An Artificial Neural Network Genetic Algorithm with Shuffled Frog Leap Algorithm for Software Defect Prediction

S.V.Achuta Rao, P.Santosh Kumar Patra

Abstract*: Defect prediction performances are significant to attain quality of the software and to understand previous errors. In this work, for assessing the classification accuracy, precision, and recall and F measure for various classifiers are used. The artificial neural network optimizations make the assumption that more than two algorithms for one optimization have been implemented. The optimization makes use of a heuristic for choosing the best of the algorithms for being applied in a particular situation. An approach of hybrid optimization for designing of the linkages method and is used for the dimensional synthesis of the mechanism. The ANN models are assisted in their convergence towards a global minimum by the multi-directional search algorithm that is incorporated in the GA. The results have shown an accuracy of classification of the NN-hybrid shuffled from algorithm to perform better by about 5.94% than that of the fuzzy classifiers and by about 3.59% of the NN-Lm training and by about 1.42% of the NN-shuffled frog algorithm..

Keywords* : Hybrid Optimization, Fuzzy Classifiers, Defect Prediction, Classification and Hybrid Shuffled Frog Leap.

I. INTRODUCTION

The practice of identifying the defective software system parts is referred to as SDP. The software defects can be effectively predicted using The Software Defect Prediction Model (SDPM). Such models can use various software metrics that are available for carrying out the SDP mechanism. The Efficiency of SDP is needed for all software systems. In the first place, it enhances both the quality and the testing efficiency. It enhances customer satisfaction in the second place. Next, it reduces the cost of defect correction and finally, it aids in the delivery of reliable software.

In order to predict the defects in an efficient manner the developers may adopt many techniques for achieving the results desired. Many techniques are needed to ease this process. The choice of technique however, is a factor that has some concern. Many soft computing approaches are suggested for SDP. Soft computing is that which is used for the combination of techniques of computer science like Artificial Intelligence (AI), Machine learning techniques and other disciplines of engineering. Many models are being proposed for the SDP that use different techniques of machine learning for learning and predicting the defected modules in the software system

A latest class of heuristics of optimization are considered here. For constructing a hybrid register allocator that makes a choice from two register allocation algorithms which are linear scan and graph colouring. Most of the SDP studies aim in correctly classifying the software artefacts like the subsystems or the files to make them fault-prone or not.The other SDP studies have been interested in the prediction of the defects that are part of the software artefacts to ensure they are ranked.

The objective is to create an allocator which will strive to get the required balance between 2 factors that strive to select a better packing of variables for the registers (ensuring efficient run time performance is achieved) also making an attempt to bring down the allocator's overhead. The complications that arise out of real world problems taking into consideration the necessity for advanced formulas of optimization for assisting in the problems of optimization.

The main aim of these problems is the calculation of minimum objective functions and most of the algorithms that are population based have been proposed for finding optimal solutions and the metaheuristics has been able to establish itself as a practical approach for simulating solutions that are optimal. But the researcher has to make not that all these methods have been designed for solving combination optimization issues and for implementing this and does not ensure the availability of simulated noises.

There are some disadvantages in the ANN and it is that there is a fixation at a local minimal point and also a snail pace learning process. like any typical ANN, for machine defect, the BP network is used and its identification will its superior ability of nonlinear mapping. But it does have many shortcomings like a sluggish convergence and low precision of solution. Normally the parameters of training of the BP network model are selected in a random manner. Because of this, the starting weights and their thresholds are capable of being assigned in an inappropriate manner. Thus, it is easy for them to fall into a local extreme fat the time of the training for this network. In this literature, certain algorithms are attempted for improving the BP network's performance.

Revised Manuscript Received on October 31, 2019.
***** Correspondence Author
Dr. S.V.Achuta Rao Professor, Dept. Of IT, St. Martin's Engineering College, Dhulapally, Secunderabad, India - 500100, drsvarao@gmail.com
Dr.P.Santosh Kumar Patra Principal & Professor, Dept of CSE, St. Martin's Engineering College, Dhulapally, Secunderabad, India - 500100, drpskpatra@gmail.com.

