International Journal of Engineering Trends and Technology
ISSN: 2231 – 5381 /doi:10.14445/22315381/IJETT-V69I8P210
Volume 69 Issue 8, 81-88, August, 2021
©2021 Seventh Sense Research Group®

A Model to Measure Software Testing Effort Estimation in the Integrated Environment of ERNN, BMO & PSO

Bijendra Singh[#1], Dr. Ankit Kumar[*2], Dheeraj Kumar Sahni[#3], Divya Shree[#4], Anu[#5], Khushboo[#6], Kapil Sirohi[#7] Dhiraj Khurana[#8]

[1] Research Scholar, CSE, BMU, Rohtak
[2] Associate Prof, CSE, BMU, Rohtak
[3,4,5,8] Assistant Prof., CSE, UIET, MDU, Rohtak,
[6,7] Assistant Prof., ECE, UIET, MDU, Rohtak, HARYANA, INDIA

[1]Bijendra.singh046@gmail.com, [2]ankit524.in@gmail.com, [3]dheerajsahni.rp.uiet@mdurohtak.ac.in,

Abstract – *The Elman neural network is a recurrent neural network which, is playing a significant role in effort estimation during software testing. It is reliable, much efficient in optimizing the results using several inputs from hidden layers while training the network. On the other hand, Barnacles Mating Optimizer (BMO) is a well-known optimization mechanism that could be used to optimize the result produced by Elman neural network during effort estimation. The particle swam optimization is considered a computational method that is capable to optimizes the problem by trying to enhance the solution with respect to the specified measure of quality iteratively. In this research, Elman recurrent neural network (ERNN), Barnacles Mating Optimizer (BMO), and particle swam optimization (PSO) are integrated to propose a multi-objective model to test the application. The PSO is applied to get more reliable and optimized results considering Accuracy, Precision, F-Score, and recall value. The research concludes that the proposed work has shown improvement in reliability as compared to the existing neural network models.*

Keywords — *BMO, ERNN, Integrated Environment, Optimization, PSO.*

I. INTRODUCTION

Software Testing effort Estimation (STEE) has been performed by dividing the whole project into different subtasks. Now, subtasks are allocated to team members to perform the effort estimation for the subtasks. Finally, the estimation is validated. But in the present scenario, these operations are becoming very difficult in the case of computer programs which are complex. Calculation of effort in an optimal way is essential during the execution stage of the computer program. Lots of designs are already available for the effort estimation, but most of them are failing to provide the solution in an optimal way. In other words, the software testing effort estimation [9] has become a challenging operation. Thus, there is a need for the proposed mechanism that should be capable of providing more optimized results with good accuracy. The proposed research has integrated ERNN, BMO, and Particle swarm optimization (PSO) [1, 2, and 3] in order to provide a more accurate and optimized solution for software testing effort estimation (STEE). The reason behind the optimization is to find a good design relative to a group of prioritized criteria or constraints. These prioritized criteria are maximizing factors. Maximizing factors could be productivity, reliability, strength, efficiency as well as utilization.

Particle swarm optimization (PSO) [1, 2, 3, 4, and 5] is a well-known optimization mechanism that depends on the population. It is inspired by the movement of bird flocks as well as schooling fish. In this swarm and control parameters are initialized in the beginning, acceleration constants, initial velocities, the position of the particle, and personal best positions are specified in the context of the basic PSO. It considers the particles as potential solutions.

Elman neural network (ERNN) [7] manages multiple inputs from the hidden layer as compare to traditional neural networks. It is capable of proposing a reliable solution by use of multiple inputs at a hidden layer at the time of training. The presence of multiple hidden layers in the network model improves the reliability of the solution. In other words, utilization of hidden layer is responsible for the increment of accuracy in ENN as compare to traditional NN [6]. Moreover, batch size, epoch size, size of dataset plays a significant role during the training of the network model.

Barnacles Mating Optimizer (BMO) [14, 15, and 16] is a well-known optimization mechanism that could be used to optimize the result produced by Elman neural network during effort estimation. Barnacles Mating Optimizer is inherited from the mating procedure of barnacles that are found in nature. Barnacles have been considered microorganisms that have been considered as hermaphroditic. These are the organisms with characteristics of both male and female during reproductions by sex. Such microorganisms are usually fertilized by the neighbor to generate further off-springs. They have unique characteristics, which are huge penises which are greater in microorganisms as compare to their own body size. Barnacle's parents are chosen randomly to produce further off springs.

