iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 12 **Issue:** III **Month of publication:** March 2024

DOI: https://doi.org/10.22214/ijraset.2024.59309

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue III Mar 2024- Available at www.ijraset.com

Optimizing Software Effort Estimation Models Using Enhancement of Firefly Algorithm

B. Prabhanjali[1], G. Kavya[2], K. Suresh[3], V. Venkataiah[4]
[1, 2, 3]*UG Students,* [4]*Associate Professor, Dept. of Computer Science & Engineering, College Of Engineering & Technology, Hyderabad, Telangana*

Abstract: ***Estimating the amount of work required in order to create software development is regarded as essential to the software development life cycle and to the control of project costs, schedules, and quality. As a result, precise estimation plays a critical role in project success and risk mitigation. Software effort estimation has drawn a lot of interest from scholars recently and presented a problem to the software business. Three COCOMO-based models' parameters are to be optimized using a metaheuristic method called improvement of Firefly Algorithm, which is proposed in this research. The basic COCOMO model and the other two models that have suggested in literature as expansions of basic COCOMO model are among these models. Root Mean Square Error (RMSE) is one of the evaluation measures used to assess the created estimate models.***
Index Terms: Firefly Algorithm, Metaheuristic Optimization, Effort Estimation, and Software Quality.

I. INTRODUCTION

Software time and effort development of estimation models are fire topic of the study over the three decades for the engineering of software community with systematic approach to the development, maintenance and retirement of software[1]. Computer program taken a toll estimation alludes to the expectations of the probable sum of effort, time, and the number of employees required to construct a program. The essential work of the venture advancement is to guarantee that the venture is completed with the objective: "high quality of computer program must be created with a moo taken a toll that's inside time and budget"[2]. The duties of venture improvement are arranging, organizing, staffing, coordinating, and controlling the exercises. Belittling computer program costs can have an inimical effect on the quality of conveyed program and hence a company's trade reputation and competitiveness. The primary vital component of computer program extends administration is compelling preparing of the improvement of the computer program which determines the assets required to total the extend effectively. The assets incorporate the number and ability level of the individuals, and the sum of computing resources[3]. The fetched of a program venture is directly relative to the individuals / assets required for the project[5]. Overestimation of computer program fetched, on other hand, can result in missed chances to use reserves in other referential ventures. The need for solid and accurate taken a toll expectations in program building is a continuous challenge[3]. Good estimation can improves performances of the company particularly in overseeing the venture plan, human asset assignment, fetched estimation, etc. Those points of interest can decrease the disappointment possibility or extend delay. The Fetched for a extend may be a function of numerous parameters. Measure may be an essential fetched figure in most models and can be measured utilizing code snippets (or) thousands of conveyed KDLOC or work focuses. The no. of models are advanced to set up the connection between Measure and Exertion for Computer program Exertion Estimation[6]. Computer program taken a toll estimation strategies can broadly classified as algorithmic and non-algorithmic models. The algorithms-based models come from the factual examination of historical venture data, for illustration, Helpful Fetched Show (COCOMO) and Program Life Cycle Administration (SLIM)No algorithmic methods incorporate Price-to-Win, Parkinson, master judgment, machine learning approaches[7]. Machine learning is utilized to group together set of methods that embody a few of the characteristics of human intellect, for the case, foggy frameworks, similarity, regression trees, run the show acceptance neural systems, and Developmental algorithms[3]. Among the machine learning approaches, foggy frameworks and neural systems, and Developmental calculations are regarded as have a place to the delicate computing group[11]. The algorithmic as well as the non-algorithmic (based on master judgment) taken a toll estimation models, be that as it may, are not without error. A few exertion estimation model have been develops and advanced over time superior forecast precision moreover, in this way for better improvement quality. The Such model range will be from complex calculations , statistical evaluation of the project's specifications to improve machine learning approaches[10]. Heuristic optimization is plan are relies on few attempt to find the best solutions. Heuristic optimizers are have used in the software cost estimation, such as to application of the genetic programming an model optimizations.

Another case is the portion that the PSO is taken as an optimizer using heuristics. Besides, the crossover methods include the combination of heuristic algorithms just like the utilization of Hereditary Algorithm , Ant Colony[12]. In spite of an outsized number tests on finding foremost satisfactory forecast show, there's not clear prove of an extremely precise and effective strategy. At same time, it is significant to create a forecasting strategy that's fewer intricate and very extra valuable. For occasion, in few forecasting models, an outsized number of factors that are utilized to build the show don't reflect or make strides the precision of prediction show. In this way, gathering additional or disconnected factors are laborious and lacking in centrality. It could become more productive on construct a demonstrate with a least number of the variables, hopefully finding foremost vital and the common factors bland project development efforts[10].