II. LITERATURE SURVEY

A hybrid optimization reduces the compilation effort that uses an effective algorithm mostly but also uses a very costly and efficient algorithm for optimization which seldom when it deserves gets the extra advantage that may be worth all its effort. This also brings together the merits of the stochastic and the deterministic optimization. This stochastic optimization method is on the basis of a real-valued EA .It is used for a vast exploring of the space of design variable that searches for the best linkage [9].

The GAs belongs to evolutionary optimization methods that imitate natural evolution laws for improving the performance of the group of individuals. They are on the basis of the 'survival of the fittest' principle of Darwin. For this approach the Classical GA is implemented. Once the representation of binary encoding of these design variables are selected, this population of individuals are generated randomly [4].

The GA permits the optimization of a different types of issues like the TSP can link to the circuit design, problems of delivery and scheduling. The mechanism for problem solving which can generate many results which are appropriate is GA. The final result is not impacted by the bad parameter and the GA also ensures the such parameters are discarded. By using the fitness function, the GA can also solve the problem with a more complex and finite number of parameter. The GA can also associate to the situation in the real world as opposed to encoding the genes [10].

The GA does not always have a global optimum more so when the entire solution has different population. A device depends on the speed of the computer and only the real time application will be able to produce a response time that is quick. This return encoded result may not be understandable in the application for a non professional, to the problem of the user as the GA will not simulate instructions but show the encoded chromosome as the solution values (that is not applicable to all). Each time the GA gives different results that allow only situations that tolerate the results of trial and error.

The complex problem difficult to apply or understand is the GA. The GA is an algorithm of heuristic optimization that is based on the principles which are inspired from the mechanisms observed in the natural systems for acquiring either an optimal or a suboptimal solution. When the structures of the NN are very complex and there are large samples for training that can be trapped easily in the local optimum, the GA's and their convergent speed is low. The same is slow at the time the NNs are complex and trapped in the local optimum. Also, the ability of the GA in fine-tuning the solutions is not good as , for escaping from a trap, the probability of the selected mutation may be very small [12]. The GA is a very important yet into a robust tool for optimization.

This algorithm is based on the heuristic computing technology of swarm intelligence that is highly efficient in its performance and also has a good capability of global search. This population is made up of frogs that can communicate with one another. Every frog can be seen as a meme vehicle. The population evolves through a population communication. The algorithm has advantages like simple steps and a few parameters like a fast speed and also easy realization. But there are some disadvantages in the original algorithm like non-uniform initial population and slow rate of convergence which limit the ability of local search and also its adaptive ability and rate of premature convergences [5].

The SFLA has been employed with the aim of optimizing the initial values of the weights and the thresholds of the ANNs in a successful manner. But, the speed of convergence and the accuracy of the solution for the SFLA comes down to a significant extent, as the degree of complicity and the problem's dimension increase. So it becomes important to develop some new algorithms for the improvement of its performance [13, 14]. Here in this work an algorithm which is novel and hybrid in nature makes an integration of the GA and the SFLA method has been proposed.

III. METHODOLOGY

Most models of defect prediction can classify the software models into the fault prone ones and the non-fault prone ones. With this type of a binary classification, the researchers often start evaluating with the counting of the number of models that are predicted correctly as well as incorrectly and further by placing them into the confusion matrix. This confusion matric has four scores. The True Positive (TP) score and the True Negative (TN) score are normally counted for each rightly classified fault prone module and also the non-fault prone module.

A False Positive (FP) score and a False Negative (FN) score are duly counted for each misclassified and non-fault-prone module and also for fault-prone modules. By means of using these scores it becomes possible to calculate many measures of evaluation [66]. The AUC is the value of the ROC curve that is integrated and spans from 0 to 1.