The whole research is divided into subsections in which; the first section is about the basic introduction of mechanism and algorithms. Section two is presenting the existing research, and section three presents the proposed model to resolve the issues in existing work. Section 4 is focusing the simulation of the proposed work and comparison of existing with proposed work. Section 5 is the conclusion where the actual finding of the result and final outcomes are presented.

II. LITERATURE

In the area of software testing effort estimation, several studies have been done with different optimization mechanisms like MODA, PSO, BMO, etc. [1, 2, 3, 4, 5]. However, several analyses are performing data rectification using recurrent (Elman) neural networks [6, 7, 8]. Some investigation has considered the mechanism like Genetic algorithm, ANN, SVM [10] for decision making and prediction. Many others also have done an analysis regarding [11, 12] the issues in the existing studies, and they were found a lack of accuracy and optimization in the results. In some investigations, authors have also compared backpropagation Neural with ERNN [13]. Barnacles Mating Optimizer has been introduced as a novel Bio-Inspired mechanism to resolve issues during optimization [14]. In some research, the Implementation of Barnacles Mating Optimizer with Evolutionary Algorithm has been made to solve optimization [15]. Some authors have presented the BMO application in order to resolve issues regarding economic dispatch [16]. Another Cost-Sensitive approach has been proposed in order to increase the utilization of classifiers supported by Machine learning [17] during efforts estimation in the testing phase. Some of the Research-based on Intelligent Network [18] has been performed for Auto Software Testing Technology.

III. PROPOSED MODEL

A model has been proposed by integrating ERNN, BMO, and PSO. Elman neural network has increased the reliability of results by using multiple inputs from the hidden layer at the time of training. It takes real input X1(t)…Xk(t) and context input H(t-1)…H(t-m) and process them on the hidden layer to produce output y(t) on the output layer. Figure 1 is presenting the working of the standard ERNN model.

In this work, BMO is acting as an optimization mechanism that supports Elman neural network during effort estimation. BMO and ERNN are combined for the determination of reliability. Barnacle's population size and arrangement of ERNN are initiated before the learning and testing phase.

