International Journal of Engineering & Extended Technologies Research (IJEETR)

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 8, Issue 2, March - April 2026 |

DOI:10.15662/IJEETR.2026.0802302

Software Fault Prediction using Fuzzy C-Means Clustering Algorithm

K.Navaneedhan[1], .T.Ishwarya[2],T.Indhuja[2], R.Rithika[2], V.Kavya[2]

Muthayammal College of Engineering, Rasipuram, Tamil Nadu, India

Department of Electronics and Communication Engineering, Muthayammal College of Engineering, Rasipuram, Tamil Nadu, India

Publication History: Received: 25.02.2026; Revised: 20.03.2026; Accepted: 25.03. 2026; Published: 28.03.2026.

ABSTRACT: Clustering is a Unsupervised technique, its used for fault prediction in software modules. In predicting fault in program modules using a hierarchical agglomerative algorithm and construct the dissimilarity matrix between each two objects. We propose a Fuzzy C-Means Clustering based software fault prediction approach. The initial cluster center are applied to the input of Fuzzy C-Means Algorithm. The concept of clustering gain has been used to determine the quality of cluster for the evaluation of Fuzzy C-Means Clustering Algorithm. Software metrics and fault data belonging to a previous software version are used to build the software fault prediction model for the next release of the software. However there are certain cases when previous fault data are not present. In other words predicting the fault-proneness of program modules when the fault labels for modules are unavailable is a challenging task frequently arised in the software industry There is need to develop some methods to build the software fault prediction model based on unsupervised learning which can help to predict the fault–proneness of a program modules when fault labels for modules are not present. One of the such method is use of clustering techniques. Unsupervised techniques like clustering may be used for fault prediction in software modules, more so in those cases where fault labels are not available. In this study, we propose a Fuzzy c-means clustering based software fault prediction approach for this challenging problem.

KEYWORDS: Fuzzy C-Means clustering, software fault prediction, cluster center.

I. INTRODUCTION

Fuzzy C-Means is a method of clustering which allows one piece of data to belong to two or more clusters. This algorithm works by assigning membership to each data point corresponding to each cluster on the basis of between the cluster center and data point. More the data is near to the cluster center more its towards the particular cluster center. Software quality estimation models, used to predict the fault-proneness of software modules based on software metrics, are often constructed by training a classier from labeled software metrics data. Two challenges often encountered in building an accurate model are the presence of \noisy" data and the possible unavailability of fault-proneness labels in real-world projects. The performance of a model often improves if outliers and noise are removed from the training data. More important, a classier cannot be trained without fault-proneness labels. This article describes an exploratory analysis method that addresses these two challenges and that is built with clustering and the help of a software engineering expert. It is an unsupervised method since labeled training data are not required to predict the fault-proneness of software modules. We present two real-world case studies to verify the electiveness of the clustering- and expert-based approaching predicting both the fault-proneness of software modules and potential \noisy" (e.g., mislabeled) modules Software quality models are useful tools toward achieving the objectives of a software quality assurance initiative. A software quality model can be used to identify program modules that are likely to be defective. Subsequently, the limited resources allocated for software quality inspection and improvement can be targeted toward only those program modules, achieving cost-effective resource utilization.

A software quality estimation model allows the software development team to track and detect potential software defects relatively early on during development, which is critical to many high-assurance systems. A software quality model is typically trained using software measurement and defect (quality) data of a previously developed release or similar project. The trained model is then applied to modules of the current project to estimate their quality. Such a supervised learning approach assumes that the development organization has experience with systems similar to the current project and that defect data are available for all program modules in the training data. K-Means clustering is a

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 8, Issue 2, March - April 2026 |

DOI:10.15662/IJEETR.2026.0802302

nonhierarchical clustering procedure in which items are moved among sets of clusters until the desired set is reached [5]. The partitioning of data set is such that the sum of intracluster distances is reduced to an optimum value [23], [27, The K-Means algorithm is very sensitive to noise.