II. BACKGROUND WORK

This here section provides an overview of some recent research which talks about the field of effort estimation for those software projects. This is actually a very active area in research and there have been lot of papers that were published recently. In this specific section, we are going to highlight and sort of review some of very important studies that kind of applied some machine learning techniques in order estimate the effort that goes to the software projects. The dynamic area that involves effort estimation for software projects, well, let's just say that recent research have actually contributed significantly to much further development of methods, especially through the uses of some ML techniques. Miltveit et al. [1], Ganesan et al. [2], and Bhattacharjee et al. [3] sort of made a pretty significant contribution by exploring this case-based approach and then proposing this model that is actually based on those expert cases. The use ML optimization algorithms has sort of become a focus on the purpose of improving that accurate of the software effort estimation, with some techniques such as Cuckoo Search and PSO, Bat Algorithm, and Firefly Algorithm gaining some traction. [3].

Shin and Goel [6] actually questioned the common use of linear regression in empirical methods and then found actually a radial basis function (RBF) neural I suggested in using the network. His proven COCOMO II model, devised by Boehm in 1981, is actually still very influential in estimating effort and development time take into accounts parameter such as effort multipliers, scaling factors, and software size [8]. A.F.Sheta [7] kind of presents this innovative model that sort of combines genetic algorithms (GA) with his COCOMO and demonstrates improved accuracy in the cost estimation. Social behavior-inspired particle swarm optimization algorithms has been sort of proven successful in variety of optimization problems and have actually demonstrated The capability for rapidly converge [9]. Additionally, continuous training of Artificial Intelligence (AI) models on data positions them as a sort of superior alternative to algorithm models inside the software cost estimation, allowing optimization of critical elements. It can sort of minimize project manpower [13]. Selecting metrics for the sizing of those software projects, including: B. Lines of code, function points and cosmic function points (CFSUs) sort of playing an important role. Cosmic FFP is characterized by use of the functional size units as a practical and kind of proven solution for size estimation and quality improvement [10]. Kara Giannopoulos et al. [11] proposed five wrapper character selection method using regression algorithms, including forward selection (FS), backward selection (BS), best first forward selection (BFFS), and best first backward selection (BFBS). By comparing, we kind of contribute to the field. Genetic Search Selection (GS) whose performance was analyzed on 12 of his UCI datasets. In domain of software cost estimation, AI techniques display superior accuracy compared to algorithmic models and iteration optimized factors to reduce the money and effort associated with software projects. As the having complexity of optimization problems increase, metaheuristic algorithms are used of particular note are genetic algorithms & Ant colony optimization (ACO) that use population-based search and evaluation. It helps to really approach the problem with an iteratively until an optimized solution is obtained with high precision!

This comprehensive background work is focusing on different approaches and innovations in software money estimation, integrating varied methods and techniques of ML, optimization algorithms, AI models, and [3] used the hybrid approach to address in-context parameter selection and models optimization. It estimating software struggle and effort. This study is focused on use genetic algorithms (GA) to optimize support vector regression (SVR) models. The authors specifically have investigated and in the impact of GA on character selection and parameter optimization, compared with other approaches, he demonstrated its effectiveness in improving on performance of SVR models. This result is highlighting the applicability of GA to improve accuracy of man power estimation models. Furthermore, a general that framework for estimate the software man power was introduced in [3].

This innovative framework aimed is to emulate human thought processes by incorporating fuzzy rule models. The generated models leveraged the expertise to ensure interoperability and applicability to different been problem areas such as risk analysis and software quality prediction.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue III Mar 2024- Available at www.ijraset.com

III. EMPLOYED TECHNIQUE

A. *Regression Methods*

"The objective of regression methods is effectively maps a group of independent variables (X1, X2..., Xn) to the dependent variables Y; in our context, our sole goal is to construct regression models using a training dataset! Using these models will; result in predicting the total effort is required for the process of creating software projects that is measured in man-months! Extra points should be the considered for the achievement of our ultimate spelling bee success."