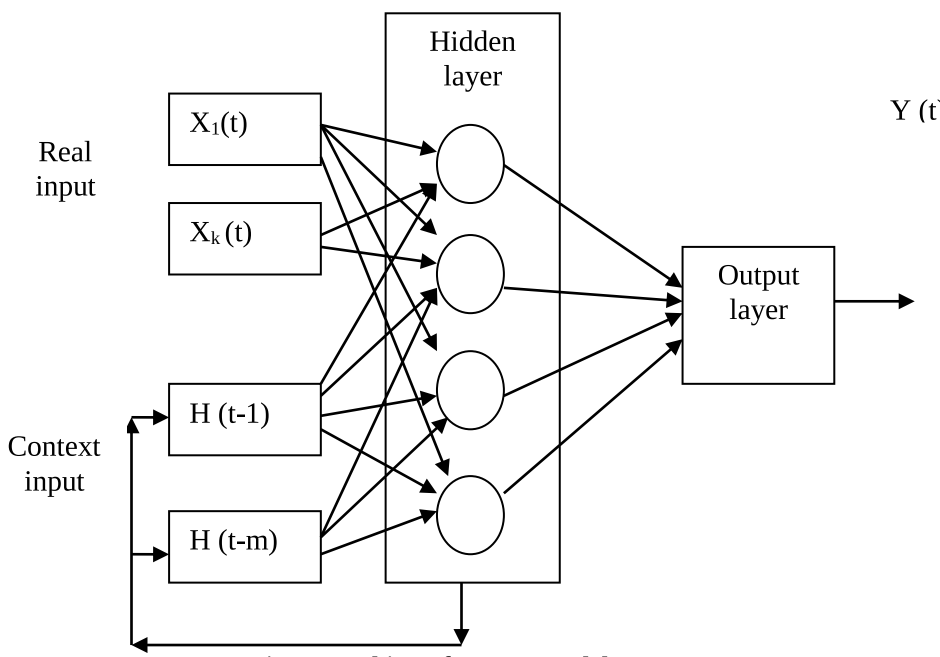

Fig. 1 Working of ERNN model

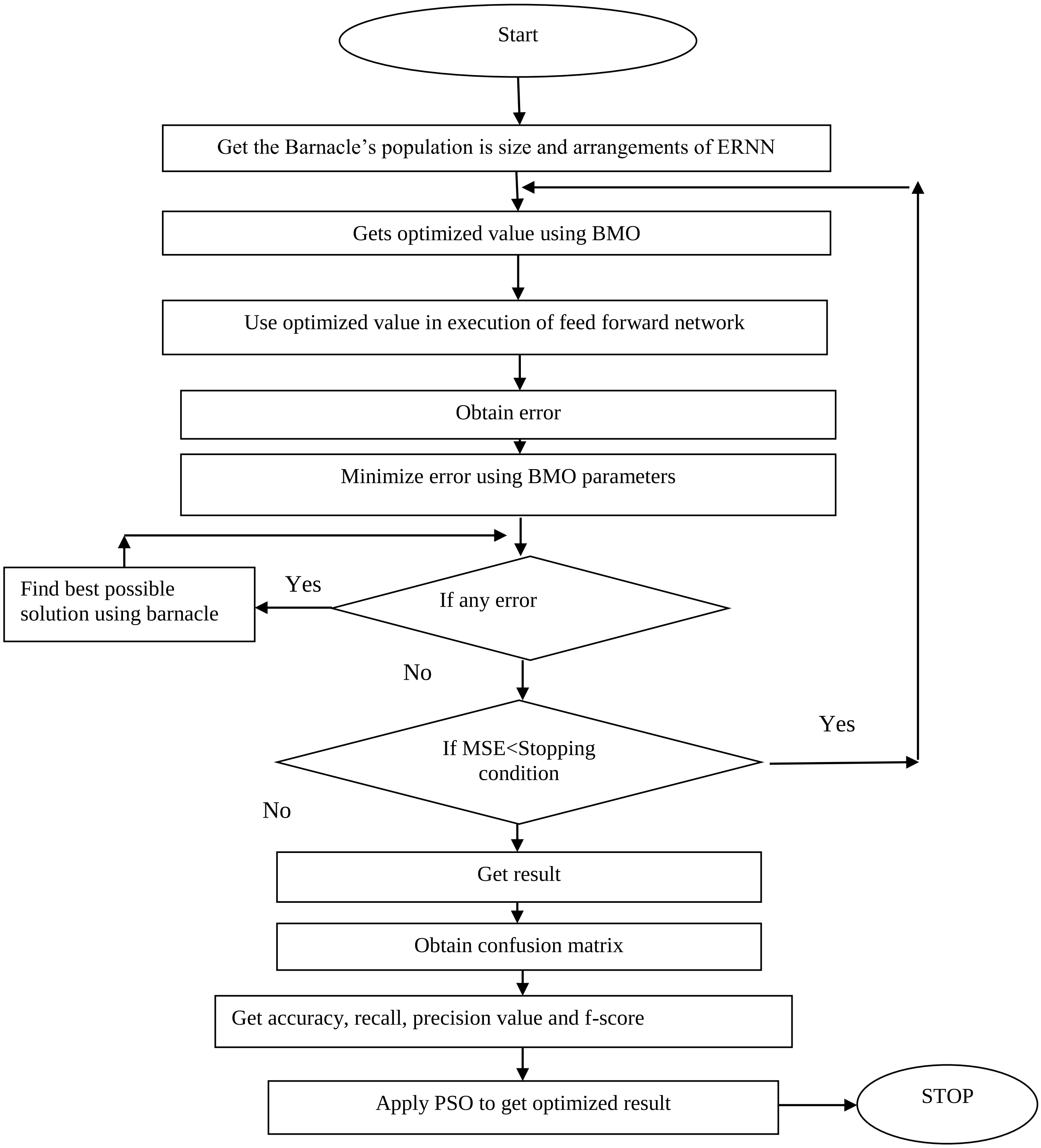

Fig. 2 Flow chart of proposed work

In this, Barnacle's preys have been established in the form of network solution, and BMO is set at an optimized value that helps the feed-forward network to execute. During the operation, errors are obtained. The obtained Error-values are minimized by configuring network parameters using BMO. Barnacles are generated with the help of Brownian motion out of some particular situation. If there is any error, the then-current position is changed, and movement is made toward an accurate solution. It is selected in an arbitrary manner considering the fitness of prey. It finds the best possible solution at each epoch until the network becomes converged and output is obtained for prediction. These operations are repeated till MSE has been found lesser as compare to terminating conditions. The testing is performed to check the accuracy and reliability of the model. Testing of the model is made to get a confusion matrix. Then accuracy, f-score, precision value, and recall value have been obtained. The obtained values are passed to the PSO mechanism in order to get the optimized accuracy, f-score, as well as precision value.