Accuracy: Accuracy is known as the "correct rate of classification". It is evaluated by computing the ratio of correct prediction to total prediction that is computed.

Sensitivity: Sensitivity which is also known as the TP rate is calculated by estimating the % of correctly identified and not-defective modules of software and is formulated.

Precision: Also known as correctness, sometimes, it is computed by taking the actual proportion of correctly recognized and defect free modules which are predicted totally and belong to not-defective modules of software by the classifier and is formulated.

F-Measure: this is measured by calculating and taking the harmonic mean of the precision and the sensitivity and is calculated.

KC1 dataset is a NASA systems Data Program inspection-cum-improvement foretelling software developing prototype. KC1 is a C++ outline performing storing control for ground data receiving/formulating comprising McCabe and Halstead attributes, code miners and module based processes [7]. KC1 dataset has 2109 occurrences and twenty-two moved characters that include 5 idiosyncratic LOC, 3 McCabe systems, 12 Halstead systems, a branch computation and 1 unbiased field. Dataset's typical data is the all out operational quantity, sketch intricacy, McCabe's Line computation of Code, Cyclomatic intricacy, software package's length, work, Halstead, class and others.

Examples from dataset:

Example 1 - 1.1, 1.4, 1.4, 1.4, 1.3, 1.3, 1.3, 1.3, 1.3, 1.3, 1.3, 1.3, 2, 2, 2, 2, 1.2, 1.2, 1.2, 1.2, 1.4, false

Example 2 - 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, true

Example 3 - 83, 11, 1, 11, 171, 927.89, 0.04, 23.04, 40.27, 21378.61, 0.31, 1187.7,65, 10, 6, 0, 18, 25, 107, 64, 21, true.

This section makes use of methods like the fuzzy classifiers, the proposed NN that uses hybrid SFLA and the NN utilizing LM training algorithm. The efficacy of such classifiers is that they are evaluated for the classification of the defects in the software.

3.1 Fuzzy Classifiers

This Classification is a learning problem that is supervised which takes labelled data and samples to generate models (classifiers) which classify the new data into various groups that are predefined. This problem of classification may be solved by using fuzzy logic. This fuzzy logic if applied to the computers makes them to emulate the process of human reasoning, quantify the information that is not precise, make the decisions on the basis of incomplete and vague data and by applying the process of defuzzification reach certain conclusions.

The knowledge base of this fuzzy classifier refers to the collection of these rules. With the expression of the input-output relationship as a group of fuzzy if-then rules where the "if" is a group of linguistic variables of every fuzzy set and the 'then' part has the class labels the performance of qualitative reasoning is made for the inference of the results [2].

3.2 Neural Network (NN) using Levenberg Marquardt (LM) Training Algorithm

Each neuron's output function is estimated based on sigmoid functions that are done in a step to step manner that is found suitable for functions consecutive to each other. The NNs adapt well to their environment and make an adjustment of their weight and strength depending upon them learning patterns. So if such networks are trained properly, their accuracy in terms of answers may solve to get new answers based on patterns that are similar [118].

This is a method of BP optimization that is made use of for the training of the network models and is perhaps the quickest available choice to the present supervised learning models. This LM algorithm is very akin to the quasi-Newtonian models which were designed for addressing a second-order training speed without the help of Hessian matrix formulas. Such models are calculated as the sum of squares (which is a training feed forward network) and may be replaced by using the equation below (1.1):

$H = J^T J$,Where the gradient may be computed as:

$$g = J^T e \quad \text{.....} \quad (1.1)$$

Where J represents the Jacobian matrix consisting some network error derivatives as the weights and the biases and e symbolizes the vector of the network errors. R the regression value measures the correlation estimated between the outputs and the targets.