In [8], a method using Quad Trees has been proposed as an initialization of K-Means algorithm prone or not fault-prone by examining not only the representative points of each cluster, but also some statistical data such as global mean, median, and percentile of each metric. However, their approach required a human expert during the prediction process and it is not always possible to find an experienced expert who would have the duty to label each cluster. In this paper, the the Quad Tree-based K-Means algorithm (QDK) [8] has been applied for predicting faults in program modules. The objectives of this paper are as follows: First, Quad Trees are applied for finding initial cluster centers for K-Means algorithm.

By varying the value of threshold parameter _ a user can generate a desired number of cluster centers to be used as input to the simple K-Means algorithm. Second, the Quad Treebased algorithm is applied for predicting faults in program modules. The overall error rates of this prediction approach are compared to other existing algorithms and are found to be better in most of the cases. Clustering gain values for the best cluster by K-Means and by Quad Tree-based algorithm are very close thereby proving the effectiveness of the algorithm. To compare the performance of QDK for initialization of K-Means, experiments have been conducted in which Quad Tree-based algorithm and two other initialization techniques, Likas et al., Global K-Means algorithm [23], [24] and SAS 2004 [23], [25] have been executed and results are compared on the basis of evaluation parameters. The QDK algorithm performs fairly well on all the parameters. The Global K-Means algorithm considers each data item in each iteration leading to high complexity when number of data items and number of clusters are large and these scalability issues have also been raised by the authors. The SAS 2004 algorithm even though being linear does not provide any guidance regarding the selection of their distance measure [23]. The remaining part of the paper is organized as follows: Section 2 presents the related work on the topic. Section 3 presents an overview on the theory of Fuzzy C-Means algorithm. Section 4 presents the experimental design. Section 5 presents analysis of the results while Section 6 presents the conclusion.

II. RELATED WORK

Software quality models are useful tools toward achieving the objectives of a software quality assurance initiative. A software quality model can be used to identify program modules that are likely to be defective. Subsequently, the limited resources allocated for software quality inspection and improvement can be targeted toward only those program modules, achieving cost-effective resource utilization. A software quality estimation model allows the software development team to track and detect potential software defects relatively early on during development, which is critical to many high-assurance systems. A software quality model is typically trained using software measurement and defect (quality) data of a previously developed release or similar project. The trained model is then applied to modules of the current project to estimate their quality. Such a supervised learning approach assumes that the development organization has experience with systems similar to the current project and that defect data are available for all program modules in the training data.

Zhong et al. [2], [3] applied clustering techniques and expert-based approach for software fault prediction problem. They applied K-Means and Neural-Gas techniques on different real data sets and then an expert explored the representative module of the cluster and several statistical data in order to label each cluster as fault-prone or not fault-prone. And based on their experience Neural-Gas-based prediction approach performed slightly worse than K-Means clustering-based approach in terms of the overall error rate on large data sets. But their approach is dependent on the availability and capability of the expert. Seliya and Khoshgoftaar [4] proposed a constrained based semi-supervised clustering scheme. They showed that this approach helped the expert in making better estimations as compared to predictions made by an unsupervised learning algorithm. Seliya et al. [12] have proposed a semi-supervised clustering approach for software quality analysis with limited fault-proneness data. Most recently Catal et al. [1] proposed a metric threshold and clustering-based approach for software fault prediction. The results of their study demonstrate the effectiveness of metrics threshold and show that the standalone application of metrics threshold is easier than the clustering and metrics thresholds-based (two stage) approach because the selection of number of clusters is performed heuristically in this clustering-based method. In our present study we have presented comparative results performed on same data sets as in [1]. Bhattacherjee and Bishnu have applied unsupervised learning approach for fault prediction in software module in [16], [28]. In their work, the false negative rate (FNR) for the clustering-based approach is less than that for metrics-based approach, while the false positive rates (FPR) are better for the metrics-based approach.