OBTAINING CONFUSION MATRIX

A confusion matrix has been considered as a table that is utilized to explain the performance of a classification model on a group of test data. True values are known in such test data. The confusion matrix is generated to present true positive (TP), true negative (TN), false positive (FP), false negative (FN).

	Actually positive (1)	Actually negative (0)
Positively Predicted (1)	TP	FP
Negatively Predicted (0)	FN	TN

Fig. 3 Confusion Matrix

Parameters

Factors that are used for the purpose of output verification become reliability, transparency, f1 score, recall, in addition to others given below:

1. Accuracy (A) is equal to
2. Precision (P) is equal to

   (TP+TN)/(TP + FP + FN + TN)

3. Recall (R) is equal to TP / TP + FP
4. F1 score is equal to
   TP / TP + FN 2 x (R x P) / (R + P)

PARTICLE SWARM OPTIMIZATION

The particle swam optimization has been considered as the computational mechanism which is able to optimize issues by trying to enhance the solution with respect to a particular measure of quality. PSO is usually started with a set of random particles, also known as solutions. They find optima by modifying the generations. The following two "best" values in each iteration change each particle. First has been considered as finest (fitness) solution that has ever attained. There is also a value for fitness. This is known as the pbest value. 'best' value has been deemed to be the best value by the PSO. Any portion of the population is produced. The best value is the finest in the world and is sometimes referred to as Gbest. In its topological neighbors, particles take part in the population. The best value is therefore termed the best local value. It's also called lbest. After finding the two best values, the particle is updating its velocity as well as positions with the following (a) and (b) equations.

V [ ] = v [ ] + c1 * rand() * (pbest [ ] – present [ ]) + c2 * rand ( ) * (gbest [ ] – present [ ]) (a)

present[ ] = persent [ ] + v [ ] (b)

v[ ] has been termed as particle velocity. On other hand persent [ ] is considered as present particle (solution). Pbest [] and gbest [ ] has been stated before. rand ( ) has been considered random number among (0 , 1). Here c1, c2 have been considered as learning factors. Generally c1 and c2 are equal to 2.

The process of PSO optimization is given below:

For every particle

1. Initiate particle
2. Do

For every particle

3. Compute the value for fitness

   If the value of fitness has been found better as compared to personal best fitness value (pBest) in history

   set current value to new pBest

4. Choose particle having the best fitness value of each particle as gBest

For every particle

5. Find particle velocity
6. Modify the location of the particle
7. Repeat until minimum error criteria are not found or iteration not completed.

The objective function used for the optimization

The PSO model is using objective function in order to get the optimized result. Here aS3 is array of data items and n is presenting the total count of number , x is showing the input to objective function and o is returning variable from objective function.

```
aS3 = [data1 data2 data3…data_n];
for j=1:n
    bS3(j) = sum((x'-aS3(:,j)).^6);
  end

o = (1/iteration + sum (1./([1:n]+bS3))).^(-1);
```

IV. RESULT

A. Parameter comparison in Hybrid model

Simulation has presented the comparison of Recall, F1 score, and precision in the case of the proposed hybrid model. Various iteration is considered to perform a comparison of Accuracy, Recall, F1 score, and precision value in the hybrid model

Table 1 Comparison of accuracy, precision, Recall value, and F-score in Hybrid model

Test cases	Accuracy	Precision	Recall value	F-score
100	99.15	92.37	97.19	96.48
200	98.34	92.19	97.32	97.06
300	98.63	91.57	96.78	97.23
400	98.94	92.52	96.87	96.67
500	98.67	92.89	97.33	97.03
600	99.06	92.26	96.97	97.29
700	98.97	92.46	97.61	97.61
800	98.65	92.21	96.96	97.47
900	98.31	92.49	96.75	97.19
1000	98.32	92.39	97.09	97.34

Considering table 1, figure 4 is presenting the graphical comparison of performance parameters.