3.3 Artificial Neural Network with Hybrid Genetic Algorithm and Shuffled Frog Leaping Algorithm (ANNGASFLA)

The GA is a search algorithm that is stochastic and this optimization technique was developed by Holland. This method imitates the natural selection process in case of the evolution of biological species. It makes used of techniques of stochastic optimization for generating potential solutions by means of a randomized number generator. An essential need of GA is an advancement that is evaluative for the objective value functions for each of the variable decisions.

Owing to this stochastic technique of SFLA not needing specific guidelines for the search variables [128], the individual GA population makes a search and an evaluation of the solutions of the candidates that mature gradually in a chronological manner for combining with that of an optimal solution. In this kind of an equation, each candidate solution is looked at through a linear string composition which contains 0s and 1s that are known as chromosomes.

The final population size solutions as well as the iterations are known as the generation. This process may be terminated by making essential certain stop criteria definers for attaining the required number of generations or also getting the fitness level desired. So to get the next generation of the GA variables we need three basic operators which are the mutations, the crossovers and the reproductions.

In the formula represented above, the ideal and best solutions or the nests that are selected portray the optimal solution (which denotes the weight space and the bias corresponding this in the optimization studies of the NN) compared to this problem and the amount of food source that is portrayed in this solution [8].

According to recent studies, the efficiency of the GA in the improvement of the performance of the ANN and bringing down its drawbacks if considered. A hybridized genetic formula is used to A and this offers an opportunity to choose easily a suitable objective function..This results in the increasing ANNs prediction power [7].

So the techniques of ANN that are used in the GA-based systems is duly trained under the GA formulas as opposed to the basic BP formulas. Which means the connection of network and their weights and biases as opposed to the random generation have been optimized with the GA. In the GASFLA there is a crossover operation done in both local and global iteration time but in the GA there is just one such.

As according to Figure 4.1 the pseudo code for the Artificial Neural Network GA-SFLA algorithm has been shown.

1. Set m Number of memeplex
2. Set n Number of frogs in each memeplex
3. Set Number of local iteration in each memeplex
4. Set Number of algorithm iteration
5. Begin;
6. Generate random population of P solution (individuals).
7. For each individual ate the fitness (i);
8. Sort the whole population P in descending order of their fitness;
9. Determine
10. Divide the population P into m memeplex;
11. For each memeplex;
    11.1. Shuffled leaping method
    a. Determine the best and worst individuals);
    b. Change in frog (D_i) = rand × ($X_b - X_w$)
    c. New position of worst frog; X_w = current Position $X_w + D_i$
    d. if $X_{wnew} \leq X_{wcurrent}$ then exchange this two individuals; (X_g is the best position which has visited b frogs). Now if the last condition does not meet, then generate the new individual;
    e. Repeat for a specific number of iteration;
    11.2. Crossover method
    a. Determine the best and worst individuals;
    b. Improve the worst individual position using GA crossover;
    b.1. Select a crossover point;
    b.2. Produce offspring 1 and offspring 2;
    b.3. For each offspring: calculate fitness;
    b.4. If any of the offspring are better than those existing in P, replace them; if not, generate a new random individual;
    c. Repeat for a specific number of iterations;
12. End.
13. Combine the evolved memeplexes;
14. Sort the population P in descending order of their fitness;
15. Check if termination = true;
16. End.

Figure. 4.1 ***Pseudo Code for Artificial Neural Network with Genetic Algorithm Shuffled Frog Leaping Algorithm (ANNGASFLA) [7]***

IV. RESULTS AND DISCUSSION

For experiments, classifiers and optimized classifiers such as NN-LM training, NN-shuffled frog, NN-hybrid shuffled frog and fuzzy classifiers are used. The result of classification accuracy, precision, recall and f measures respectively are shown in Table 4.1 to 4.4 and Figure 4.2 to 4.5.

