ORIGINAL RESEARCH ARTICLE UHD JOURNAL OF SCIENCE AND TECHNOLOGY

A New Approach for Software Cost Estimation with a Hybrid Tabu Search and Invasive Weed Optimization Algorithms

Hoshmen Murad Mohamedyusf[1], Hawar Othman Sharif[2], Mazen Ismaeel Ghareb[3]

[1]Department of Plastic Arts, Halabja Fine Arts Institute, Halabja, Iraq, [2]Department of Computer Science, College of Science, University of Sulaimani, Iraq,[3]Department of Computer Science, College of Science and Technology, University of Human Development, Kurdistan Region, Iraq.

ABSTRACT

Due to the ever-increasing progress of software projects and their widespread impact on all industries, models must be designed and implemented to analyze and estimate costs and time. Until now, most of the software cost estimation (SCE) has been based on the analyst's experiences and similar projects and these models are often inaccurate and inappropriate. The project will not be finished in the specified time and will include additional costs. Algorithmic models such as COCOMO are not very accurate in SCE. They are linear and the appropriate value for effort factors is not considered. On the other hand, artificial intelligence models have made significant progress in the cost estimation modeling of software projects in the past three decades. These models determine the correct value for effort factors through iteration and training, providing a more accurate estimate compared to algorithmic models. This paper employs a hybrid model incorporating the Tabu Search (TS) algorithm and the Invasive Weed Optimization (IWO) algorithm for SCE. IWO algorithm solutions are improved using the TS algorithm. The NASA60, NASA63, NASA93, KEMERER, and MAXWELL datasets are used for the evaluation. The proposed model has been able to reduce the MMRE rate compared to the IWO algorithm and the TS algorithm. The proposed model on the NASA60, NASA63, NASA93, KEMERER, and MAXWELL datasets obtained values of MMRE of 15.43, 17.05, 28.75, 58.43, and 22.46, respectively.

Index Terms: COCOMO Model, Tabu Search Algorithm, Invasive Weed Optimization Algorithm, Software Cost Estimation, Optimization

1. INTRODUCTION

A more comprehensive examination of a project's viability and more effective management of the software development process enable development organizations to significantly reduce project risks [1], [2]. Possessing an accurate and reliable software cost estimation (SCE), especially at the beginning of software projects, is considered a crucial factor for project success. The accurate estimation of the production SCE gives the project manager strong support for making different decisions during the software life cycle. The amount of time and effort required to develop a software project should also be known by analysts, designers, programming teams, and other software production forces [1], [3].

The project manager cannot estimate how much time, people, and other resources he will need to finish the project without a solid SCE. The project may face dangers if the diagnosis is incorrect, and it is impossible to gauge the likelihood of success [4]. Hence, metaheuristic algorithms have been employed in this article to meticulously examine the influential factors in SCE. These algorithms do not necessitate complex mathematical equations, and they

Access this article online

DOI: 10.21928/uhdjst.v8n1y2024.pp42-54 | E-ISSN: 2521-4217 | P-ISSN: 2521-4209

Corresponding author's e-mail: Hoshmen Murad Mohamedyusf, Department of Plastic Arts, Halabja Fine Arts Institute, Halabja, Iraq. hoshmandmorad@gmail.com

Received: 28-11-2023 Accepted: 07-01-2024 Published: 13-02-2024

address optimization problems without delving into the internal intricacies of the problem's performance [1]. These algorithms are computationally simple but powerful, and they are not limited by restrictive assumptions regarding the search space. A common approach that can be used in SCE is to use an optimal function along with an algorithmic method to find the values of the cost estimation factors that create the most optimality [29]. Still, when the dimensions of the problem become larger with the increase in the number of factors and response variables, algorithmic methods will be unable to find real solutions. Therefore, using a hybridization of Invasive Weed Optimization (IWO) [6] and Tabu Search (TS) [7] algorithms is a new methodology for optimizing SCE. Then, we will compare the effectiveness of IWO and TS algorithm with algorithmic models and we will present the best solution.

Accurate effort estimation significantly influences the outcome of software projects, determining their success or failure. Inaccurate estimations pose challenges for both companies and clients, with underestimations leading to projects exceeding planned budgets and overestimation resulting in resource wastage [8], [28]. Precision in effort estimation is crucial for risk assessment, resource allocation, and progress monitoring. Although preventing all risks in software development is often unattainable, our proposed effort estimation model can assist in mitigating risks associated with resource assignment and task scheduling. To address this issue, researchers, managers, and developers have explored various methods to enhance the accuracy of effort estimation during the early stages of software projects. For instance, managers can utilize the COCOMO model to predict the effort required at the medium level of software development. This ongoing estimation process contributes to minimizing the recurrence of risks in future endeavors.

During the past three decades, variations of methods such as COCOMO 81, COCOMO 2000, COCOMO I, and COCOMO II have been used to estimate software costs [9]. These methods are linear and do not have accurate and correct estimations. The SCE and its measurement are crucial for controlling costs in the software life cycle. One of the primary contributors to sharp increases in development and maintenance expenses is software effort estimation (SEE) (Araújo *et al.*, 2012). It is a factor that is less known in the development of software projects and cannot be easily identified or described. It is often ignored during the project planning process. In this paper, the hybridization of IWO and TS algorithms is employed to predict SCE. The margin of error can be minimized through precise measurement and control of estimation factors. The level of collaboration needed to comprehend the program is often decided by SCE and SEE. The cost in the software development phase greatly affects the effort required to test and debug the program, modules, and subsystems. Therefore, it is necessary to provide an effective and correct model for SCE and SEE. These are the key contributions of this paper:

- In this paper, the main goal is to estimate the amount of effort factors using the hybrid model and reduce the amount of MMRE error. Usually, linear models such as COCOMO have a large amount of error and do not determine the value of the factors based on the size of the projects.
- Improvement of IWO algorithm solutions using the TS algorithm. The TS algorithm is designed to escape the trap in local optimization. It avoids recently visited solutions in the future by preserving memory (Tabu list) and leads to a more comprehensive exploration of the solution space.
- Use the TS algorithm to explore the search space and find the best answers, avoiding local optima.
- Evaluation of the hybrid model on five data sets (NASA60, NASA63, NASA93, KEMERER, and MAXWELL).

This paper's general organization is as follows: Section 2 reviews related works that use artificial intelligence algorithms in the field of SCE. The proposed model and its phases are explained in Section 3. The proposed model is assessed using various datasets in Section 4. In addition, Section 5 examines conclusions and future research.