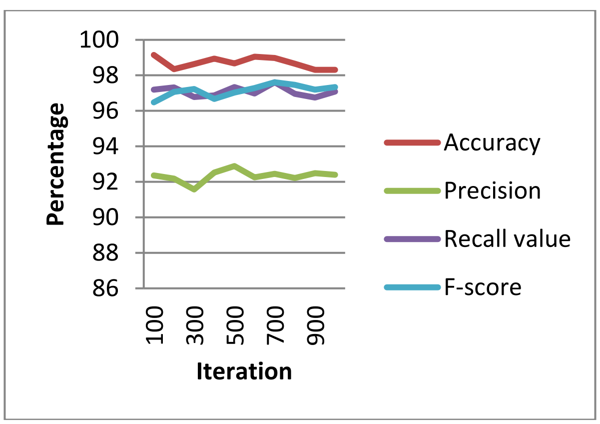

Fig. 4 Comparison of Accuracy, precision, and recall value and F-score

B. PSO integration in Proposed Model

a) Accuracy

Accuracy has been optimized for previous and hybrid work to perform a comparison of overall **accuracy** in the case of existing and hybrid models.

Optimization of existing accuracy

Best solution found ans = 97.9164
Best objective value ans =.3457
Elapsed time is 0.065276 seconds.

Optimization of proposed work accuracy

Best solution found ans = 98.7898
Best objective value ans =0.3417
Elapsed time is 0.073711 seconds.

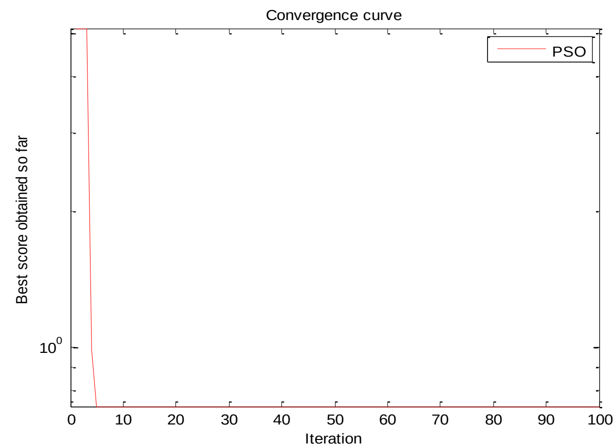

Fig. 5 Optimization of accuracy using PSO

Table 2 Accuracy comparison in case of Existing, Hybrid model with average, and a hybrid model with optimization

Existing model	Hybrid model average	Hybrid model optimized
97.9164	98.64839	98.7898

The following chart has been plotted considering accuracy for Existing, Hybrid model with average, and a hybrid model with optimization

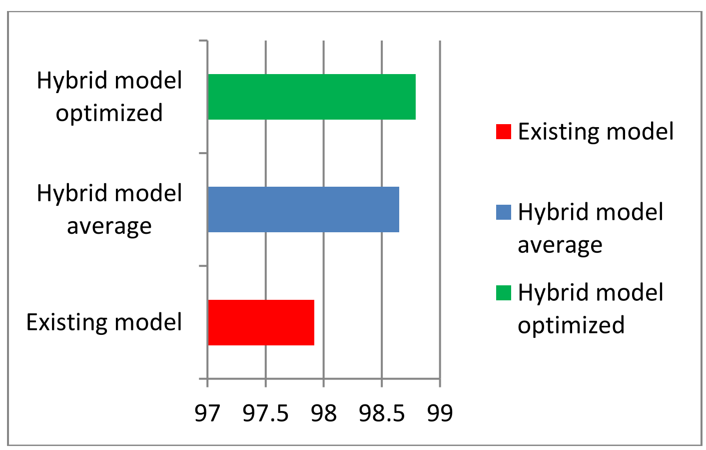

Fig. 6 Comparison of optimization accuracy in case of Existing, Hybrid model with average, and a hybrid model with optimization

b) Precision

Precision has been optimized for existing and hybrid work to perform a comparison of overall precision in the case of existing and hybrid models.

Optimization of existing precision

Best solution found ans = 90.8713
Best objective value ans = 0.3426
Elapsed time is 0.064067 seconds.

Optimization of proposed work precision

Best solution found ans = 92.1650
Best objective value ans = 0.3424
Elapsed time is 0.108411 seconds.