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 8, Issue 2, March - April 2026 |

DOI:10.15662/IJEETR.2026.0802302

The overall error rates for both approaches remain the same. Supervised techniques have however been applied for software fault prediction [13] and software effort prediction [14], [15]. Several methods for initialization of K-Means algorithm are available in literature. Tibshirani et al. suggest a statistical method based on gap statistic to find the optimal number of clusters [20]. Pelleg and Moore suggest an algorithm which efficiently searches the space of cluster locations and number of clusters to optimize the Bayesian Information Criterion and Akaike Information Software quality models are useful tools toward achieving the objectives of a software quality assurance initiative. A software quality model can be used to identify program modules that are likely to be defective. Subsequently, the limited resources allocated for software quality inspection and improvement can be targeted toward only those program modules, achieving cost-effective resource utilization. A software quality estimation model allows the software development team to track and detect potential software defects relatively early on during development, which is critical to many high-assurance systems. A software quality model is typically trained using software measurement and defect (quality) data of a previously developed release or similar project. The trained model is then applied to modules of the current project to estimate their quality. Such a supervised learning approach assumes that the development organization has experience with systems similar to the current project and that defect data are available for all program modules in the training data.

III. PROPOSED METHODOLOGY

In the proposed system we propose a Fuzzy c-means clustering technique for software fault prediction. First select the input dataset and apply the attribute selection technique. Attribute selection technique is used to select the important attribute and reduce the number of attributes. Attribute Evaluation approach is used to evaluate the attribute and return the weight of the attribute. Then apply the ranking method for get the most weighted attributes. After the Attribute selection we get the reduced number of attributes. This reduced attributes are used to clustering approach. Given a database of n objects, a partitional clustering algorithm constructs k partitions of the data, where each cluster optimizes a clustering criterion, such as the minimization of the sum of squared distance from the mean within each cluster. Fuzzy-C means is a one type of partitional clustering approach. To cluster the data points we are using Fuzzy-C means clustering. After the clustering approach each cluster maintain the centroid value. Metric threshold is approach used to define the threshold value. Compare this metric threshold and centroid data values. If any metric value of the centroid data point of a cluster was greater than the threshold, that cluster was labeled as faulty and otherwise it was labeled as nonfaulty.

As the majority of faults are found in a few of its modules so there is a need to investigate the modules that are affected severely as compared to other modules and proper maintenance need to be done on time especially for the critical applications. In this paper, we have explored the different predictor models to NASA's public domain defect dataset coded in Perl programming language. Different machine learning algorithms belonging to the different learner categories of the WEKA project including Mamdani Based Fuzzy Inference System and Neuro-fuzzy based system have been evaluated for the modeling of maintenance severity or impact of fault severity. The results are recorded in terms of Accuracy, Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE). The results show that Neuro-fuzzy based model provides relatively better prediction accuracy as compared to other models and hence, can be used for the maintenance severity prediction of the software. When a software system is developed, the majority of faults are found in a few of its modules. In most of the cases, 55 % of faults exist within 20 % of source code. It is, therefore, much of interest is to find out fault-prone software modules at early stage of a project [1]. Using software complexity measures, the techniques build models, which classify components as likely to contain faults or not. Quality will be improved as more faults will be detected. Predicting the impact of the faults early in the software life cycle can be used to improve software process control and achieve high software reliability. Timely predictions of faults in software modules can be used to direct cost-effective quality enhancement efforts to modules that are likely to have a high number of faults. Prediction models based on software metrics, can estimate number of faults in software modules. Prediction of severity of faults:

- Supports software quality engineering through improved scheduling and project control.
- Can be a key step towards steering the software testing and improving the effectiveness of the whole process.