2. RELATED WORKS

The SCE and SEE have been the subject of substantial research. Most of them done in this field have dealt with the impact of cost on software projects at the head of them is the quality and cost of the product. The attention of managers and engineers of software projects to the SCE is to control and predict the quality and productivity of the software. In this section, we discuss artificial intelligence techniques and algorithmic models that have tested and analyzed software projects. SCE and SEE models are proposed based on fuzzy C-means clustering and metaheuristic algorithms [1]. Compared to previous algorithms, the proposed algorithm has demonstrated superior results in cost prediction. The International Software Benchmarking Standards Group (ISBSG) dataset has been utilized for a linear regression model based on the criteria of relative error values [10]. The

ISBSG dataset includes 3042 software projects from different companies during the last 6 years. The correlation coefficient of factors in linear regression is calculated according to Eq. (1). In the Eq. (1), x is the independent variable, y is the dependent variable, ε is the partial error value, k is the number of variables, and α_0 and β are constant values. The act_i parameter is the actual value, est_i is the estimated value, n is the number of projects, and x is the pred value.

$$y = (\alpha_0 + \log(\prod_{i=1}^{k} x_i^{\alpha_i}) + \varepsilon)^{\beta} \quad (1)$$

$$MRE_i = \frac{|act_i - est_i|}{act_i} \times 100 \quad (2)$$

$$MMRE = \frac{1}{n}\sum_{i=1}^{n} MRE_i, i = 1, 2, ..., n \quad (3)$$

$$PRED(x) = \frac{1}{n} \times \sum_{i=1}^{n} \begin{cases} 1, & if MRE \le x \\ 0, & otherwise \end{cases} \quad (4)$$

In different implementations, the magnitude of relative error (MRE) has less error compared to the real data set. The optimization model of the artificial neural networks (ANN) and particle swarm optimization (PSO) derived from the optimization of the PSO and the ANN has been implemented on NASA93, COCOMO81, and MAXWELL datasets [11]. In this model, a PSO is used to train and test data in ANN. The results showed that the value of PRED (0.25) in the PSO-ANN has higher accuracy compared to other models. In addition, the classification and regression tree (CART) model is less accurate than the step-wise regression (SWR) model. Artificial bee colony (ABC)-ANN model derived from ABC algorithm and ANN has been implemented on COCOMO81, NASA93, and COCOMO_SDR datasets [12]. In the ABC-ANN model, the link function of the ABC algorithm is used to train and test the data in the ANN. The results showed that the relative error value in the ABC-ANN is less compared to COCOMO.

The hybrid chi-means model and PSO algorithm have been implemented on the COCOMO81 dataset [13]. K-means has been employed to cluster similar projects together, and the PSO algorithm has been utilized to determine effort factor values. Evaluation criteria include mean absolute relative error (MARE), variance accounted for (VAF), and variance absolute relative error (VARE). The results showed that the values of VAF, MARE, and VARE for the COCOMO model are 94.66, 17.35, and 3.72, respectively. For the Chi-means, the hybrid algorithm is equal to 95.47, 20.76, and 4.12, respectively. Therefore, the accuracy of the hybrid algorithm is higher compared to COCOMO.

$$VAF = \left[1 - \frac{var\ (actual_{effort} - estimated_{effort})}{var\ (actual_{Effort})}\right] \times 100 \quad (5)$$

$$MARE = Mean\left[\frac{abs\ (actual_{effort} - estimated_{effort})}{(actual_{effort})}\right] \times 100 \quad (6)$$

$$VARE = var\left[\frac{abs\ (actual_{effort} - estimated_{effort})}{(actual_{effort})}\right] \times 100 \quad (7)$$

The OCFWFLANN model is implemented by combining genetic algorithm and functional link ANN on the NASA93 dataset [14]. Genetic algorithm has been used to test and train data. The results showed that the OCFWFLANN model has a lower mean relative error value compared to functional link ANN models and SWR and CART models. In [27], a new model is implemented by hybridizing genetic algorithm and functional link ANN on the NASA93 dataset. The data have been trained and tested using a genetic algorithm. The results showed that the proposed model has a lower mean relative error value compared to functional link ANN models and SWR and CART models.

The multi-objective PSO (MPSO) model has been implemented on the COCOMO dataset [15]. This model has been utilized to determine the values of "a" and "b," representing cost parameters. The initial stage yielded results with 20 projects, 100 iterations, and 50 particles. The second stage involved 21 projects, 100 iterations, and 50 particles. The results in the first stage showed that the MARE and PRED (0.25) in the COCOMO model are 16.13 and 20, respectively. And, in the MPSO model, it is equal to 9.01 and 24, respectively. In the second stage, the MARE and PRED (0.25) in the COCOMO model are 18.15 and 17, respectively. In the MPSO model, it is equal to 20.97 and 15, respectively.

The hybrid model of the TS algorithm and genetic algorithm has been implemented on NASA60, NASA63, and NASA93

datasets [16]. The data have been trained and tested using a genetic algorithm. The TS algorithm's factors have been optimized using a genetic algorithm. The results showed that the TS algorithm-genetic algorithm model has a lower error value compared to the genetic algorithm and the TS algorithm. A harmony search algorithm is implemented on the NASA93 dataset [17]. Finding appropriate effort factor values is the aim of the harmony search method. The proposed model's evaluation function has the same value as the MMRE. The number of program execution iterations is 5000 times. The results show that the harmony search algorithm has improved the MMRE by about 21% compared to COCOMO.

Genetic Algorithm – Bayesian Network and PSO Algorithm – Bayesian Network under Bayesian Belief Network are proposed for SCE [20]. The evaluation is done on the NASA63 data set with 15 effort factors. 40 and 20 projects have been used for training and testing data, respectively. The goal of genetic algorithm and PSO algorithm is to optimize the Bayesian belief network and reach the optimal value. The results showed that the average value of the relative error in the genetic algorithm – Bayesian network and PSO algorithm – probabilistic Bayesian network models is lower compared to COCOMO. Hybrid models of PSO algorithm – fuzzy clustering and PSO algorithm – learning automata have been proposed to estimate the SCE [18]. The evaluation is done on the NASA60 dataset. The learning automata model strategy provides the possibility for particles to achieve multiple local optima according to the reward criteria for the PSO algorithm. The trials' findings demonstrate that the hybrid model has a higher MMRE than the PSO algorithm – fuzzy clustering model.

SCE is simulated using linear regression techniques, ANN, support vector machine (SVM), and KNN [19]. The dependency of the effective qualities in SCE may be discovered using the linear regression model. The influencing elements of the hybrid model for estimation have been trained using both ant colony optimization (ACO) and the genetic test technique. Comparing the results to the COCOMO model, better outcomes have been achieved. One of the popular techniques in SCE is multi-layered ANN. The findings reveal that the multi-layer ANN has delivered a much better estimate than the COCOMO model in more than 90% of the situations. As a result, it can be said that algorithmic approaches are a strong complement to AI-based methods.