B. *Firefly Algorithm*

The Firefly Algorithm (FA) is a multimodal optimization algorithm that is inspired by the activity of fireflies and falls under the category of nature-inspired algorithms. At the University of Cambridge, Xin-She first introduced FA in 2007. It has been demonstrated through empirical evidence that FA approaches the problem more organically and could outperform other metaheuristic algorithms. FA is predicated on his three core principles. According to the first, fireflies of all genders are drawn to one another. According to the second rule, attraction is connected with luminescence or brightness, therefore brighter flies will draw in fewer brighter flies, and movement will be random in the absence of brighter flies. According to the final main rule, brightness varies with the goal function because the objective's landscape

Pseudo code of FA
Begin
1) Initialisation max iteration, α β_0, γ
2) Generate initial population
3) Define the Objective function f(x),
4) Determine Intensity(I) at cost (x) of each individual determined by $f(x_i)$
5) While (t < Iter max)
For i= 1 to n
For j = 1 to n
If ($I_j > I_i$)
Moves firefly i towards j in k dimension
End if
Evaluate new solutions and update light intensity End for j
End for i
Rank the fireflies and finds current best
End while
6) Post process result and visualization
End procedure

C. *Root Mean Square Error (RMSE)*

Root mean square error, often confused on root mean squared deviation, are not really the same thing! They differ in terms of how they calculate prediction quality. RMSE or RMSD (let's keep it fancy) is super widely used for evaluating predictions - it's like the ruler of measures! Like, it tells us how far predictions be sliding away from actual true values using some magical Euclidean distance trick.

$$RSME = \sqrt{(\sum((y_i - \hat{y}_i)^2)/n)} \quad \text{--(1)}$$

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue III Mar 2024- Available at www.ijraset.com

IV. PROPOSED SYSTEM

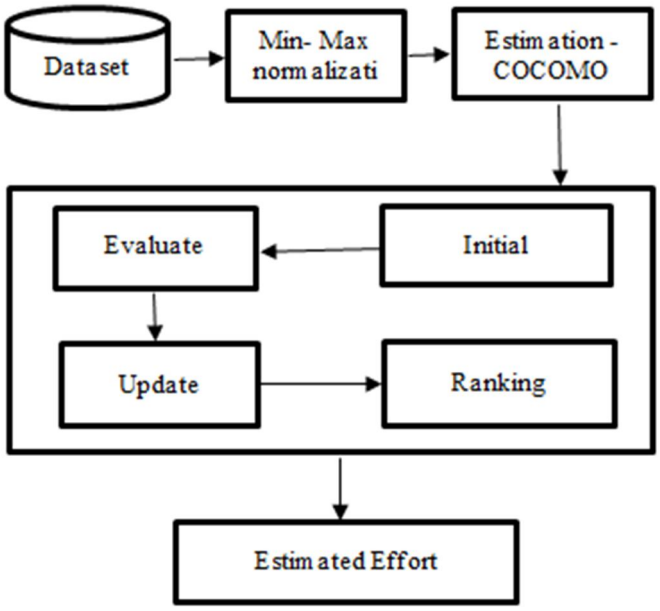


Fig 1 : Architecture of Firefly

Data set (COCOMO81): In this Paper we are taking dataset from the literature review (i.e COCOMO81 dadaset).

1) Preprocessing:

MIN-MAX Normalization: In this process the Preprocessing of the dataset to Normalize the values in between 0 and 1. This can improves performance of Firefly Algorithm by making the data more uniform.

$$X_{new} = X - X_{min} / X_{max} - X_{min} \quad \text{--(2)}$$

Estimation Model:

Cocomo model: The Constructive Cost Model (COCOMO), It primarily relies on the measurements of the software, measured in Delivered KLOC.

Basic COCOMO: This is the simplest model, suitable for organic mode projects with experienced staff and well-defined requirements. It uses the following formula:

*Effort = a * (KLOC^b) --(3)

2) Optimization Loop:

Firefly Algorithm: Each firefly iteratively adjusts its position in search space based on its brightness and the attractiveness of other fireflies. Brighter fireflies (those representing more accurate estimations based on evaluation criteria) attract other fireflies, moving the population towards better solutions.

Initialize Firefly Algorithm: Initializing firefly algorithm, which adjusts its position in the search apace based on brightness and attractiness of the firefly.

Evaluate: The fitness of each firefly (how well its solution matches the actual effort values) is calculated using a fitness function.

Update: The steps involve the representing of the weights of the every fireflies after calculating the fitness function.

