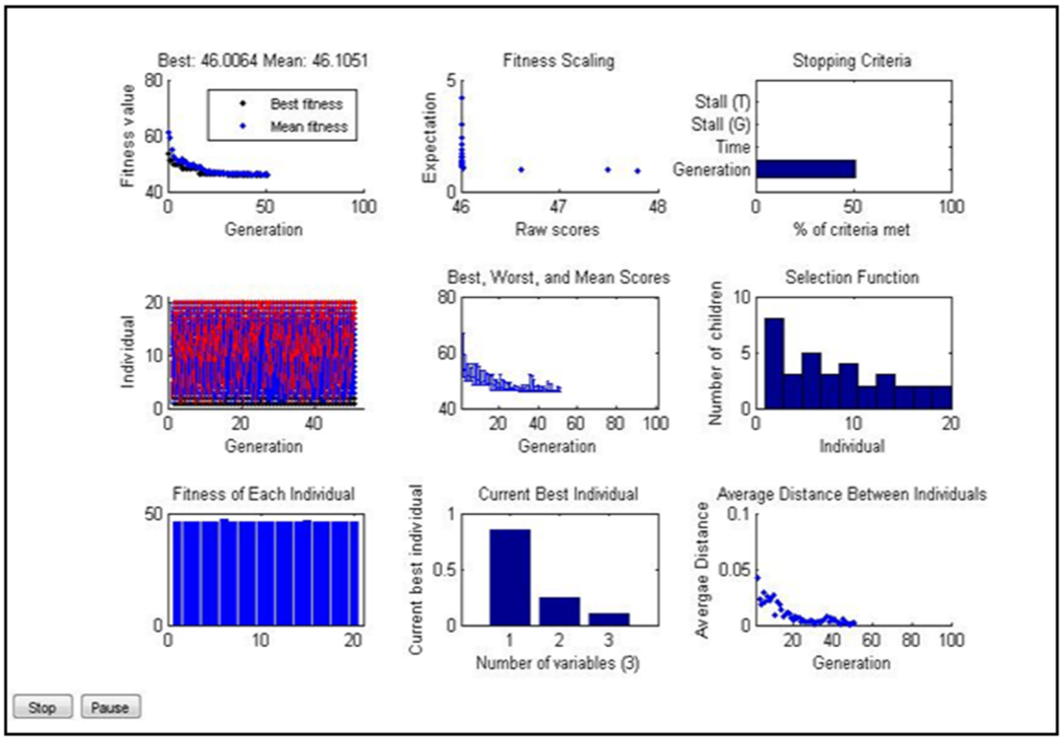

Figure 8. Output Plot of various GA plot functions in MatLab platform

7. COMPARATIVE ANALYSIS

From Table II it can be observed, the aforementioned choices that improved the prediction model over the regression one have been developed, resulting in a lower RMSE and MAE values of 61.66 and 0.098818 respectively as compared to the earlier one from the regression model as 96.31 and 0.1718818 respectively. Comparative plots of observed and predicted effort values as given in Figure 6 & 7, both for RR and GA based. It is seen that for both RR and GA based model the predicted values closely follows the observed trend, but still GA based trend is almost superimposed over one another. The various fitness function values of parameters which are to be optimized have been plotted as Figure 8.

8. CONCLUSION & FUTURE RESEARCH

This study examined genetic algorithm approaches in software for cost estimation as prediction methods. The heterogeneous and imprecise data used in this study might be handled using GA models, which are resilient, fast, and accurate. GA performs well for effort estimates prediction. Nonlinear data makes it a good quantitative method. An advanced model based on the robust regression model equations has been developed in the proposed investigation applying the genetic algorithm optimization methodology, in which the constant values "a", "b" & "c" are enhanced for producing a better method for solving. In the effort to reduce the root mean square error (RMSE) as well as mean absolute error (MAE), calculated values of estimating efforts are made to be very near to the measured value.