On comparing all the classes of WEKA's machine learning algorithms, it is observed that Logistic Model Trees and Simple Logistic algorithms are better techniques as compared with other classes of machine learning algorithms with the 65% Accuracy in prediction of fault tolerance. In both the algorithms of the WEKA project the classification algorithm is the same i.e. logistic classifier. Both the algorithms have least Mean Absolute Error and Root Mean Square Error values: 0.2145 and 0.3285. During the testing phase LMT and Simple Logistic algorithm has shown 86.66%

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 8, Issue 2, March - April 2026 |

DOI:10.15662/IJEETR.2026.0802302

Accuracy. The results of the Mamdani based fuzzy inference system are comparatively equivalent for the testing data as that of the Logistic Model Trees and Simple Logistic algorithm with 0.2183, 0.3066 and 80 as Mean Absolute Error, Root Mean Square Error and Accuracy values. The Neuro-fuzzy based Modeling technique has outperformed the other technique on the basis of the testing data with 0.1571, 0.2140 and 93.3333 as Mean Absolute Error, Root Mean Square Error and Accuracy values. It is therefore, concluded the model is implemented and the best algorithm for classification of the software components into different level of severity of impact of the fault is found to be Neuro-Fuzzy based technique. The algorithm can be used to develop model that can be used for identifying modules that are heavily affected by the faults and those can be debugged.

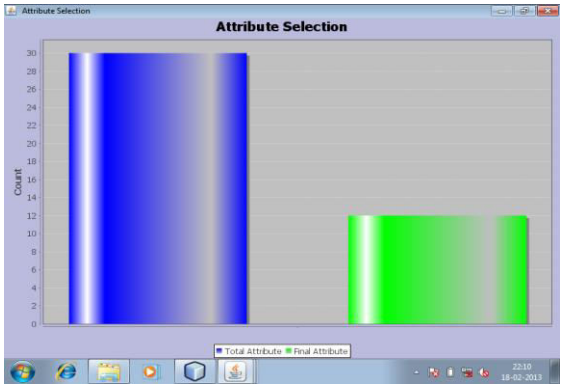
IV. EXPERIMENT AND RESULTS

4.1 Data Sets

We conducted experiments on four real data sets to test our algorithm. These data sets are: AR3, AR4, AR5 available at [22] and Iris data set [7]. Of these, the first three data sets are related to software fault prediction. The synthetic two dimensional two class data sets (SYD1 and SYD2) have been taken to illustrate the initialization algorithm. Collect Software Fault Dataset (AR3, AR4 and AR5).Read the data of the Dataset and stored into DB.Get the Attribute names for Attributes Selection.

4.2 Attribute Selection

Get all Attribute in our dataset. The dataset contains totally 30 numbers of attributes. Attribute Selection is the technique of selecting a subset of relevant features for building robust learning models. We take all attributes into our process it takes so much time for processing and increase the work burden. So, we reduce the total number of attributes and consider high relevance attributes only. Calculate the relevance of an attribute using Attribute Evaluation. We use Weka tool for attribute selection and ranker.



4.3 Fuzzy C-Means Clustering

A cluster is a collection of objects which are “similar” between them and are “dissimilar” to the objects belonging to other clusters. Fuzzy c-means is a method of clustering which allows one piece of data to belong to two or more clusters. The data’s are initially clustered and then performing Fuzzy c-means algorithm. After the clustering process we separate clustered data’s and cluster centroid. Create an algorithm for fuzzy clustering that partitions the data set into an optimal number of clusters. This algorithm should account for variability in cluster shapes, cluster densities, and the number of data points in each of the subsets. Cluster prototypes would be generated through a process of unsupervised learning.

4.4 Metric Threshold

In order to determine acceptable metrics thresholds, there are three methods described as follows [11]: 1) Experience and Hints from literature: The threshold values are specified according to the empirical researchers, previously introduced in the literature. 2) Tuning machine: This approach uses a repository of problematic items (faulty modules). Accordingly, there are chosen threshold values that maximize the number of correctly detected items. 3) Analysis of multiple versions: This method does not parameterize a strategy with several thresholds, but adds an important time viewpoint for each suspected entity. The dimensions and metrics we used in our experiments for AR# data sets are same as Catal et al. and are as follows: Lines of Code (LoC), Cyclomatic Complexity (CC), Unique Operator (UOp),

International Journal of Engineering & Extended Technologies Research (IJEETR)