Table 4.1 Classification Accuracy

Techniques	Classification Accuracy %
Fuzzy Classifier	91.71
Neural Network - LM training	93.78
Neural Network-Shuffled frog	95.85
Neural Network-Hybrid Shuffled frog	97.23

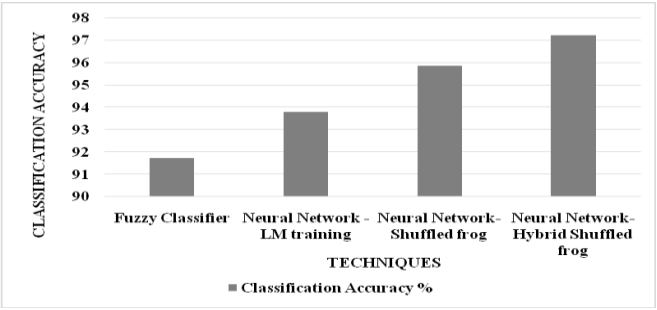

Fig 4.2 Classification Accuracy

From the figure 4.2 it is observed that the classification accuracy of NN-hybrid shuffled frog algorithm performs better, by 3.6% NN-LM training ,by 1.43% NN- shuffled frog algorithm and by 5.84% than fuzzy classifiers.

Table 4.2 Precision

Techniques	Precision
Fuzzy Classifier	0.8679
Neural Network - LM training	0.8900
Neural Network-Shuffled frog	0.9287
Neural Network-Hybrid Shuffled frog	0.9432

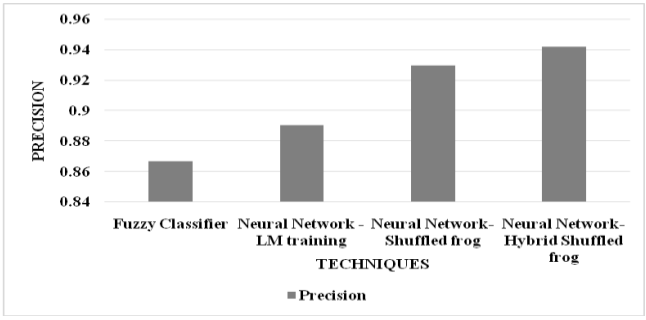

Fig 4.3 Precision

From the figure 4.3 it is observed that the precision of NN-hybrid shuffled frog algorithm performs better by 1.24% NN- shuffled frog algorithm, by 5.5% NN-LM training and by 8.22% than fuzzy classifiers .

Table 4.3 Recall

Techniques	Recall
Fuzzy Classifier	0.8455
Neural Network - LM training	0.9031
Neural Network-Shuffled frog	0.9287
Neural Network-Hybrid Shuffled frog	0.9672

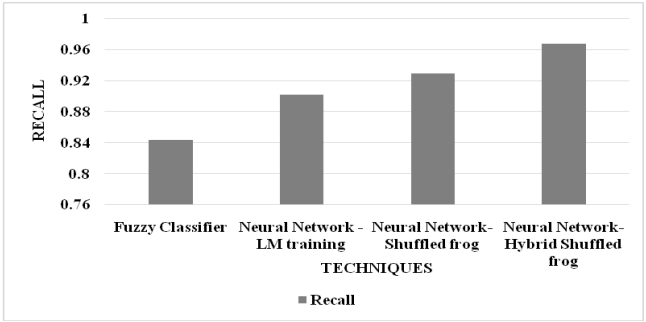

Fig 4.4 Recall

From the figure 4.4, it can be observed that the recall of NN-hybrid shuffled frog algorithm performs better by 4.0% NN- shuffled frog algorithm, by 12.7% than fuzzy classifiers and by 7.06% NN-LM training.