Machine learning techniques such as hybrid ANN, SVM, and genetic algorithm have been proposed for SCE [22]. Optimizing input data factors and optimizing the parameters of SVM and ANN methods are the two main goals of using genetic algorithms. *Desharnais*, NASA, COCOMO Albrecht, KEMERER, and *KotenGray* datasets are used for the evaluation. The hybrid models based on the genetic algorithm have improved significantly, according to the trials' findings across all data sets. In comparison to the SVM and multilayer ANN models, the criterion for assessing the MMRE in the hybrid models has a smaller error value. Also, the PRED criterion has higher accuracy in hybrid models. Detailed planning for the development of software projects increases efficiency and optimal use of resources and reduces time.

SCE has been analyzed using fuzzy functions. Methods employing triangular membership function, trapezoidal membership function, and Gaussian membership function have been utilized for evaluation on the NASA93 dataset [23]. The proposed method is combined with the COCOMO II model. COCOMO II model includes 17 effort factors and 5 scale factors. Evaluation and results are done on 10 projects from NASA93 dataset. The results of their experiments show that fuzzy methods have better accuracy in SCE and have a lower relative error value compared to the COCOMO II model. The combined genetic algorithm – radius function model has been proposed to estimate the cost of software projects [24]. The evaluation is done on the COCOMO81 dataset. The training and testing of the network data are considered equal to 80% and 20%, respectively. Genetic algorithm has been used to optimize the training and testing data of the basic radial function network. The results show that the average errors of relative error and mean square error in Kokomo model are equal to 2.92 and 0.0287, respectively. The average value of the relative error in the basic radial function model is equal to 0.4220 and 0.9665, respectively.

A hybrid model of genetic algorithm and fuzzy logic (GFUZZY) is presented for SCE [25]. Four hybrid data sets taken from NASA2 and COCOMO81 software projects have been used for evaluation. A new hybridization of multilayer ANN and COCOMO II has been proposed for SCE [26]. The test results demonstrate that the hybrid model is less inaccurate than the COCOMO II model. In comparison to the COCOMO II model, the COCOMO-multilayer ANN model exhibits two lower error values for the MMRE and PRED criteria. The MMRE on 63 projects using the COCOMOII model is equal to 0.58 and by the multi-layer ANN-COCOMO model is equal to 0.41. In Table 1, the models presented by researchers to SCE have been evaluated and compared.

3. PROPOSED MODEL

To SCE, the amount of effort factors should be determined based on the type of project in terms of degree, i.e., small, medium, and large. The effort aspects are highly crucial, and their quantity will determine the project's success. Software projects necessitate the allocation of people, hardware, and software resources in production and development teams, contingent on the quantity of program code lines. If the amount of effort factors is too large, as a result, there is an incorrect estimate and the possibility of project failure due to high cost. If the amount of effort factors is too low, as a result, the project is not properly budgeted and it faces a lack of funds and is left half-finished. Therefore, the optimal value for them should be found. The best model for this is the use of metaheuristic algorithms. This algorithm can detect the optimal solution in difficult and sensitive conditions.

For some software projects, similar models and expert opinions are used, which is why the methods have two basic flaws. First, a lot of time may have been spent on the previous project. Second, more modern tools such as advanced programming languages are used in new projects. In today's programming languages, far fewer lines of code are required for software projects. In this article, the SCE is determined using a hybrid model utilizing the IWO and TS algorithms. NASA60, NASA63, NASA93, KEMERER, and MAXWELL datasets are used in the proposed model. The proposed model's flowchart is displayed in Fig. 1.

The starting population in the proposed model is generated depending on the values of the effort components. To find the optimal value for the effort factors, the length of the vectors should be determined according to the software data set. For NASA60, we consider the length of vectors equal to 15 because it has 15 effort factors. We calculate the values found in the problem space in each vector and select the vector that has a minimum based on the average calculation. Other vectors for the next steps as optimal test points and their values are optimized by using different operators of the proposed model.

TABLE 1: Review of the models presented for SCE

References	Model	Data set	Function
[10]	LR	ISBSG	MRE, PRED (25)
[11]	PSO-ANN	COCOMO81, NASA93, and MAXWELL	PRED (0.25)
[12]	ABC-ANN	COCOMO81, NASA93, and COCOMO_SDR	MRE
[13]	Chi-means- PSO	COCOMO81	VAF, MARE, and VARE
[14]	OCFWFLANN	NASA93	MMRE, MDMRE, and PRED (0.25)
[27]	OCFWANN	NASA93	MMRE, MDMRE, RED (0.25)
[15]	MPSO	NASA60, NASA63, and NASA93	MMRE
[16]	Tabu-Genetic algorithm	NASA60, NASA63, and NASA93	MMRE
[17]	Harmony Search Algorithm	NASA93	MMRE
[20]	GA-Bayesian Network	NASA63	MMRE
	PSO-Bayesian Network	NASA63	MMRE
[18]	PSO-Fuzzy clustering	NASA60	MRE, MMRE
	PSO-Learning automata	NASA60	MRE, MMRE
[19]	Linear regression	NASA63	MRE
	Multilayer perceptron ANN	NASA63	MRE
	SVM	NASA63	MRE
	KNN	NASA63	MRE
[21]	Multilayer perceptron ANN	NASA63	MRE
[22]	SVM-Genetic Algorithm	NASA, COCOM, and KEMERER	PRED
	Multilayer Perceptron ANN-Genetic Algorithm	NASA, COCOM, and KEMERER	MRE
[23]	Fuzzy Logic	NASA93	MRE
[24]	Genetic algorithm and Radius Function	COCOMO 81	MMRE, MSE
[25]	Genetic Algorithm-Fuzzy Logic	NASA93	MMRE, MDMRE, and PRED (0.25)
[26]	Multilayer ANN	COCOMO	MARE, PRED (0.25)

ANN: Artificial neural network, SCE: Software cost estimation, PSO: Particle swarm optimization, SVM: Support vector machine

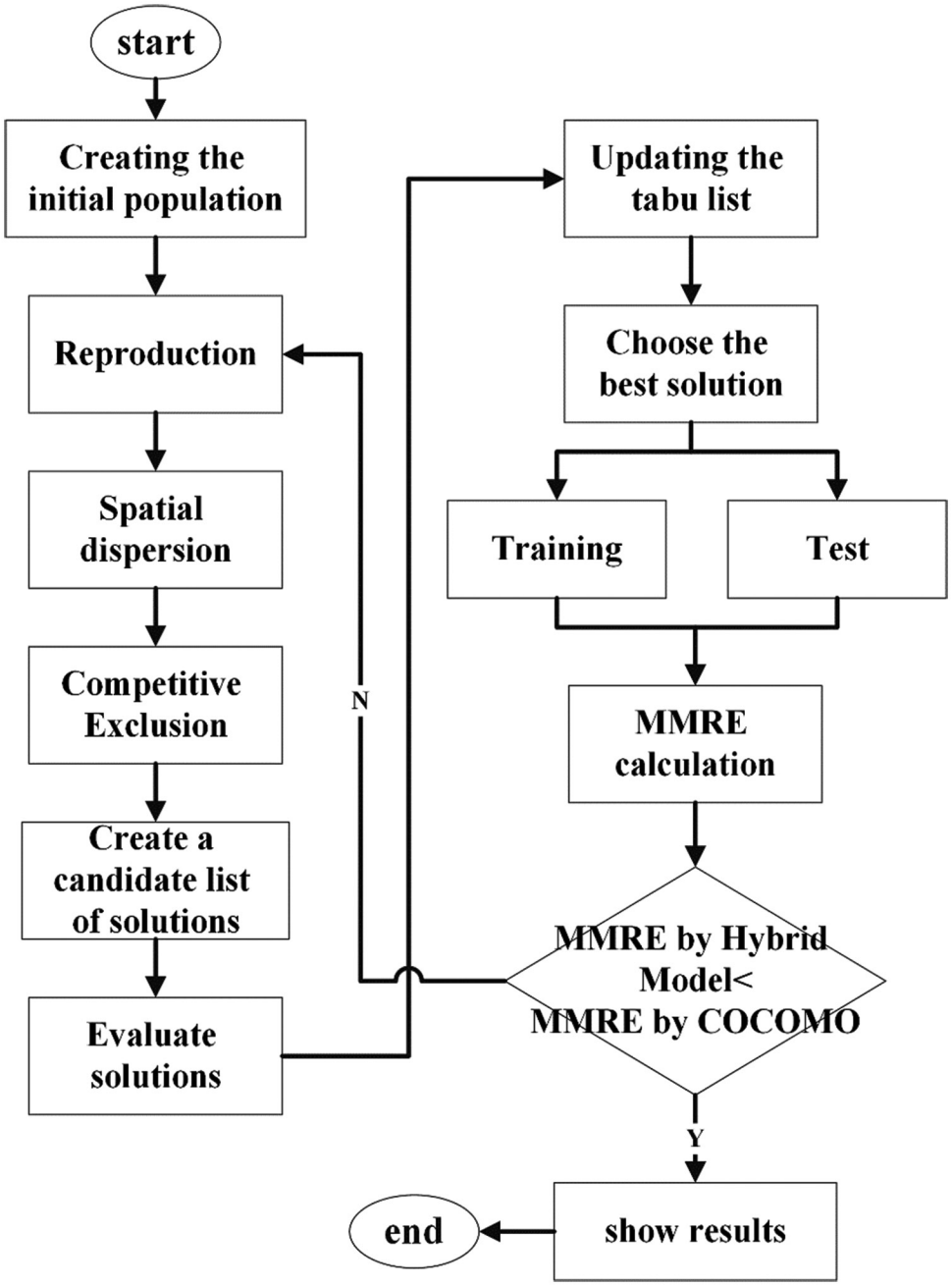

Fig. 1. Flowchart of the proposed model.

3.1. Initial Population

We create the initial population for the suggested model in the range [0.9, 1.4]. To create diversity in the initial population, optimization operations must be performed. Using the reproduction operator, we recreate the appropriate values for the initial population and then spread them over the problem environment. The new value for each effort factor is calculated according to Eq. (8). In this equation, S_{max} and S_{min} are produced in the interval [0.9, 1.4], respectively.

$$EM_i^k = S_{max} - \left(S_{max} - S_{min}\right)\frac{f_i^k - f_{min}^k}{f_{max}^k - f_{min}^k} \quad (8)$$

In Eq. (8), the parameter f_{min}^k is the minimum colony in iteration k and f_{min}^k is the maximum colony in iteration k.

And, the parameter f_i^k is the main value of the effort factor. In each iteration, effort factors are generated in the colony, and then, the lowest and highest values are selected in each colony.

In this space, we save the optimal points close to the values of the effort factors and remove inappropriate values. Then, to optimize the solution vectors created by the IWO algorithm, we transfer them to the TS algorithm. In the TS algorithm, a list of candidate solutions is created and we choose the best solutions from among them. Solutions are chosen based on how close the values of the effort factors are to each other. The selection of solutions is done based on the proximity to the values of the effort factors. Then, we put the vectors that make the solution of the problem non- optimal in the forbidden list. With this method, optimal points can be reached at a better speed, and only the solutions that make the MMRE more optimal than the COCOMO model participate in the solution. Then, the data sets for the test and training phase are read the new values are inserted into the effort factors, and the MMRE is calculated. A fitness function is represented by the average relative error measurement. If it has a value lower than COCOMO, the result will be shown. Otherwise, the program is repeated and more optimal values are found.

3.2. Spatial Dispersion

The produced values are now dispersed at random throughout the proposed model's multidimensional space. In each step, the defined initial value ($\sigma_{initial}$) is reduced to a final value (σ_{final}). The initial value and the final value are within the range of effort factors. The number of points in the spatial distribution is defined according to Eq. (9). The goal is to guide people to the local optimal points and achieve the optimal value.

$$\sigma^k = \frac{\left(I_{max} - k\right)^n}{\left(I_{max} - 1\right)^n}\left(\sigma_{initial} - \sigma_{final}\right) + \sigma_{final} \quad (9)$$

In Eq. (9), The σ^k is the value of the standard deviation, I_{max} is the maximum number of iterations, and n is the amount of non-linearity, which is randomly generated in the range (0-1).

3.3. Update Vectors

The vectors are updated according to Eq. (10). Each vector is obtained to determine the optimal value for the effort factors based on the next value (x^{i+1}) and the predicted value ($\hat{x}^i$). The range of all factors in the standard range is determined by Eq. (10). Therefore, each person only explores the optimal space in the proposed model.

$$x = \left| \frac{\left(x^{i+1}\right) - \left(\hat{x}^i\right) - \left(\left(x^{i+1}\right) - \left(\hat{x}^i\right)\right)}{\left(x^{i+1}\right) - \left(\hat{x}^i\right)} \right| \quad (10)$$

The solution vectors in the hybrid model contain the values of the software projects. These vectors are transformed into the optimal state based on IWO and TS operators, and their fitness is measured with each update step. If the MMRE error value is reduced, then the current solution vector is selected as the best solution and replaces the previous value of the effort factors. The combined model removes the worst solutions using the TS algorithm and places them in the forbidden list. Therefore, the agents in the search space are able to search the problem space only by following the receiving agents. As a result, the hybrid model escapes from getting stuck in the local optimum and can find the best effort values based on the size of the projects.

3.4. Evaluation Criteria

The MMRE [8] is considered a fitness function in the proposed model. The proposed model's fitness function aims to reduce relative error values when compared to the COCOMO model. Eq. (11) serves as the definition of the fitness function for the proposed model. The y parameter is equal to the real value and the $\bar{y}$ parameter is equal to the estimated value obtained by the proposed model.

$$\text{Mean Magnitude of Relative Error (MMRE)} = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{\left|y_i - \bar{y}_i\right|}{y_i} \times 100\right) \quad (11)$$

$$\text{Mean of Magnitude Error Relative(MMER)} = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{\left|y_i - \bar{y}_i\right|}{\bar{y}_i} \times 100\right) \quad (12)$$

$$\text{Mean Squared Error (MSE)} = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \bar{y}_i\right)^2 \quad (13)$$

$$\text{Root Mean Square Error (RMSE)} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(y_i - \bar{y}_i\right)^2} \quad (14)$$

$$\text{Mean Absolute Percentage Error (MAPE)} = \sum_{i=1}^{n}\left(\frac{\left|y_i - \bar{y}_i\right|}{y_i}\right) / n \times 100 \quad (15)$$

$$\text{Mean of Absolute Errors (MAE)} = \frac{1}{n}\sum_{i=1}^{n}\left|y_i - \bar{y}_i\right| \quad (16)$$

Median Magnitude of Relative Error (MdMRE)=Median (MRE) (17)

The total error derived from the effort components may be calculated using Eq. (11). After optimization, the proposed model's effort components are added to the COCOMO model. The relative error value and MMRE values are calculated. A model with a lower relative error value performs better when assessing estimate criteria than a model with a greater relative error value. When compared to models with greater MMRE values, the model with a lower MMRE value performs better.

4. RESULTS AND EVALUATION

In the C# programming environment, the proposed model's performance was assessed, and the results were obtained using NASA60, NASA63, NASA93, KEMERER, and, MAXWELL data sets. There are 50 solutions in the starting population, and there are 200 iterations. The outcomes of the proposed model on various data sets are displayed in Table 2. As you can see in Table 2, the proposed model has a lower mean relative error compared to the COCOMO models, the IWO algorithm, and the TS algorithm. The proposed model cannot get stuck in local optima and finds the appropriate value for the effort factors based on the operators of the IWO algorithm and the TS algorithm (Fig. 2). The value of MDMRE on NASA60 for IWO, TS, and the proposed model is 25.89, 28.32, and 28.06, respectively.

The proposed model demonstrates superior performance with the lowest MDMRE, signifying more accurate predictions compared to IWO and TS on the NASA60 dataset. The proposed model has achieved less error due to strengthening the solutions. The value of MDMRE on NASA63 for IWO, TS, and the proposed model is 37.11, 34.23, and 33.94, respectively. The proposed model outperforms both IWO and TS by achieving the lowest MDMRE, indicating enhanced accuracy on the NASA63 dataset. The value of MDMRE on NASA93 for IWO, TS, and the proposed model is 37.85, 46.54, and 45.21, respectively. The proposed model continues to showcase improved accuracy compared to IWO and TS, as reflected in its lower MDMRE on the NASA93 dataset. The value of MDMRE on KEMERER for IWO, TS, and the proposed model is 71.75, 43.06, and 42.22, respectively. The value of MDMRE on MAXWELL for IWO, TS, and the proposed model is 46.58, 41.97, and 45.62 respectively.

In Table 2, the results of the proposed model on NASA63 are shown. The proposed model has a lower MMRE value compared to the IWO algorithm and the TS algorithm. The IWO algorithm also performs better than the TS algorithm in terms of mean relative error, mean square error, RMSE, and MAPE. The proposed model has a lower MMRE value in the NASA93 data set compared to COCOMO and the TS algorithm. The TS algorithm has a lower MMRE value compared to the IWO algorithm. In the KEMERER dataset, by comparing the proposed model with COCOMO, the proposed model has greatly reduced the MDMRE value and the MMRE value. Also, the TS algorithm has better accuracy compared to the IWO algorithm. In the MAXWELL data set, the results show that the MMRE in the proposed model is lower compared to other models. Also, the value of MAPE and MDMRE in the TS algorithm is higher compared to other models. The comparison graphs of the MMRE on the NASA60 and NASA63 datasets are displayed in Fig. 3. In comparison to existing models, the proposed model has a lower MMRE value. The MMRE value on the NASA60 dataset for IWO, TS, and the proposed model is 24.99, 23.86, and 21.93 respectively. The MMRE value on the NASA63 dataset for IWO, TS, and the proposed model is 24.01, 26.79, and 21.56 respectively.

In Fig. 4, the comparison chart of MMRE on NASA93 and KEMERER datasets is shown. In comparison to existing models, the proposed model has a lower MMRE value. The value of MMRE on the NASA93 dataset for IWO, TS, and the proposed model is 22.52, 42.79, and 38.64 respectively. The MMRE value on the KEMERER dataset for the IWO, TS, and proposed models is 112.58, 94.53, and 78.44, respectively.

The MMRE comparison graph for the MAXWELL dataset is displayed in Fig. 5. The value of MMRE on the MAXWELL dataset for IWO, TS, and the proposed model is 42.09, 33.79, and 33.25, respectively.

4.1. Iteration Based Evaluation

In Table 3, the comparison of models based on iteration is shown. Table 3 shows that the MMRE has reduced as

TABLE 2: Results of the 200 iterations of the proposed model for various data sets

Data sets	Models	MMRE	MMER	MSE	RMSE	MAPE	MAE	MDMRE
NASA60	COCOMO	29.64	40.18	31908.53	178.63	29.67	91.71	28.23
	IWO	24.99	39.46	31747.86	178.23	26.94	87.15	25.89
	TS	23.86	39.12	31669.41	177.98	26.97	86.85	28.32
	Proposed model	21.93	39.17	31257.49	176.86	26.92	86.42	28.06
NASA63	COCOMO	36.00	39.49	408448.58	639.10	102.55	210.43	37.51
	IWO	24.01	41.96	331076.69	575.39	74.01	163.46	37.11
	TS	26.79	16.95	331634.71	575.82	70.62	159.74	34.23
	Proposed model	21.56	36.51	339454.22	581.71	73.64	155.63	33.94
NASA93	COCOMO	58.50	49.05	4096.40	64.00	115.55	1137.84	48.14
	IWO	22.52	63.38	2970.93	29.87	72.43	644.46	37.85
	TS	42.79	57.98	15016.18	38.75	92.65	806.88	46.54
	Proposed model	38.64	51.19	3733.71	38.68	92.29	804.82	45.21
KEMERER	COCOMO	502.81	75.71	339382.84	1504.18	502.81	1024.08	415.09
	IWO	112.58	159.42	12619.07	290.05	112.58	168.54	71.75
	TS	94.53	133.91	10508.51	264.68	94.53	148.22	43.06
	Proposed model	78.44	190.24	12549.27	289.24	78.44	146.66	42.22
MAXWELL	COCOMO	59.92	46.26	130153.29	3607.68	59.92	2981.01	13.92
	IWO	42.09	59.38	8660.82	1019.46	92.09	2355.89	46.58
	TS	33.79	58.73	66770.25	2583.73	83.79	1287.49	41.97
	Proposed model	33.25	57.68	15023.52	387.73	92.84	809.25	45.62

IWO: Invasive Weed Optimization, TS: Tabu Search

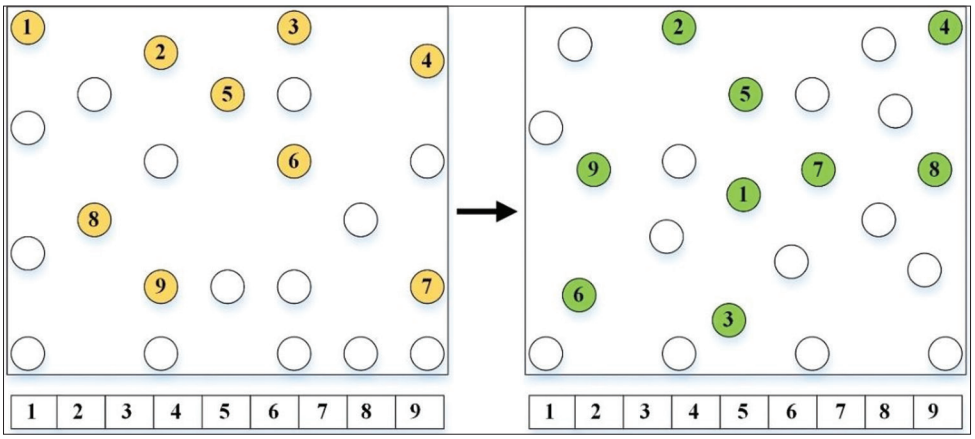

Fig. 2. Generation of optimal points for effort factors based on reproduction.

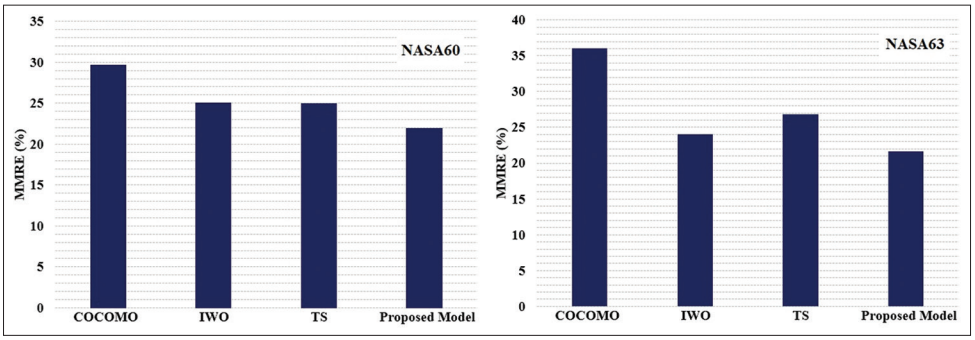

Fig. 3. MMRE comparison chart on NASA60 and NASA63 datasets.

the number of iterations has increased. The models search more points and finding the right value for the factors is more accurate. Advanced search capability allows the algorithm to not simply settle for a local optimal solution.

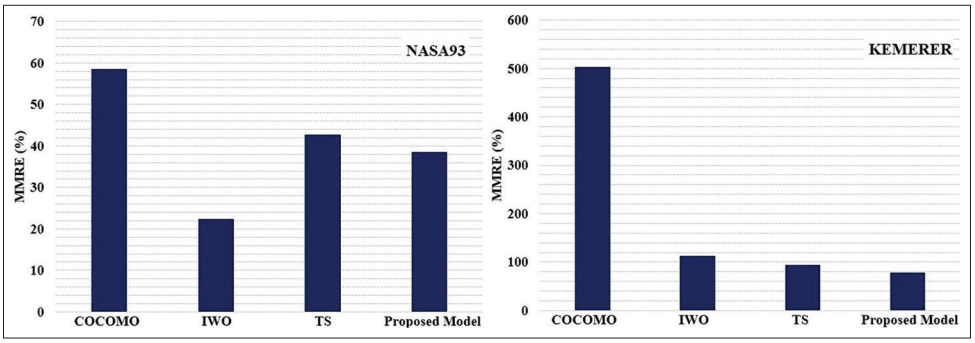

Fig. 4. Comparison chart of MMRE on NASA93 and KEMERER dataset.

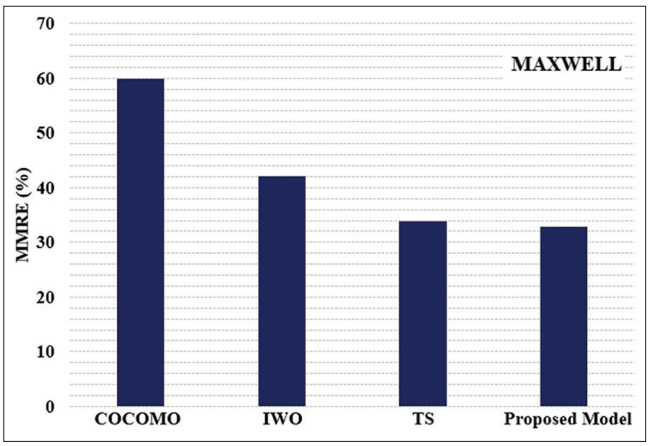

Fig. 5. MMRE comparison chart on MAXWELL dataset.

Rather, it continues to dig deeper to discover different potential solutions. By evaluating a larger number of points, the chances of finding the most appropriate and optimal solutions increase. The hybrid model is a powerful instrument for complex optimization problems as a result.

Investigations revealed that the proposed model's and the LSTM-RNN model's [29] MMRE values on NASA93 were 28.75 and 18.82, respectively. In addition, it was determined that the proposed model and the LSTM-RNN model had MMRE on MAXWELL values of 22,46 and 23,65, respectively.

4.2. Comparison and Evaluation

This section shows the comparison of the proposed model with other models. Table 4 shows that the proposed model performs better compared to other models. The percentage of training and test stages is 80 and 20, respectively. The proposed model has population diversity and avoids local optimal points. It can distinguish similar samples based on the distance index and update the banned list during the training process. The PSO-FLANN model is a technique that uses the global optimization capabilities of PSO to fine-tune the extended functional neural network and allows it to learn complex patterns with a relatively simpler structure. The hybrid of GA and FLANN can lead to a more consistent and robust model, especially when the data have intricate and non-linear interactions. The most important advantage of the proposed model is that its computing time is less compared to models like PSO-FLANN. Also, the number of parameters of the proposed model is less compared to other models. As a result, the implementation and efficiency of the proposed model are simpler.

According to the comparative results, the proposed model has a lower error value than other models. When compared to ANN models and decision trees like CART, the proposed model's error percentage is often smaller. ANNs often require significant data for training. And sometimes, they can be prone to overfitting. In contrast, the heuristic nature of the proposed model allows it to explore the solution space more broadly and possibly converge to an optimal solution with a lower error rate. The proposed model addresses the challenge of software projects by taking advantage of the strengths of TS and IWO algorithms. The proposed model is able to generate new solutions in each iteration and can find the best values for the effort factors. Software projects are getting wider and bigger every day, and the best tool to reduce cost and error is to use meta-heuristic algorithms.

5. CONCLUSION AND FUTURE WORKS

SCE includes resources that can only be learned and implemented with practical experience. The final goal of the estimate is to be as close as possible to the realities of the project. The time needed for each stage of the project must first be estimated before determining the entire project duration can be done. In this article, the improvement of

TABLE 3: Comparison of models based on the number of iterations

Iteration	Models	MMRE				
		NASA60	NASA63	NASA93	KEMERER	MAXWELL
100	IWO	24.99	24.01	22.43	112.58	42.09
	TS	24.96	26.82	42.79	94.53	33.79
	Proposed model	21.93	21.63	38.48	78.44	32.84
200	IWO	23.14	22.48	21.06	105.56	38.15
	TS	21.65	24.80	39.74	90.52	31.49
	Proposed model	18.35	19.81	34.96	62.76	26.31
500	IWO	21.03	20.43	19.86	98.65	35.79
	TS	19.31	21.17	35.24	82.11	24.82
	Proposed model	15.43	17.05	28.75	58.34	22.46

IWO: Invasive Weed Optimization, TS: Tabu Search

TABLE 4: Comparison of the proposed model with other models

Datasets	Models	MMRE		MDMRE		PRED (0.25)	
		Train	Test	Train	Test	Train	Test
NASA60	ANN [30]	0.44	0.44	-	-	-	-
	Proposed model	0.22	0.25	0.27	0.29	0.60	0.64
NASA63	PSO-FLANN [11]	0.43	0.37	0.48	0.42	0.39	0.52
	FLANN [11]	0.45	0.38	0.49	0.47	0.35	0.49
	SWR [11]	0.34	0.35	0.42	0.44	0.52	0.50
	Proposed model	0.29	0.31	0.35	0.40	0.32	0.58
NASA93	PSO-FLANN [11]	0.49	0.34	0.44	0.45	0.39	0.50
	FLANN [11]	0.42	0.49	0.46	0.48	0.38	0.48
	SWR [11]	0.39	0.34	0.47	0.49	0.44	0.44
	OCFWFLANN [14]	0.28	0.27	0.24	0.19	0.31	0.26
	OFWFLANN [14]	0.38	0.33	0.39	0.28	0.30	0.38
	FLANN [14]	0.43	0.37	0.37	0.33	0.46	0.39
	SWR [14]	0.92	0.79	0.58	0.44	0.45	0.41
	CART [14]	0.85	0.64	0.48	0.37	0.34	0.30
	OCFWANN [27]	0.33	0.32	0.29	0.24	0.36	0.31
	OFWANN [27]	0.43	0.38	0.44	0.33	0.35	0.43
	ANN [27]	0.48	0.42	0.42	0.38	0.51	0.44
	Proposed model	0.32	0.28	0.34	0.21	0.26	0.45
MAXWELL	PSO-FLANN [11]	0.55	0.38	0.49	0.42	0.32	0.48
	FLANN [11]	0.48	0.42	0.39	0.40	0.45	0.28
	SWR [11]	0.42	0.42	0.47	0.39	0.39	0.45
	LEMABE [31]	0.47	0.48	-	-	-	-
	Proposed model	0.38	0.36	0.34	0.36	0.28	0.56

IWO: Invasive Weed Optimization, TS: Tabu Search

the IWO algorithm based on TS was used for SCE. The purpose of the TS algorithm was to optimize the effort factors and find the appropriate value for them based on the determined interval. The MMRE value for the hybrid model on NASA60, NASA63, and NASA93 is 21.93, 21.56, and 38.64, respectively. The results showed that the TS algorithm was able to improve the IWO algorithm. Also, the MMRE value for the hybrid model on KEMERER and MAXWELL was obtained as 78.44 and 33.25, respectively. According to the findings, the proposed model has a lower MMRE than both the IWO and the TS algorithms. The TS algorithm's effectiveness was superior to the IWO method in several cases. In addition, the proposed model's MMRE was lower than that of COCOMO and other models. Project processes and stages should be prioritized and the dependencies between them should be carefully identified and all this information should be recorded in the project schedule. The dependence of different stages of the project as well as the amount of access to resources can have a significant impact on the time of the project.

The main limitations of this paper are that the proposed model has not been evaluated on the projects of an organization or a software company. Software companies

use different factors such as database programmers and computational programmers in software projects. Also, today's projects use different languages in coding, and the coding cost of each programming language is different. The second limitation is that meta-heuristic algorithms may use a specific procedure such as exploration and exploitation to find solutions, therefore, in these algorithms, there are no combined procedures such as increasing the number of layers such as ANNs.

For the work, we intend to use a combination of meta-heuristic algorithms and deep learning algorithms. Deep learning methods need to strengthen the parameters and the number of optimal layers, and these deficiencies are solved using meta-heuristic algorithms. Deep learning models, particularly neural networks, necessitate a meticulous tuning of parameters such as learning rates, regularization terms, and activation functions. Thus, determining the optimal number of layers and their configurations is a critical aspect, as an inadequate or excessive number of layers can impede the model's learning capacity or induce overfitting. To address these challenges, we propose to employ meta-heuristic algorithms. These algorithms, inspired by natural processes or optimization strategies, excel in exploring vast solution spaces and identifying optimal configurations. By leveraging the strengths of deep learning in capturing complex patterns and the optimization capabilities of meta-heuristic algorithms, we aim to develop a robust framework that excels in addressing the intricacies of diverse datasets and complex problem domains.

REFERENCES

[1] S. K. Gouda and A. K. Mehta. "Software cost estimation model based on fuzzy C-means and improved self-adaptive differential evolution algorithm". *International Journal of Information Technology*, vol. 14, no. 4, p. 2171-2182, 2022.

[2] S. K. Gouda and A. K. Mehta. "A self-adaptive differential evolution using a new adaption based operator for software cost estimation". *Journal of the Institution of Engineers* (*India*)*: Series B*, vol. 104, no. 1, pp. 23-42, 2023.

[3] W. Rhmann, B. Pandey and G. A. Ansari. "Software effort estimation using ensemble of hybrid search-based algorithms based on metaheuristic algorithms". *Innovations in Systems and Software Engineering*, vol. 18, pp. 1-11, 2021.

[4] Z. A. Dizaji and F. S. Gharehchopogh. "A hybrid of ant colony optimization and chaos optimization algorithms approach for software cost estimation". *Indian Journal of Science and Technology*, vol. 8, no. 2, p. 128, 2015.

[5] R. Mehrabian and C. Lucas. "A novel numerical optimization algorithm inspired from weed colonization". *Ecological Informatics*, vol. 1, no. 4, pp. 355-366, 2006.

[6] B. Boehm, C. Abts and S. Chulani. "Software development cost estimation approaches-a survey". *Annals of Software Engineering*, vol. 10, no. 1-4, pp. 177-205, 2000.

[7] O. Adalier, A. Ugur, S. Korukoglu, K. Ertas, H. Yin, P. Tino, E. Corchado, W. Byrne and X. Yao. *"A New Regression Based Software Cost Estimation Model Using Power Values*". *In: Proceedings of Intelligent Data Engineering and Automated Learning-IDEAL 2007: 8th International* Conference, *Birmingham, UK"*. Springer, Germany, 2007.

[8] T. R. Benala, K. Chinnababu, R. Mall and S. Dehuri. "*A Particle Swarm Optimized Functional Link Artificial Neural Network (PSO- FLANN) in Software Cost Estimation. In: Proceedings of the International Conference on Frontiers of Intelligent Computing: Theory and Applications* (*FICTA*)". Springer, Germany, 2013.

[9] Z. H. Wani and S. Quadri. "*Artificial Bee Colony-Trained Functional Link Artificial Neural Network Model for Software Cost Estimation. In: Proceedings of Fifth International Conference on Soft Computing for Problem Solving: SocProS 2015*". vol. 2, Springer, Germany, 2016.

[10] T. S. Sethi, C. V. Hari, B. T. Kaushal and S. Sharma. "*Cluster Analysis and PSO for Software Cost Estimation*. In: *Proceedings of Information Technology and Mobile Communication*. Springer, Germany, 2011.

[11] T. R. Benala, S. Dehuri, S. C. Satapathy and S. Madhurakshara. "*Genetic Algorithm for Optimizing Functional Link Artificial Neural Network based Software Cost Estimation. In: Proceedings of the International Conference on Information Systems Design and Intelligent Applications 2012* (*INDIA 2012*) *held in Visakhapatnam, India*". Springer, Germany, 2012.

[12] G. S. Rao, C. V. P. Krishna and K. R. Rao. "*Multi Objective Particle Swarm Optimization for Software Cost Estimation. In: ICT and Critical Infrastructure: Proceedings of the 48th Annual Convention of Computer Society of India*". vol. 1. Springer International Publishing, Cham, 2014.

[13] F. S. Gharehchopogh, R. Rezaii and B. Arasteh. "*A New Approach by Using Tabu Search and Genetic algorithms in Software Cost Estimation. In 2015 9th International Conference on Application of Information and Communication Technologies (AICT)*". IEEE, United States, 2015.

[14] S. S. Jafari and F. Ziaaddini. "*Optimization of Software Cost Estimation Using Harmony Search Algorithm. In: 2016 1st Conference on Swarm Intelligence and Evolutionary Computation (CSIEC*)". IEEE, United States, 2016.

[15] F. S. Gharehchopogh, L. Ebrahimi, I. Maleki and S. J. Gourabi. "A novel PSO based approach with hybrid of fuzzy C-means and learning automata in software cost estimation". *Indian Journal of Science and Technology*, vol. 7, no. 6, p. 795, 2014.

[16] Z. A. Khalifelu and F. S. Gharehchopogh. "Comparison and evaluation of data mining techniques with algorithmic models in software cost estimation". *Procedia Technology*, vol. 1, pp. 65-71, 2012.

[17] F. S. Gharehchopogh. "*Neural Networks Application in Software Cost Estimation: A Case Study. In: 2011 International Symposium on Innovations in Intelligent Systems and Applications*, *Istanbul, Turkey"*. IEEE, United States.

[18] L. Oliveira, P. L. Braga, R. M. F. Lima and M. L. Cornélio. "GA-based method for feature selection and parameters optimization for machine learning regression applied to software effort estimation". *Information and Software Technology*, vol. 52, no. 11, pp. 1155-1166, 2010.

[19] Malik, V. Pandey and A. Kaushik. "An analysis of fuzzy approaches for COCOMO II". *International Journal of Intelligent Systems and*

Applications, vol. 5, no. 5, p. 68, 2013.

[20] M. Molani, A. Ghaffari and A. Jafarian. "A new approach to software project cost estimation using a hybrid model of radial basis function neural network and genetic algorithm". *Indian Journal of Science and Technology*, vol. 7, no. 6, pp. 838-843, 2014.

[21] A. Hamdy. "Genetic fuzzy system for enhancing software estimation models". *International Journal of Modeling and Optimization*, vol. 4, no. 3, p. 227, 2014.

[22] Attarzadeh and S. H. Ow. "*Proposing A New Software Cost Estimation Model Based on Artificial Neural Networks. In: 2010 2nd International Conference on Computer Engineering and* Technology". IEEE, United States, 2010.

[23] T. R. Benala, S. Dehuri, S. Satapathy and C. S. Raghavi. "*Genetic Algorithm for Optimizing Neural Network Based Software Cost Estimation. In: Swarm, Evolutionary, and Memetic Computing*. Springer, Berlin, Heidelberg, 2011.

[24] G. S. Rao, C. V. P. Krishna and K. R. Rao. "*Multi Objective Particle Swarm Optimization for Software Cost Estimation. In: ICT and Critical Infrastructure: Proceedings of the 48th Annual Convention of Computer Society of India*". vol. 1. Springer International Publishing, Cham, 2014.

[25] Kaushik, N. Choudhary and P. Srivastava. "*Software Cost Estimation Using LSTM-RNN. In: Proceedings of International Conference on Artificial Intelligence and Applications: ICAIA 2020*". Springer, Germany, 2021.

[26] N. Rankovic, D. Rankovic, M. Ivanovic and L. Lazic. "Improved effort and cost estimation model using artificial neural networks and Taguchi method with different activation functions". *Entropy* (*Basel*), vol. 23, no. 7, p. 851, 2021.

[27] M. Dashti, T. J. Gandomani, D. H. Adeh, H. Zulzalil and A. B. Sultan. "LEMABE: A novel framework to improve analogy-based software cost estimation using learnable evolution model". *PeerJ - Computer Science*, vol. 8, p. e800, 2022.