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

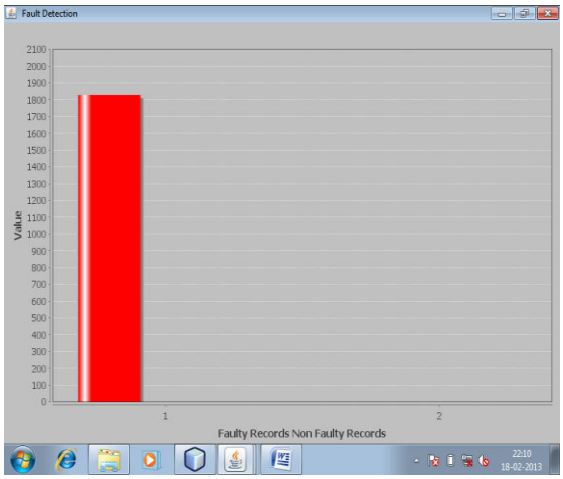
| Volume 8, Issue 2, March - April 2026 |

DOI:10.15662/IJEETR.2026.0802302

Unique Operand (UOpnd), Total Operator (TOp), Total Operand (TOpnd). Threshold vector [LoC, CC, UOp, UOpnd, TOp, TOpnd] was chosen as [65, 10, 25, 40, 125, 70] [1]. For the Iris data set we have used all the four attributes

4.5 Detect Fault

After the clustering process each cluster maintain the data point. Get the metric threshold for each cluster. If any metric value of the centroid data point of a cluster was greater than the threshold, that cluster was labeled as faulty and otherwise it was labeled as nonfaulty.



V. ANALYSIS OF THE RESULTS

The main contribution of this paper is the development of an automated way of assigning fault-proneness labels to the modules and the removing the subjective expert opinion. Subjective human interaction directly affects the quality of the software fault prediction model and adds unnecessary complexity. Software metrics and fault data belonging to a previous software version are used to build the software fault prediction model for the next release of the software. Until now, different classification algorithms have been used to build this kind of models. However, there are cases when previous fault data are not present; and hence, supervised learning approaches cannot be applied.

In this study, we propose a fully automated technique which does not require an expert during the prediction process. In addition, it is not required to identify the number of clusters before the clustering phase, as required by K-means clustering method. Software metrics thresholds are used to remove the expert necessity. Our technique first applies X-means clustering method to cluster modules and identifies the best cluster number. After this step, the mean vector of each cluster is checked against the metrics thresholds vector. A cluster is predicted as fault-prone if at least one metric of the mean vector is higher than the threshold value of that metric. Comparing Fault-Proneness Estimation Models" compared Fault-Proneness Estimation Models and concluded that over the last years, software quality has become one of the most important requirements in the development of systems and fault proneness estimation could play a key role in quality control of software products [1]. Their main objective was to find a compromise between the fault-proneness estimation rate and the size of the estimation model in terms of number of metrics used in the model itself. The methodologies used in their study were logistic regression and discriminate analysis Challagulla in his paper "A Unified framework for Defect Data Analysis using the MBR Technique" suggested the use of MBR classifier to effectively estimate defects from prior data. Better prediction models facilitate better project planning and risk/cost estimation to reduce the V&V cost for achieving high confidence levels.

A framework was developed using MBR classifier for software defect data by logical variations of its configuration parameters. Accuracy was predicted using different parameters such as true positives, false positives, true negatives, and false negatives. Finally, they concluded that if accuracy is the only criteria to estimate the software quality, then MBR with Euclidean distance and one nearest neighbour comes out be best prediction method. Costa and Oliveira in their paper on "Cluster Analysis using Growing Neural Gas and Graph Partitioning" discussed about the size and complexity of data sets. The size and complexity of data sets is ever increasing. Clustering, considered the most important unsupervised learning problem, is used to reveal structures and identify "natural" groupings on the multivariate data.

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 8, Issue 2, March - April 2026 |

DOI:10.15662/IJEETR.2026.0802302

VI. CONCLUSION

We proposed a Fuzzy c-means clustering and Metrics threshold based software fault prediction approaches for the cases where there is no priori fault data. Attribute selection method is used to select a weighted attributes and returns a most weighted attributes only. After the clustering method the data points are clustered and each cluster have the own centroid value. Metric Threshold was used to define the threshold and it's compared to each cluster centroid. Finally faulty and non faulty data points were displayed after implementation.

Proposed scheme analysed the project by using the Quad Tree based Fuzzy c-means algorithm is mainly used for clustering and its applied to over the limitations of Fuzzy c-means algorithm. But it takes so much time for executing for quad tree and Fuzzy c-means. To reduce the time and increase the accuracy of clustering in future we will apply Hierarchical Clustering Algorithm.

REFERENCES

1. C. Catal, U. Sevim, and B. Diri, "Clustering and Metrics Threshold Based Software Fault Prediction of Unlabeled Program Modules," Proc. Sixth Int'l Conf. Information Technology: New Generations, pp. 199-204, 2009.
2. S. Zhong, T.M. Khoshgoftaar, and N. Seliya, "Unsupervised Learning for Expert-Based Software Quality Estimation," Proc. IEEE Eighth Int'l Symp. High Assurance Systems Eng., pp. 149-155, 2004.
3. S. Zhong, T.M. Khoshgoftaar, and N. Seliya, "Analyzing Software Measurement Data with Clustering Techniques," IEEE Intelligent Systems, vol. 19, no. 2, pp. 20-27, Mar./Apr. 2004.
4. N. Seliya and T.M. Khoshgoftaar, "Software Quality Classification Modeling Using the PRINT Decision Algorithm," Proc. IEEE 14th Int'l Conf. Tools with Artificial Intelligence, pp. 365-374, 2002.
5. J. Han and M. Kamber, Data Mining Concepts and Techniques, second ed, pp. 401-404. Morgan Kaufmann Publishers, 2007.
6. M. Ester, H.P. Kriegel, J. Sander, and X.Xu., "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise," Proc. Second Int'l Conf. Knowledge Discovery and Data Mining (KDD '96), pp. 226- 231, 1996. http://archive.ics.uci.edu/ml/datasets/Iris, 2012.
7. P.S. Bishnu and V. Bhattacharjee, "A New Initialization Method for KMeans Algorithm Using Quad Tree," Proc. Nat'l Conf. Methods and Models in Computing (NCM2C), pp. 73-81, 2008.
8. M. Laszlo and S. Mukherjee, "A Genetic Algorithm Using Hyper-Quadtrees for Low-Dimensional K-Means Clustering," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 28, no. 4, pp. 533-543, Apr. 2006.
9. D. Pelleg and A. Moore, "X-Means: Extending K-Means with Efficient Estimation of the Number of Cluster," Proc. 17th Int'l Conf. Machine Learning, pp. 727-734, 2000.
10. C.Nagarajan and M.Madheswaran - 'Stability Analysis of Series Parallel Resonant Converter with Fuzzy Logic Controller Using State Space Techniques'- Taylor &Francis, Electric Power Components and Systems, Vol.39 (8), pp.780-793, May 2011. DOI: 10.1080/15325008.2010.541746
11. C.Nagarajan and M.Madheswaran - 'Experimental verification and stability state space analysis of CLL-T Series Parallel Resonant Converter' - Journal of Electrical Engineering, Vol.63 (6), pp.365-372, Dec.2012. DOI: 10.2478/v10187-012-0054-2
12. C.Nagarajan and M.Madheswaran - 'Performance Analysis of LCL-T Resonant Converter with Fuzzy/PID Using State Space Analysis'- Springer, Electrical Engineering, Vol.93 (3), pp.167-178, September 2011. DOI 10.1007/s00202-011-0203-9
13. S.Tamilselvi, R.Prakash, C.Nagarajan,"Solar System Integrated Smart Grid Utilizing Hybrid Coot-Genetic Algorithm Optimized ANN Controller" Iranian Journal Of Science And Technology-Transactions Of Electrical Engineering, DOI10.1007/s40998-025-00917-z,2025
14. S.Tamilselvi, R.Prakash, C.Nagarajan," Adaptive sliding mode control of multilevel grid-connected inverters using reinforcement learning for enhanced LVRT performance" Electric Power Systems Research 253 (2026) 112428, doi.org/10.1016/j.epsr.2025.112428
15. S.Thirunavukkarasu, C. Nagarajan, 2024, "Performance Investigation on OCF and SCF study in BLDC machine using FTANN Controller," Journal of Electrical Engineering And Technology, Volume 20, pages 2675–2688, (2025), doi.org/10.1007/s42835-024-02126-w
16. C. Nagarajan, M.Madheswaran and D.Ramasubramanian- 'Development of DSP based Robust Control Method for General Resonant Converter Topologies using Transfer Function Model'- Acta Electrotechnica et Informatica Journal , Vol.13 (2), pp.18-31,April-June.2013, DOI: 10.2478/aeei-2013-0025.