Table 4.4 F Measure

Techniques	F Measure
Fuzzy Classifier	0.8552
Neural Network - LM training	0.8960
Neural Network-Shuffled frog	0.9294
Neural Network-Hybrid Shuffled frog	0.9544

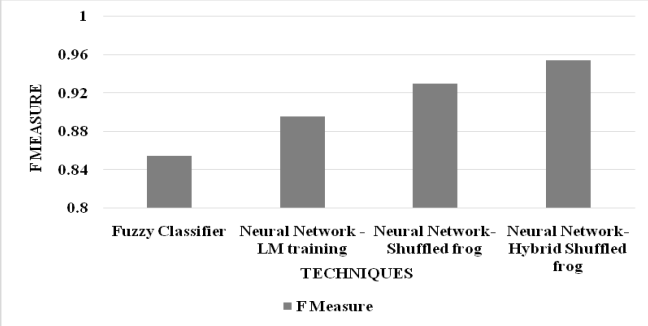

Fig. 4.5 F Measure

From the figure 3.5 it is observed that the f measure of NN-hybrid shuffled frog algorithm performs better by 6.29% NN-LM training ,by 2.56% NN- shuffled frog algorithm, and by 10.97% than fuzzy classifiers.

V. CONCLUSION

The NN with the LM and their optimization methods are used for the training of the network. This is the fastest method available today for supervised learning. This also needs less memory. This SFLA is a very simple concept and is very fast as well as accurate. The results have shown an accuracy of classification of the NN-hybrid shuffled from algorithm to perform better by about 5.94% than that of the fuzzy classifiers and by about 3.59% of the NN-Lm training and by about 1.42% of the NN-shuffled frog algorithm In the software field, the managers always try to reduce costs because of limited budgets. One way is human resources who have lesser experience. However, there is a risk of increased incidences of bugs when less experienced people are involved in coding. These do not include syntax errors. This work employs software metrics for determining faulty modules. This will help the managers to use more efficient testing techniques for those software modules that have a higher vulnerability to defects.

ACKNOWLEDGEMENT

This research was designed and supported by St.Martin's Engineering College, Dhulapally, Secunderabad, Management Personnel's. We thank to our Beloved, Visionary Dynamic Principal Dr. P. Santosh Kumar. Patra for given inception about particularly Genetic Algorithm combined with Neural Networks Optimization, which meets the Global minima accomplishing best fit function to get good results than previous Research. I am using this opportunity to express my gratitude to our Principal sir supported me throughout the work and invaluably constructive criticism and friendly advice during the Research Paper.

REFERENCES

1. Cavazos, J., Moss, J. E. B., & O'Boyle, M. F. (2006, January). Hybrid optimizations: Which optimization algorithm to use?. In Compiler Construction (pp. 124-138). Springer Berlin Heidelberg.
2. Che, H., Li, C., He, X., & Huang, T. (2015). An intelligent method of swarm neural networks for equalities-constrained non convex optimization. Neurocomputing, 167, 569-577.19
3. Devaraj, D., & Ganesh Kumar, P. (2010). Mixed genetic algorithm approach for fuzzy classifier design. International Journal of Computational Intelligence and Applications, 9(01), 49-67.
4. Duvigneau, R., & Visonneau, M. (2004). Hybrid genetic algorithms and artificial neural networks for complex design optimization in CFD. International journal for numerical methods in fluids, 44(11), 1257-1278.
5. Jiang, J. G., Su, Q., Li, M., Liu, M., & Zhang, L. (2013). An improved Shuffled Frog Leaping Algorithm. J. Inf. Comput. Science, 10, 4619-4626.
6. Kumar, R. K., & Rao, S. A. (2016). Neural Network with Hybrid Shuffled Frog: Algorithm for Software Defect Prediction. International Journal of Computer Science and Information Security, 14(5), 392.
7. Mohammadi, M., Tavakkoli-Moghaddam, R., Ghodratnama, A., & Rostami, H. (2011). Genetic and improved shuffled frog leaping algorithms for a 2-stage model of a hub covering location network. International Journal of Industrial Engineering & Production Research, 22(3), 179-187.
8. Nawi, N. M., Khan, A., & Rehman, M. Z. (2013). A new Levenberg Marquardt based back propagation algorithm trained with cuckoo search. Procedia Technology, 11, 18-23.
9. Sedano, A., Sancibrian, R., de Juan, A., Viadero, F., & Egana, F. (2012). Hybrid optimization approach for the design of mechanisms using a new error estimator. Mathematical Problems in Engineering, 2012.
10. Tabassum, M., & Mathew, K. (2014). A genetic algorithm analysis towards optimization solutions. International Journal of Digital Information and Wireless Communications (IJDIWC), 4(1), 124-142.
11. Wilamowski, B. M., Cotton, N., Hewlett, J., & Kaynak, O. (2007, June). Neural network trainer with second order learning algorithms. In 2007 11th International Conference on Intelligent Engineering Systems (pp. 127-132). IEEE.Wu, J. (2016).
12. Hybrid Optimization Algorithm to Combine Neural Network for Rainfall-Runoff Modeling. International Journal of Computational Intelligence and Applications, 15(03), 1650015 1-19.
13. Zanganeh, S., Javanmard, R., & Ebadzadeh, M. (2010). A Hybrid Approach for Features Dimension Reduction of Datasets using Hybrid Algorithm Artificial Neural Network and Genetic Algorithm-in Medical Diagnosis. In 4rd Iran Data Mining Conference (IDMC).
14. S.V.Achuta Rao, Dr. R.Kiran Kumar, October (2016) "Neural Network Optimization using shuffled Frog algorithm for Software Defect Prediction", Journal of Theoretical and Applied Information Technology (JATIT), Volume 92, No.2, pp 284-293 ISSN: 1992 -8648 E-ISSN: 1817-3195 (Elsevier Scopus, Thomson Science, Reuters Monitor, DBLP, USA Indexed).
15. S.V.Achuta Rao, Dr. R.Kiran Kumar, October (2016)" Neural Network with Hybrid Shuffled Frog Algorithm for Software Defect Prediction" International Journal of Computer Science & Information Systems (IJCSIS), Volume 14, No.5 , pp 392-398 ISSN: 1947-5500 (Scopus, SJR Indexed).

AUTHORS PROFILE

Dr. S V Achuta Rao is working as Professor & HOD-Information Technology Department at St.Martins Engineering College, Secunderabad, India. He has completed his Bachelors and Masters in CS&E from Andhra University and JNTU h Hyderabad and Ph.D., in CSE from Krishna University,

A State Governement University, Machilipatnam, Andhra Pradesh, India He is a Life Time Member of ISTE & CSE Society. He is Reviewer and Editorial Board member of reputed International Journals. He has 22 Secunderabad, India. He has 22 years of Teaching experience. His research areas are Empirical Software Engineering, Data Mining, Machine Learning, Soft computing Big Data Analytics and Network Security. He has two numbers of Patents in Machine Learning & Network Security. He has published more than 60 numbers of Publications in reputed Journals and Conferences.

Dr.P.Santosh Kumar Patra is working as Principal & Professor, Dept of CSE, St. Martin's Engineering College, Dhulapally, Secunderabad, India. He is holding B.E in CSE, M.Tech in CSE & Ph.D in CSE Degrees. His area of interest is Artificial Intelligence, Software Engineering, Data Mining and Warehousing & Wireless Network. He is an active member of ISTE, CSI, IAENG and many. Recently he has been honored as an Adviser to NAAC, Bengaluru, UGC-Paramarsh, National Cyber Safety & Security Standards, Govt. of India. He is on the editorial board of TATA MCGRAW HILL International Publication, Charulatha Publication, Hi Tech Publication etc. and also for many Journals and Publications. He has filed 9 Patents in different technologies he has been honored with "RASTRIYA GAURAV AWARD" by IIFS New Delhi in 2011, "RASTRIYA VIKAS RATAN AWARD" by EGSI New Delhi in 2012 and "Dronacharya Award" by IKON Telangana in 2018 and young leader of year 2019 by ICCI , New Delhi in 2019.He has Received the prestigious award twice from Sri E.S.L. Narasimhan, Honorable Governor of Telangana at Raj Bhawan. India.