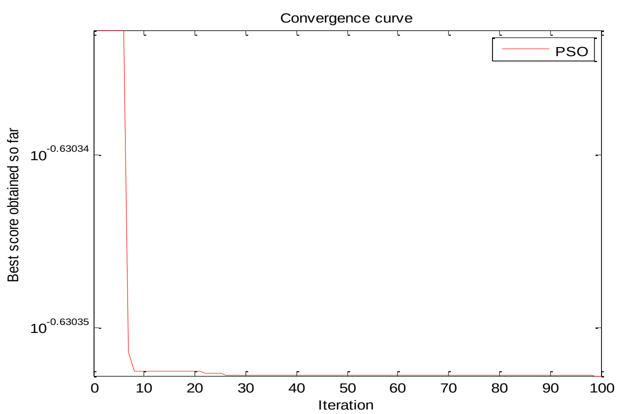

Fig. 7 Optimization of precision value using PSO

Table 4 Precision comparison in case of Existing, Hybrid model with average, and a hybrid model with optimization

Existing model	Hybrid model average	Hybrid model optimized
90.8713	92.26697	92.1650

The following chart has been plotted considering the Existing, Hybrid model with an average and hybrid model with optimization

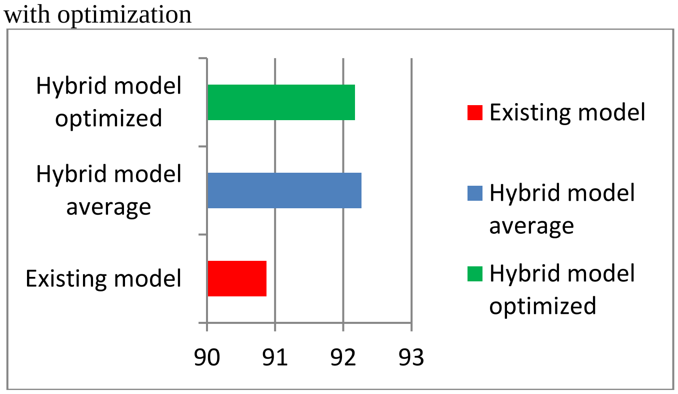

Fig. 8 Comparison of precision in case of Existing, Hybrid model with average, and a hybrid model with optimization

c) Recall

Recall has been optimized for existing and proposed hybrid work to perform a comparison of overall recall in case of existing and hybrid model

Optimization of existing Recall

The best solution found Best objective value
Elapsed time is 0.064308 seconds.
ans = 96.7675 ans = 0.3433

Optimization of proposed work Recall

Best solution found Best objective value Elapsed time is 0.073951 seconds
ans = 97.1111 ans = 0.3412

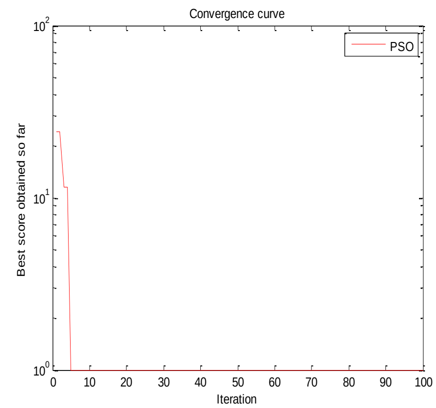

Fig. 9 Optimization of Recall value using PSO

Table 4 Optimization Recall in case of Existing, Hybrid model with average, and a hybrid model with optimization

Existing model	Hybrid model average	Hybrid model optimized
96.7675	97.00662	97.1111

The following chart has been plotted considering the optimization of the Existing, Hybrid model with an average and hybrid model with optimization

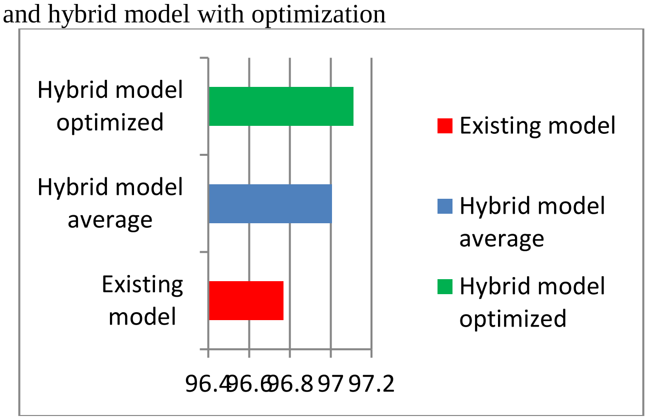

Fig. 10 Comparison of Recall in case of Existing, Hybrid model with average, and a hybrid model with optimization

d) F-Score

F-score has been optimized for existing and proposed hybrid work to perform the comparison of overall **f1score** in case of existing and hybrid model

Optimization of existing f1 score

Best solution found ans = 93.2437
Best objective value ans = 0.3412
Elapsed time is 0.076961 seconds.

Optimization of proposed work f1 score

The best solution found ans = 96.8475
Best objective value ans = 0.3421
Elapsed time is 0.065332 seconds.

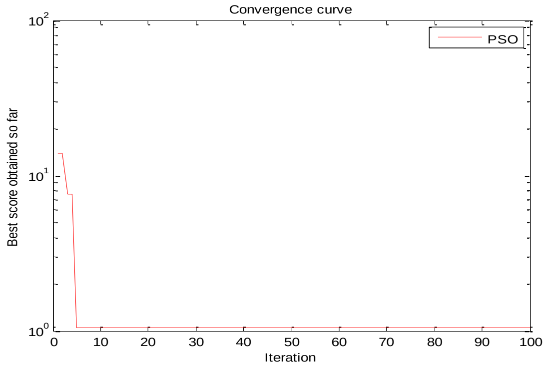

Fig. 11 Optimization of F-Score using PSO

Table 6 Optimization f1score in case of Existing, Hybrid model with average, and a hybrid model with optimization

Existing work	Hybrid model average	Hybrid model optimized
93.2437	97.07336	96.8475

The following chart has been plotted considering the optimization of the Existing, Hybrid model with the average and hybrid model with optimization

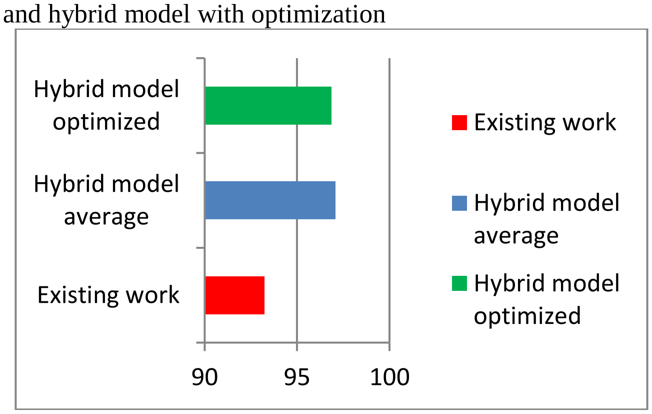

Fig. 12 Comparison of optimization f1score in case of Existing, Hybrid model with average, and a hybrid model with optimization

V. CONCLUSIONS

The research concludes that the proposed work has shown improvement in reliability as compared to the traditional neural network model. It has been concluded that the proposed model takes less time as compare to existing models. Optimization of the model has increased the reliability during effort estimation. This model has been provided scalability along with increased performance. The optimized time in the case of the proposed work is 0.5350, which is 0.6180 in the case of the existing model. On the other hand, the existing model has provided optimized accuracy, f1score, recall, precision value 97.92, 93.25, 96.77, 90.88, respectively. But the proposed work has provided optimized accuracy, f1score, recall, precision value to 98.79, 96.85, 97.12, and 92.17, respectively, which is more as compared to existing work.

VI. SCOPE OF WORK

The use of PSO is increasing day by day for optimization of the result. Such could be used in different areas such as IOT, Health care, cloud computing by providing an efficient, optimized approach. Proposed work has played a significant role in getting the optimized solution for the testing effort by integration of PSO to ERNN and BMO.

REFERENCES

[1] Ioan Cristian Trelea., The particle swarm optimization algorithm: convergence analysis and parameter selection, Information Processing Letters, 85 (2003) 317–325.

[2] W. Pedrycz , B.J. Park , N.J. Pizzi., Identifying core sets of discriminatory features using particle swarm optimization, Expert Systems with Applications, 36 (2009) 4610–4616.

[3] Alper Unler,Alper Murat., A discrete particle swarm optimization method for feature selection in binary classification problems". European Journal of Operational Research 206 (2010) 528–539.

[4] Jun Wang, Yan Zhao, Ping Liu., Effective Feature Selection with Particle Swarm Optimization based One-dimension Searching. Systems and Control in Aeronautics and Astronautics (ISSCAA), (2010) 702-705.

[5] Dian Palupi Rini, Siti Mariyam Shamsuddin, Siti Sophiyati Yuhaniz., Particle Swarm Optimization: Technique, System and Challenges, International Journal of Computer Applications,14(1) (2011) 0975 – 8887.

[6] T. W. Karjala, D. M. Himmelblau and R. Miikkulainen, "Data rectification using recurrent (Elman) neural networks, [Proceedings 1992] IJCNN International Joint Conference on Neural Networks, Baltimore, MD, USA, 2 (1992) 901-906. doi: 10.1109/IJCNN.1992.226873.

[7] Yuan-Chu Cheng, Wei-Min Qi, and Jie Zhao., A new Elman neural network and its dynamic properties, IEEE Conference on Cybernetics and Intelligent Systems, Chengdu, (2008) 971-975. doi: 10.1109/ICCIS.2008.4670784.

[8] N. Chowdhury and M. A. kashem., A comparative analysis of Feed-forward NN & RNN to find intrusion, International Conference on Electrical and Computer Engineering, Dhaka, (2008) 488-492. doi: 10.1109/ICECE.2008.4769258.

[9] Kerstner, M. (2011). Software Test Effort Estimation Methods.

[10] Amin Dastanpour, Suhaimi Ibrahim, Reza Mashinchi, Ali Selamat., Comparison of GA optimization on ANN and SVM in IDS, Published in IEEE Conference on Open Systems (ICOS), (2014).

[11] Kafle, L. P., An empirical study on software test effort estimation. Int. J. Soft Comput. Artif. Intell, 2(2) (2014).

[12] Adali, O. E., Karagöz, N. A., Gürel, Z., Tahir, T., & Gencel, C.

(2017, August). Software Test Effort Estimation: State of the Art in Turkish Software Industry. In 43rd Euromicro Conference on Software Engineering and Advanced Applications (SEAA), (2017) 412-420. IEEE

[13] B. Y. Suprapto and B. Kusumoputro, A Comparison of Back Propagation Neural Network and Elman Recurrent Neural Network Algorithms on Altitude Control of Heavy-lift Hexacopter Based on Direct Inverse Control, International Conference on Electrical Engineering and Computer Science (ICECOS), Pangkal Pinang, (2018) 79-84, doi: 10.1109/ICECOS.2018.8605240.

[14] M. H. Sulaiman et al., Barnacles Mating Optimizer: A Bio-Inspired Algorithm for Solving Optimization Problems, 19th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD), Busan, (2018) 265-270. doi: 10.1109/SNPD.2018.8441097.

[15] M. H. Sulaiman, Z. Mustaffa, M. M. Saari, H. Daniyal, I. Musirin, and M. R. Daud., Barnacles Mating Optimizer: An Evolutionary Algorithm for Solving Optimization, IEEE International Conference on Automatic Control and Intelligent Systems (I2CACIS), Shah Alam, (2018) 99-104. doi: 10.1109/I2CACIS.2018.8603703.

[16] M. H. Sulaiman, Z. Mustaffa and O. Aliman, "An Application of Barnacles Mating Optimizer for Solving Economic Dispatch Problems, IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT), Amman, Jordan, (2019) 835-839, doi: 10.1109/JEEIT.2019.8717377.

[17] Moreira Nascimento, L. F. Vismari, P. S. Cugnasca, J. B. Camargo Júnior, and J. Rady de Almeira Júnior, A Cost-Sensitive Approach to Enhance the use of ML Classifiers in Software Testing Efforts, 18th IEEE International Conference On Machine Learning And Applications (ICMLA), Boca Raton, FL, USA, (2019) 1806-1813. doi: 10.1109/ICMLA.2019.00292.

[18] F. Xiaorong, J. Shizhun, and M. Songtao, Research on Intelligent Network Auto Software Testing Technology, 2nd International Conference on Safety Produce Informatization (IICSPI), Chongqing, China, (2019) 1-5. doi: 10.1109/IICSPI48186.2019.9096016.

[19] B. Kamanditya and B. Kusumoputro., Elman Recurrent Neural Networks Based Direct Inverse Control for Quadrotor Attitude and Altitude Control, International Conference on Intelligent Engineering and Management (ICIEM), London, United Kingdom, (2020) 39-43. doi: 10.1109/ICIEM48762.2020.9160191.

[20] S. Ahmed, K. K. Ghosh, S. K. Bera, F. Schwenker and R. Sarkar, Gray Level Image Contrast Enhancement Using Barnacles Mating Optimizer., in IEEE Access, 8 (2020) 169196-169214. doi: 10.1109/ACCESS.2020.3024095.