Ranking: Ranking is the process arranging the fireflies with the least values to get optimal out of all firefly weight.

Optimal Result: The firefly with the best fitness value, representing the accurate effort estimation for new software project, is selected as the optimal result.

There were two main problems with the fly attraction in FA: attraction modeling and light intensity differences. For particular firefly at location X, the brighter I is formulated as I (X) a f(X). On the other hand of the attractive force ß is proportional to the fly and related by the distance Rij in between fireflies i & j. Inverse square of the light intensity I (r). Here, I0 represent the light intensity as the light source.

$I(r) = I0 - \gamma r2$ --(4)

Considering that the environment has as been absorption coefficient γ, intensity is perfectly represented in which I0 is the questionable original intensity.

$I(r) = I0 / 1 + \gamma\ r2$ --(5)

Typically, the distance among a certain firefly at one location (Xi) & another an enigmatic location is representes by the Euclidean distance. The perplexing kth component by the spatial coordinate Xi is represented by X j, in which Xi k

$Rij = \sqrt{(X1 - Y1)2 + (X2 - Y2)2}$ --(6)

A firefly drawn a brighter one, j, it is represented by the attraction as $\beta 0e\text{-}\gamma r^2 (Xj - xi)$, and the randomness it is represented by $\alpha (rand - ½)$, which is determined by the enigmatic randomization parameter α.

$Xnew = Xi + \beta 0e\text{-}\gamma r^2(Xj - Xi) + \alpha\delta\ (rand - ½)$ --(7)

Moreover, FA's silly movements and convergence speed are influenced by γ, which also explains variations in ugly attractiveness.

V. EXPERIMENTS AND RESULTS

This research says the COCOMO 81 project effective data set to maybe produce almost comparable results. The data set kind a challenges due to small no. of the instances & limited variables analyzed, almost. However, for the objective of this remarkably unique research, by the data set kind a sort considered kind a maybe adequate, possibly. The COCOMO 81 data set is, like, split in to two parts: a training set comprising 13 instances, which accounts for like about 60% of the data, and a testing set with 63 instances, making kind up about 30% of the total projects, I guess.
They are three main variable kind a sort considered in this research are Project Size will be in KLOC, Methodology and Actual Effort . The training data set includes instances 1 to 13, and instances 14 to 63 .

Table 1 – COCOMO 81 Dataset

KLOC	Measured Effort	Size(MIN_MAX)	Effort(MIN_MAX)
113	2040	0.096706	0.178522
293	1600	0.253497	0.139906
132	243	0.113256	0.020809
60	240	0.050539	0.020546
16	33	0.012212	0.002378
4	43	0.00176	0.003256
6.9	8	0.004286	0.000184
22	1075	0.017439	0.093829
30	423	0.024407	0.036661
29	321	0.023536	0.027655
32	218	0.026149	0.018615
37	201	0.030505	0.017123

25	79	0.020052	0.006416
3	60	0.000888	0.004748
3.9	61	0.001672	0.004836
6.1	40	0.003589	0.002993
3.6	9	0.001411	0.000272
320	11400	0.277016	1
1150	6600	1	0.578729
299	6400	0.232592	0.561173
252	2455	0.217784	0.219446
118	724	0.101061	0.063024
77	539	0.065347	0.046787
90	453	0.076671	0.03924
38	523	0.031376	0.045383
48	387	0.040086	0.033447
9.4	88	0.006463	0.007205
13	98	0.009599	0.008083
2.14	7.3	0.000139	0.000123
1.98	5.9	0	0
62	1063	0.052281	0.092776
390	702	0.337991	0.061093
42	605	0.03486	0.05258
23	230	0.01831	0.019668
12	82	0.008728	0.066789
15	55	0.011341	0.004309
60	47	0.050539	0.003607
15	12	0.011341	0.000535
6.2	8	0.003676	0.000184
3	8	0.000888	0.000184
5.3	6	0.002892	8.78E-06
45.5	45	0.037909	0.003432
28.6	83	0.023188	0.006767
30.6	87	0.02493	0.007117
35	106	0.028763	0.008785
73	126	0.061863	0.010541
23	36	0.01831	0.002642
464	1272	0.402443	0.111119
91	156	0.077542	0.013173
24	176	0.019181	0.014929
10	122	0.006959	0.010189
8.2	41	0.005418	0.003081
5.3	14	0.002892	0.000711
4.4	20	0.002108	0.001237
6.3	18	0.003763	0.001062
27	958	0.021734	0.083561