A GA model, using different options that are population, fitness, selection, mutation, crossover, hybrid, stopping and their combinations have been developed for estimation prediction. On the basis of "Results & Discussions", we observed that improved the prediction model over the regression that have been developed, resulting in a lower RMSE and MAE values of 61.66 and 0.098818 respectively as compared to the earlier one from the regression model as 96.31 and 0.1718818 respectively. In Future to decrease the issue of early confluence. To resolve the problem of computational complexity, new innovative crossover and mutation strategies will be use.

ACKNOWLEDGMENT

We heartily thank to the respondents who participated in this work.

References

[1] S. Wang, T. Liu, and L. Tan, "Automatically learning semantic features for defect prediction," in *Proceedings of the 38th International Conference on Software Engineering*, 2016, pp. 297–308.

[2] M. Haleem and M. R. Beg, "Impact analysis of requirement metrics in software development environment," in *2015 IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT)*. IEEE, 2015, pp. 1–6.

[3] M. Islam and V. Katiyar, "Development of a software maintenance cost estimation model: 4th gl perspective," *International Journal of Technical Research and Applications*, vol. 2, no. 6, pp. 65–68, 2014.

[4] M. Islam, V. Katiyar, and P. S. Q. Abbas, "Estimating cost of software maintenance using component based 4th gl approach," *International Journal of Latest Technology in Engineering, Management & Applied Science*, vol. 4, no. 6, pp. 86–92, 2015.

[5] M. Haleem, M. R. Beg, and F. Ahmad, "Overview of impact of requirement metrics in software development environment," *International Journal of Advanced Research in Computer Engineering & Technology*, vol. 2, no. 5, pp. 1811–1815, 2013.

[6] R. Verdecchia, P. Kruchten, P. Lago, and I. Malavolta, "Building and evaluating a theory of architectural technical debt in software-intensive systems," *Journal of Systems and Software*, vol. 176, p. 110925, 2021.

[7] M. Haleem, M. F. Farooqui, and M. Faisal, "Tackling requirements uncertainty in software projects: A cognitive approach," *International Journal of Cognitive Computing in Engineering*, vol. 2, pp. 180–190, 2021.

[8] K.-K. Phoon, J. Ching, and T. Shuku, "Challenges in data-driven site characterization," *Georisk: Assessment and Management of Risk for Engineered Systems and Geohazards*, vol. 16, no. 1, pp. 114–126, 2022.

[9] M. Haleem, M. F. Farooqui, and M. Faisal, "Cognitive impact validation of requirement uncertainty in software project development," *International Journal of Cognitive Computing in Engineering*, vol. 2, pp. 1–11, 2021.

[10] J. R. Fraser, R. Quail, and B. J. Simkins, "Questions asked about enterprise risk management by risk practitioners," *Business Horizons*, vol. 65, no. 3, pp. 251–260, 2022.

[11] B. Bharadwaj, D. Pullar, L. S. To, and J. Leary, "Why firewood? exploring the co-benefits, socio-ecological interactions and indigenous knowledge surrounding cooking practice in rural nepal," *Energy Research & Social Science*, vol. 75, p. 101932, 2021.

[12] K.-a. Jang and W.-J. Kim, "A method of activity-based software maintenance cost estimation for package software," *The Journal of Supercomputing*, pp. 1–21, 2021.

[13] K. Benzekki, A. El Fergougui, and A. Elbelrhiti Elalaoui, "Software-defined networking (sdn): a survey," *Security and communication networks*, vol. 9, no. 18, pp. 5803–5833, 2016.

[14] R. Seethamraju, "Adoption of software as a service (saas) enterprise resource planning (erp) systems in small and medium sized enterprises (smes)," *Information systems frontiers*, vol. 17, pp. 475–492, 2015.

[15] H. Islam, M. Jollands, and S. Setunge, "Life cycle assessment and life cycle cost implication of residential buildings—a review," *Renewable and Sustainable Energy Reviews*, vol. 42, pp. 129–140, 2015.

[16] S. Bilgaiyan, S. Sagnika, S. Mishra, and M. Das, "A systematic review on software cost estimation in agile software development." *Journal of Engineering Science & Technology Review*, vol. 10, no. 4, 2017.

[17] K. Mao, L. Capra, M. Harman, and Y. Jia, "A survey of the use of crowdsourcing in software engineering," *Journal of Systems and Software*, vol. 126, pp. 57–84, 2017.

[18] C. Canizares, T. Fernandes, E. Geraldi, L. Gerin-Lajoie, M. Gibbard, I. Hiskens, J. Kersulis, R. Kuiava, L. Lima, F. DeMarco *et al.*, "Benchmark models for the analysis and control of small-signal oscillatory dynamics in power systems," *IEEE Transactions on Power Systems*, vol. 32, no. 1, pp. 715–722, 2016.

[19] Y. Chen, Y. Chen, C. Shi, Z. Huang, Y. Zhang, S. Li, Y. Li, J. Ye, C. Yu, Z. Li *et al.*, "Soapnuke: a mapreduce acceleration-supported software for integrated quality control and preprocessing of high-throughput sequencing data," *Gigascience*, vol. 7, no. 1, p. gix120, 2018.

[20] C. Salazar, A. Lelah, and D. Brissaud, "Eco-designing product service systems by degrading functions while maintaining user satisfaction," *Journal of Cleaner Production*, vol. 87, pp. 452–462, 2015.

[21] P. R. Daugherty and H. J. Wilson, *Human+ machine: Reimagining work in the age of AI*. Harvard Business Press, 2018.

[22] S. L. Colby and J. M. Ortman, "Projections of the size and composition of the us population: 2014 to 2060. population estimates and projections. current population reports. p25-1143." *US Census Bureau*, 2015.

[23] R. Purvis, R. M. Henry, S. Tams, V. Grover, J. D. McGregor,

and S. Davis, "The impact of residual risk and resultant problems on information systems development project performance," *Project Management Journal*, vol. 47, no. 4, pp. 51–67, 2016.

[24] H. Kerzner, *Project management: a systems approach to planning, scheduling, and controlling*. John Wiley & Sons, 2017.

[25] K. Schwalbe, *Information technology project management*. Cengage Learning, 2015.

[26] N. Alaoui, A. B. H. Adamou-Mitiche, and L. Mitiche, "Effective hybrid genetic algorithm for removing salt and pepper noise," *IET Image Processing*, vol. 14, no. 2, pp. 289–296, 2020.

[27] L. D. Pacifico and T. B. Ludermir, "An evaluation of k-means as a local search operator in hybrid memetic group search optimization for data clustering," *Natural Computing*, vol. 20, pp. 611–636, 2021.

[28] S. H. Toman, M. H. Abed, and Z. H. Toman, "Cluster-based information retrieval by using (k-means)-hierarchical parallel genetic algorithms approach," *arXiv preprint arXiv:2008.00150*, 2020.

[29] L. Andeobu, S. Wibowo, and S. Grandhi, "Artificial intelligence applications for sustainable solid waste management practices in australia: A systematic review," *Science of The Total Environment*, p. 155389, 2022.

[30] A. Afram, F. Janabi-Sharifi, A. S. Fung, and K. Raahemifar, "Artificial neural network (ann) based model predictive control (mpc) and optimization of hvac systems: A state of the art review and case study of a residential hvac system," *Energy and Buildings*, vol. 141, pp. 96–113, 2017.

[31] B. Celik and J. Vanschoren, "Adaptation strategies for automated machine learning on evolving data," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 9, pp. 3067–3078, 2021.

[32] N. A. Farooqui, Ritika, and A. Saini, "Sentiment analysis of twitter accounts using natural language processing," *Int. J. Eng. Adv. Technol.*, vol. 8, no. 3, pp. 473–479, 2019.

[33] V. Verardi and C. Croux, "Robust regression in stata," *The Stata Journal*, vol. 9, no. 3, pp. 439–453, 2009.

[34] N. A. Farooqui, "A machine learning approach to simulating farmers' crop choices for drought prone areas," in *Proceedings of ICETIT 2019: Emerging Trends in Information Technology*. Springer, 2020, pp. 472–481.

[35] F. Herrera and M. Lozano, "Adaptive genetic operators based on coevolution with fuzzy behaviors," *IEEE Transactions on Evolutionary computation*, vol. 5, no. 2, pp. 149–165, 2001.

[36] N. A. Farooqui, A. K. Mishra, and R. Mehra, "Automatic crop disease recognition by improved abnormality segmentation along with heuristic-based concatenated deep learning model," *Intelligent Decision Technologies*, vol. 12, no. 2, pp. 407–429, 2022.

[37] "index @ promisedata.org."

[38] T. A. El-Mihoub, A. A. Hopgood, L. Nolle, and A. Battersby, "Hybrid genetic algorithms: A review." *Eng. Lett.*, vol. 13, no. 2, pp. 124–137, 2006.

[39] N. A. Farooqui and R. Ritika, "An identification and detection process for leaves disease of wheat using advance machine learning techniques," *Biosci. Biotechnol. Res. Commun*, vol. 12, no. 4, pp. 1081–1091, 2019.

Mohammad Islam received his UG degree in B.Sc. from Lucknow University, Lucknow, India, in 2000, PG degree in MCA from IGNOU, New Delhi, India in 2008, and Ph.D. in Computer Science from Shri Venkateshwara University, Amroha, U.P., India in 2020. He is currently working as an Assistant Professor & Head in the Department of Computer Science, at Era University, Lucknow, U.P., India. He has published many research papers, and book chapters in conference proceedings and international journals. He has an overall experience of above more than thirteen years. His research interests include Software Engineering, Computer Networks, Artificial Intelligence, Cyber Security, etc.

Nafees Akhter Farooqui has a Ph.D. (Computer Science), from DIT University, Dehradun, Uttarakhand, India. He received his master's degree in computer application from Integral University, Lucknow, India in 2010, and a graduation degree in Statistics from Aligarh Muslim University, Aligarh, Uttar Pradesh, India in 2005. He has 13 years of teaching experience and presently working as Assistant Professor in the School of Computer Applications, BBD University, Lucknow, UP, India. His research interests include Artificial Intelligence, Machine Learning, Deep Learning, Computer Vision, Pattern Recognition, Natural Language Processing and Data Mining. He has published approx. 25 papers in International Journals and Conference proceedings including Web of Science and Scopus indexed Journals. He also published various book chapters in Scopus indexed Journals. He has published two books for National Publishers etc.He has been a member of International Association of Engineers (IAENG) since 2017, ACM since 2011. He guided various projects of UG and PG level Students. He received awards from various professional bodies.

Mohd Haleem He had completed a BSc (Hons.) in Statistics from AMU, MCA from HBTI, Kanpur, M.Tech. (CSE) from Integral University, Lucknow, Uttar Pradesh, and a PhD (CS) from Integral University, Lucknow, Uttar Pradesh, India. He had more than 12 years of teaching experience. Currently working as an Associate Professor in the department of computer science at Era University, Lucknow, India. Published more than 10 research papers in reputed international journals and conferences. Member of various professional societies. The research area is requirement engineering, requirement uncertainty data science.

Syed Ali Mehdi Zaidi received his graduation degree in B.Sc. from Lucknow University, Lucknow, India, in 1997, a post-graduation degree in MCA from IGNOU, New Delhi, India in 2006, and Ph.D. in Computer Science from Shri Venkateshwara University, Amroha, U.P, India in 2020. He is currently working as an Assistant Professor in the Department of Computer Application, at Shia P.G College, Lucknow, U.P, India. He has published many research papers in conference proceedings and international journals. He has an overall experience of above more than 15 years. His research interests include Software Engineering, Artificial Intelligence, Data Mining, etc.