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 8, Issue 2, March - April 2026 |

DOI:10.15662/IJEETR.2026.0802302

17. C.Nagarajan and M.Madheswaran - 'DSP Based Fuzzy Controller for Series Parallel Resonant converter'- Springer, Frontiers of Electrical and Electronic Engineering, Vol. 7(4), pp. 438-446, Dec.12. DOI 10.1007/s11460-012-0212-0.
18. C.Nagarajan and M.Madheswaran - 'Experimental Study and steady state stability analysis of CLL-T Series Parallel Resonant Converter with Fuzzy controller using State Space Analysis'- Iranian Journal of Electrical & Electronic Engineering, Vol.8 (3), pp.259-267, September 2012.
19. C.Nagarajan and M.Madheswaran, "Analysis and Simulation of LCL Series Resonant Full Bridge Converter Using PWM Technique with Load Independent Operation" has been presented in ICTES'08, a IEEE / IET International Conference organized by M.G.R.University, Chennai.Vol.no.1, pp.190-195, Dec.2007
20. Suganthi Mullainathan, Ramesh Natarajan, "An SPSS and CNN modelling based quality assessment using ceramic materials and membrane filtration techniques", Revista Materia (Rio J.) Vol. 30, 2025, DOI: https://doi.org/10.1590/1517-7076-RMAT-2024-0721
21. M Suganthi, N Ramesh, "Treatment of water using natural zeolite as membrane filter", Journal of Environmental Protection and Ecology, Volume 23, Issue 2, pp: 520-530,2022
22. R.A. Finkel and J.L. Bentley, "Quad Trees: A Data Structure for Retrieval on Composite Key," Acta information, vol. 4, no. 1, pp. 1-9, 1974.
23. R. Tibshirani, G. Walther, and T. Hastie, "Estimating the Number of Clusters in a Dataset via the Gap Statistic," J. Statistical Soc., vol. 63, no. 2, pp. 411-423, 2001.
24. Y. Jung, H. Park, and D.Z. Du, "A Decision Criterion for the Optimal Number of Clusters in Hierarchical Clustering," J. Global Optimization, vol. 25, pp. 91-111, 2003.
25. D. Steinley and M.J. Brusco, "Initializing K-Means Batch Clustering: A Critical Evaluation of Several Techniques," J. Classification, vol. 24, pp. 99- 121, 2007.
26. Likas, N. Vlassis, and J. Verbeek, "The Global K-means Clustering Algorithm," Pattern Recognition, vol. 36, pp. 451-461, 2003.
27. **Anand, L. (2023).** An Intelligent AI and ML–Driven Cloud Security Framework for Financial Workflows and Wastewater Analytics. *International Journal of Humanities and Information Technology, 5(02), 87-94.*
28. **Soundappan, S. J. (2020).** Big Data Analytics in Healthcare: Applications for Pandemic Forecasting. *International Journal of Advanced Research in Computer Science & Technology, 3(1), 2248-2253.*
29. **Rajasekar, M. (2024).** Real-Time Predictive DevOps Intelligence for Risk-Aware Digital Business Processes in Cloud and SAP Ecosystems. *International Journal of Advanced Research in Computer Science & Technology, 7(4), 10713-10718.*
30. **Poornima, G., & Anand, L. (2024, May). Novel AI Multimodal Approach for Combating Against Pulmonary Carcinoma. In 2024 5th International Conference for Emerging Technology (INCET) (pp. 1-6). IEEE.**
31. **Prabha, P. S., & Rengarajan, A. (2025).** Adaptive Cloud Resource Allocation Using Attention-Driven Deep Reinforcement Learning. *Engineering, Technology & Applied Science Research, 15(6), 29334-29340.*
32. **Jagadeesh, S., & Sugumar, R. (2017).** A Comparative study on Artificial Bee Colony with modified ABC algorithm. *European Journal of Applied Sciences, 9(5), 243-248.*
33. **Varma, K. K., & Anand, L. (2025, March). Deep Learning Driven Proactive Auto Scaler for High-Quality Cloud Services. In International Conference on Computing and Communication Systems for Industrial Applications (pp. 329-338). Singapore: Springer Nature Singapore.**
34. **Kumar, S. A., & Anand, L. (2025). A Novel EEG-Based Deep Learning Framework for Enhancing Communication in Locked-In Syndrome Using P300 Speller and Attention Mechanisms. KSII TRANSACTIONS ON INTERNET AND INFORMATION SYSTEMS, 19(11), 3841-3855.**
35. **Poornima, G., & Anand, L. (2025). Medical image fusion model using CT and MRI images based on dual scale weighted fusion based residual attention network with encoder-decoder architecture. Biomedical Signal Processing and Control, 108, 107932.**
36. **Archana, R., & Anand, L. (2025). Residual u-net with Self-Attention based deep convolutional adaptive capsule network for liver cancer segmentation and classification. Biomedical Signal Processing and Control, 105, 107665.Kumar, S. A., & Anand, L. (2025).** A Novel EEG-Based Deep Learning Framework for Enhancing Communication in Locked-In Syndrome Using P300 Speller and Attention Mechanisms. *KSII Transactions on Internet and Information Systems, 19(11), 3841-3855.*
37. **Rengarajan, A. (2025).** Cloud-Based AI-Driven Threat Detection Framework for Smart Grid Cybersecurity. *International Journal of Future Innovative Science and Technology, 8(6), 16065.*
38. **Murugeshwari, B., Sudharson, K., Panimalar, S. P., Shanmugapriya, M., & Abinaya, M. (2020).** SAFE–Secure Authentication in Federated Environment using CEG Key code.

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 8, Issue 2, March - April 2026 |

DOI:10.15662/IJEETR.2026.0802302

39. **Raj A. A., & Sugumar, R. (2023).** Early Detection of COVID-19 with Impact on Cardiovascular Complications using CNN Utilising Pre-Processed Chest X-Ray Images. *2023 International Conference on Applied Intelligence and Sustainable Computing (ICAISC), IEEE.*
40. **Jagadeesh, S., & Sugumar, R. (2017). A Comparative study on Artificial Bee Colony with modified ABC algorithm. European Journal of Applied Sciences, 9(5), 243-248.**
41. **Selvi, G. V., Anbarasan, A. B., Murthy, B. A., & Prabavathy, S. (2023).** An Application Oriented Integrated Unequal Clustering Algorithm for Wireless Sensor Network. In *Underwater Vehicle Control and Communication Systems Based on Machine Learning Techniques* (pp. 140-154). CRC Press.
42. **Sruthi, R. S., Ananya, S., & Murugeshwari, B. (2010).** Web Based Virtual Control System Laboratory and On-Line Temperature Control of Electrophoresis Equipment using LabVIEW. *International Journal of Computer Applications, 975, 8887.*