17	237	0.013083	0.020282
25	130	0.020052	0.010892
23	70	0.01831	0.005629
6.7	57	0.004111	0.004485
28	50	0.022665	0.00387
9.1	38	0.006202	0.002817
10	15	0.006986	0.000799

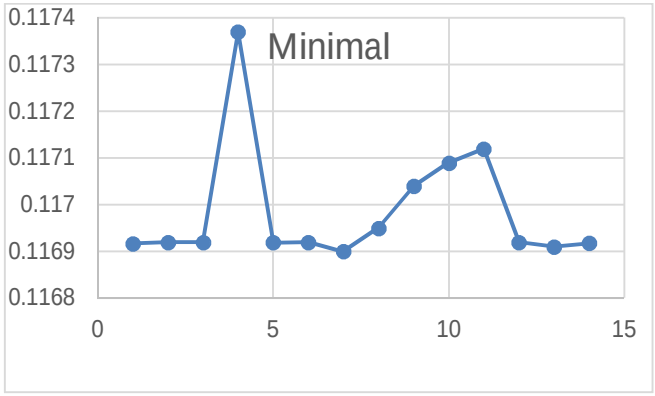
Firefly algorithm parameters settings:

Table 2 – Parameters of Firefly Algorithm

Parameters	values
Maximum iterations	1000
Numbers of Fireflies	63
Alpha	0.1
Betamin	1.0
Gamma	0.01
Theta	0.97

To scaling dada by using min-max normalization which send as input to proposed algorithm. Setting the parameters of Firefly algorithm for tuning the COCOMO81 model parameters for accurate SCE. Thorough The experiment was conducted with will be get the best estimated effort. To evaluate the performance of proposed technique by using RMSE.

FA, a metaheuristic optimization technique, outperforms COCOMO81-based software effort models in terms of estimation accuracy. The minimal value of this is 0.1167917



Graph 1 : Minimal value of RMSE

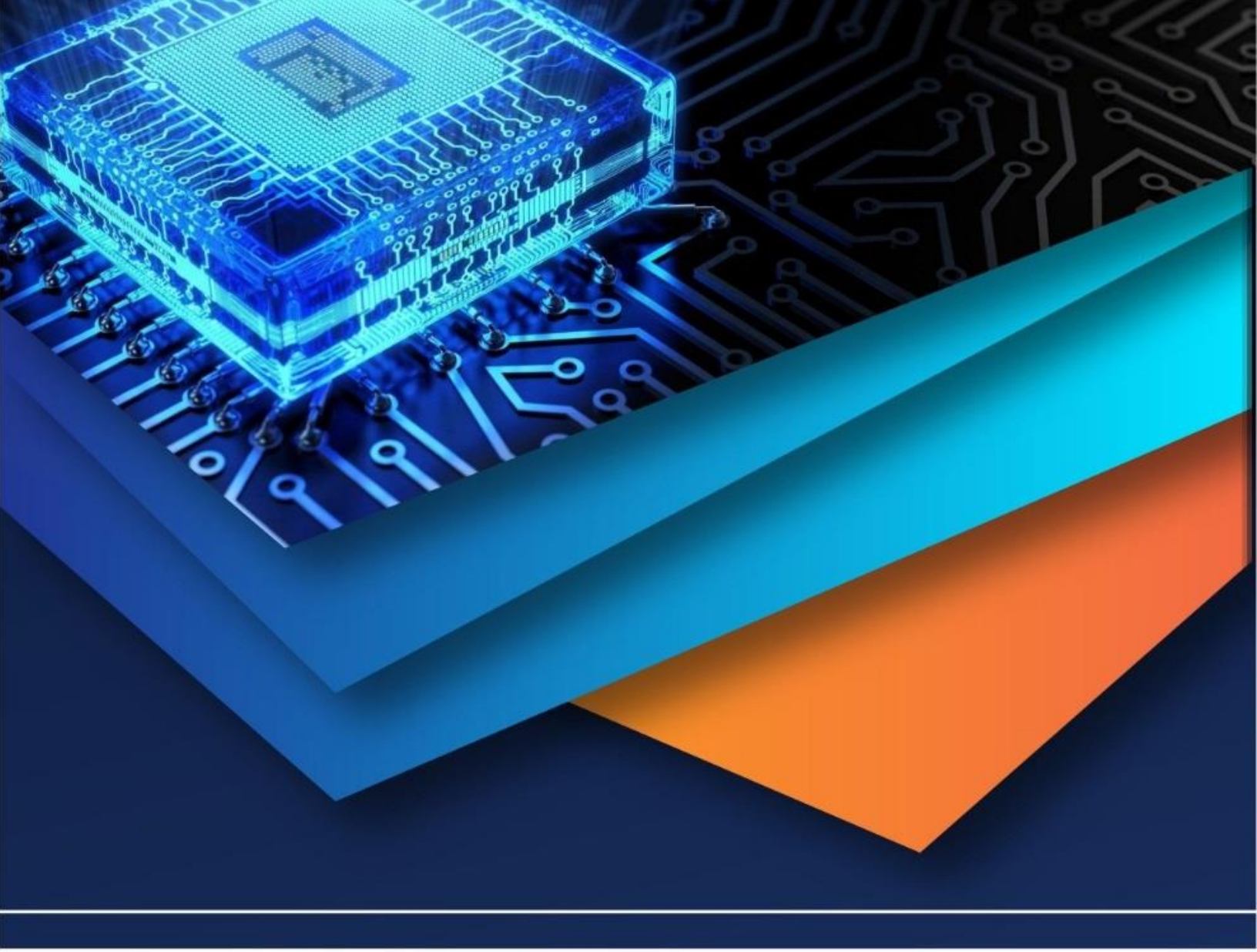
VI. CONCLUSION

The Firefly calculation seems potentially in program exertion estimation because of its capacity to optimize parameters and managing complex, non-linear connections within datasets. Its iterative nature permits for continuously advancements and adjustment, possibly leading to more accurate estimations over time. However, encouraging investigate and approvals are essential to fully evaluate its adequacy and comparing it with the existing estimation strategies.

REFERENCES

[1] Zareei, M. and Hassan-Pour, H.A., 2015. A multi- objective resource-constrained optimization of time-cost trade-off problems in scheduling project. Iranian Journal of Management Studies, 8(4).
[2] Ranichandra, S., 2020. Optimizing non-orthogonal space distance using ACO in software cost estimation. Mukt Shabd J, 9(4), pp.1592-1604
[3] Ghatasheh, N., Faris, H., Aljarah, 2019. Optimizing software cost estimation models using firefly algorithm. arXiv preprint arXiv:1903.02079.

[4] Ahadi, M. and Jafarian, A., 2016. A new hybrid for software effort estimation using particle swarm optimization and differential evolution algorithms. Informatics Engineering, an International Journal (IEIJ), 4(1).
[5] Dizaji, Z.A. and Khalilpour, K., 2014. APPARATUS BASED ON CHAOS THEORY AND PARTICLE SWARM OPTIMIZATION FOR SOFTWARE COST ESTIMATION. International Journal of Academic Research, 6(3).
[6] Braga, P.L., Oliveira, A.L. and Meira, S.R., 2007, September. estimating software effort with solid confidence intervals with machine learning algorithms. During the 7th International Conference on Hybrid Intelligent Systems (HIS 2007) (pp. 352-357). IEEE.
[7] Dizaji, Z.A., Ahmadi, R., Gholizadeh, H. and Gharehchopogh, F.S., 2014. A software cost estimating method using a bee colony optimization algorithm. International Journal of Computer Applications, 104(12).
[8] Puspaningrum, A. and Sarno, R., 2017 A software cost estimating technique combining harmony search and cuckoo optimization. Procedia Computer Science, 124, pp.461-469.
[9] Wu, D., Li, J. and Bao, C., 2018. Software effort estimation using case-based reasoning and optimum weight obtained from particle swarm optimization. Soft Computing,22, pp.5299-5310.
[10] Li, Y.F., Xie, M. and Goh, T.N., 2009. An analysis of feature weighting and project selection for analogy-based software cost estimating. Journal of systems and software, 82(2), pp.241-252.
[11] Harou, J.J., Pulido-Velazquez, M., Rosenberg, D.E., Medellín-Azuara, J., Lund, J.R. and Howitt, R.E., 2009. Hydro-economic models: Concepts, design, applications, and future prospects. Journal of Hydrology, 375(3-4), pp.627-643.
[12] Adamu, A., Abdullahi, M., Junaidu, S.B. and Hassan, I.H., 2021. A hybrid particle swarm optimization method for feature selection that combines the crow search technique. Machine Learning with Applications, 6, p.100108.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY