Journal of East-West Thought
ISSN: 2168-2259 (online)
(https://jetjournal.us/)
Volume 14, Issue 4 – Dec 2024
Impact Factor : 7.665
JOURNAL OF EAST-WEST THOUGHT
UGC CARE I

Early Fault Detection and Quality Prediction in Software Development: A Fuzzy C-Means Clustering Approach with Attribute Selection and Evaluation

Vishal Choudhary
Computer Science Department
cvishal.choudhary@gmail.com
Renaissance University,Indore
Dr.Nidhi Sethi
Computer Science Department
Renaissance University Indore

Abstract -Modern systems heavily rely on software-based systems, where software quality and reliability have emerged as significant concerns during the software development process. Developing software without any defects have proven challenging, as faults are inherent in software modules and can lead to failures in executable projects. Detecting fault-prone software components at an early stage becomes crucial for verification experts to focus their efforts effectively. This research proposes a novel approach using Fuzzy C-Means clustering for software fault and quality prediction. The approach employs attribute selection to identify crucial attributes and reduce their number, followed by attribute evaluation to assign weights. Through a ranking process, the most significant attributes are identified. Subsequently, Fuzzy C-Means clustering is utilized to group data points, where each cluster maintains a centroid value. A metric threshold is employed to determine cluster quality, with clusters having centroid values exceeding the threshold labeled as faulty, while those below the threshold are labeled as non-faulty. This methodology contributes to early fault detection and quality prediction in software development.

Keywords-Deep neural network, Software fault detection, Feature selection, fuzzy cluster

I INTRODUCTION

Software quality prediction involves using various techniques and models to forecast the potential quality attributes and defects of a software system during its development lifecycle. By analyzing historical data, metrics, and relevant factors, these methods aim to estimate the likelihood of defects, reliability, performance, and other quality characteristics. Leveraging statistical analysis, machine learning algorithms, and data mining approaches, software quality prediction provides valuable insights to guide decision-making, allocate resources effectively, and implement proactive measures to improve software quality, enhance user satisfaction, and mitigate risks before the software is deployed or release Software Faults refer to omissions, malfunctions, and errors in the software that cause unpredictable results. Faults are the essential property of the organization. They appear in intend and manufacture or the external situation. Software errors are indoctrination errors or they cause diverse presentations than expected. Most errors come from source code or design, some of which come from incorrect code generated by the compiler. For software developers or customers, software errors are a dangerous problem. Software errors will not only reduce software quality and enlarge costs but also delay progress. A software error prediction is suggested to fix this type of error. SDP can effectively improve the efficiency of software testing and manage resource allocation. To

Journal of East-West Thought
ISSN: 2168-2259 (online)
(https://jetjournal.us/)
Volume 14, Issue 4 – Dec 2024
Impact Factor : 7.665
JOURNAL OF EAST-WEST THOUGHT
UGC CARE I

expand high-quality software, software errors must be perceived or accurate in the early stages of SDLC.[1-2]

Software Metrics

The extensive examination has also been accepted to calculate insufficiency in a section through software metrics. Software metrics is a quantitative measure that is used to charge the progress of software. Three parameters are used or calculated as depict in Figure 1

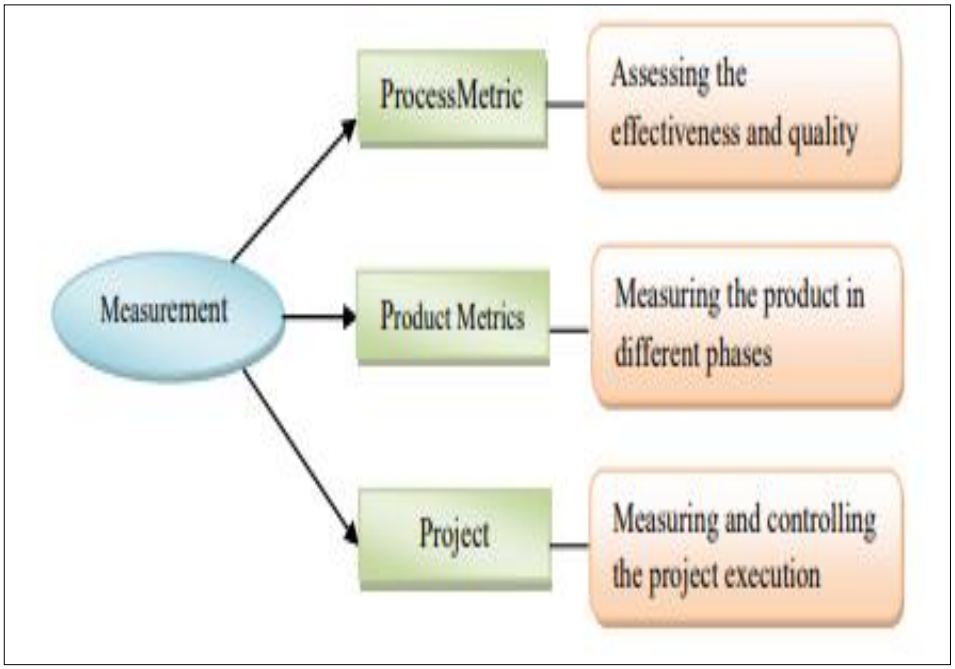

Fig.1 Software Metrics

Software metrics are quantifiable measurements used to assess various aspects of a software system's quality, complexity, efficiency, and other characteristics. These metrics provide insights into the software development process, help in tracking progress, and aid in making informed decisions.[3] There are various types of software metrics that address different aspects of software development:

Size Metrics:

Lines of Code (LOC): Measures the number of lines in the source code. Function Points (FP): Measures the functionality delivered by a software system based on user interactions.

Complexity Metrics:

- **Cyclomatic Complexity:** Measures the number of linearly independent paths through a program, indicating its structural complexity.
- **Depth of Inheritance Tree (DIT):** Measures the depth of the class inheritance hierarchy.
- **Coupling Metrics**: Measure the degree of interdependence between software modules.

Journal of East-West Thought
ISSN: 2168-2259 (online)
(https://jetjournal.us/)
Volume 14, Issue 4 – Dec 2024
Impact Factor : 7.665
JOURNAL OF EAST-WEST THOUGHT
UGC CARE I

Quality Metrics:

- **Code Duplication:** Measures the extent of duplicated code in a software system.
- **Code Coverage**: Measures the percentage of code that is executed by tests.
- **Code Smells:** Identifies potential design issues or poor coding practices.

Efficiency Metrics:

- **Execution Time**: Measures the time taken by a program to execute.
- **Memory Usage**: Measures the amount of memory consumed by a program during execution.

Maintainability Metrics:

- **Maintainability Index:** Measures the ease of maintaining the codebase.
- **Code Churn:** Measures the frequency of code changes.

Testing Metrics:

- **Defect Density:** Measures the number of defects per unit of code.
- **Test Case Coverage:** Measures the percentage of code covered by test cases.

Agile Metrics:

- **Velocity**: Measures the amount of work completed by a development team in iteration.
- **Lead Time:** Measures the time taken from a work item's creation to its completion.

Security Metrics:

- **Vulnerability Count**: Measures the number of known security vulnerabilities in a system.
- **Security Testing Coverage:** Measures the extent of security testing coverage.

Fuzzy K-Means Clustering

Fuzzy C-Means (FCM) algorithm is a clustering technique that assigns data points to clusters based on their membership degrees, allowing for partial memberships and accommodating data points that may belong to multiple clusters to varying degrees. It works by iteratively updating cluster centroids and membership values. In each iteration, data points are assigned membership values for each cluster using a fuzzy membership function that considers the Euclidean distance between data points and cluster centroids, while incorporating a fuzziness parameter 'm' to control the degree of fuzziness. Cluster centroids are then updated by recalculating their positions based on the weighted average of data points using their membership values. This iterative process continues until convergence, yielding clusters with associated membership values that indicate the likelihood of data points belonging to each cluster. FCM has wide applications in data clustering, image segmentation, and pattern recognition, where traditional hard clustering methods may not capture the nuances of data relationships.[4]

Journal of East-West Thought
ISSN: 2168-2259 (online)
(https://jetjournal.us/)
Volume 14, Issue 4 – Dec 2024
Impact Factor : 7.665
JOURNAL OF EAST-WEST THOUGHT
UGC CARE I

Fuzzy K-Means clustering is an extension of the traditional K-Means clustering algorithm that allows data points to belong to multiple clusters with varying degrees of membership. In traditional K-Means, each data point is assigned exclusively to one cluster. In Fuzzy K-Means, the degree of membership of a data point in a cluster is represented by a membership value between 0 and 1, indicating the likelihood that the data point belongs to that cluster.[5-7]

Fuzzy K-Means algorithm Initialization: Randomly initialize cluster centroids for each of the 'k' clusters.

Membership Calculation: For each data point, calculate its membership values for all clusters based on distances to cluster centroids. The membership values are calculated using a membership function, often the Fuzzy C-Means membership function.

The membership value for a data point 'i' and cluster 'j' is calculated as:

u_ij = 1 / ∑(distance(data point i to cluster j) / distance(data point i to all clusters))^2/(m-1)

'm' is a fuzziness parameter that determines the degree of fuzziness in the membership values. Higher values of 'm' make the memberships closer to 1, allowing data points to belong to multiple clusters more evenly.

Update Centroids: Calculate new cluster centroids based on the weighted average of all data points using their membership values as weights.

Repeat Membership and Centroid Updates: Iterate the membership calculation and centroid update steps until convergence or a maximum number of iterations is reached.

Cluster Assignment: Once the algorithm converges, each data point will have membership values for all clusters. To assign data points to clusters, choose the cluster with the highest membership value for each data point. Fuzzy K-Means aims to capture the uncertainty in cluster assignments, which is especially useful when data points have ambiguous memberships to different clusters. This can be applicable in scenarios where objects can belong to multiple categories or when the boundaries between clusters are not well-defined.[8-10]

Journal of East-West Thought
ISSN: 2168-2259 (online)
(https://jetjournal.us/)
Volume 14, Issue 4 – Dec 2024
Impact Factor : 7.665
JOURNAL OF EAST-WEST THOUGHT
UGC CARE I

II LITERATURE REVIEW

[11] In order to quickly and effectively respond to a newly received criminal case, information regarding the type and severity of the case is crucial for authorities. This paper designs and develops a crime type and risk level prediction technique based on machine learning technology and verifies its performance. The designed technology can predict crime type and crime risk level using a text-based criminal case summary, which is criminal case receipt data. For the text-based criminal case summary data, the KICS data format is considered, which is actual policing data that contains information about criminal cases. For the crime type, 21 representative types of crimes are considered; therefore, the system can predict one of 21 types of crime for each criminal case. Furthermore, to predict the crime risk level, we developed a crime risk calculation formula. The developed formula calculates the crime risk level and outputs the risk score in numerical terms considering the severity and damage level of the criminal case. To predict the crime type and crime risk score, both DNN and CNN-based prediction models were designed and developed. The performance evaluation section shows that, in the case of crime type prediction, the proposed prediction models can achieve better performance than traditional classification algorithms such as naïve Bayes and SVM. The performance of the CNN-based crime type prediction model is about 7% and 8% better than those of the SVM algorithm and the naïve Bayes algorithm, respectively. The performance of the designed technology was comprehensively analyzed and verified through various performance measurement parameters. It is also developed in the form of a software platform with a GUI, allowing field personnel (e.g. police officers) to intuitively identify the type of criminal case and the level of risk from a text-based criminal case summary upon receipt of a new criminal case.

[12] Categorizing the level of software risk components is very important for software developers. This categorization allows the developers to increase software availability, security, and provide better project management process. This research proposes a novel approach risk estimation system that aims to help software internal stakeholders to evaluate the currently existing software risk by predicting a quantitative software risk value. This risk value is estimated using the earlier software bugs reports based on a comparison between current and upcoming bug-fix time, duplicated bugs records, and the software component priority level. The risk value is retrieved by using a machine learning on a Mozilla Core dataset (Networking: HTTP software component) using Tensor flow tool to predict a risk level value for specific software bugs. The total risk results ranged from 27.4% to 84% with maximum bug-fix time prediction accuracy of 35%. Also, the result showed a strong relationship for the risk values obtained from the bug-fix time prediction and showed a low relationship with the risk values from the duplicated bug records.

III PROPOSED SYSTEM

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.

Journal of East-West Thought
ISSN: 2168-2259 (online)
(https://jetjournal.us/)
Volume 14, Issue 4 – Dec 2024
Impact Factor : 7.665
JOURNAL OF EAST-WEST THOUGHT
UGC CARE I

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 get the reduced number of attributes. This reduced attributes are used to clustering approach. a database of n objects, a partition 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 no-fault.

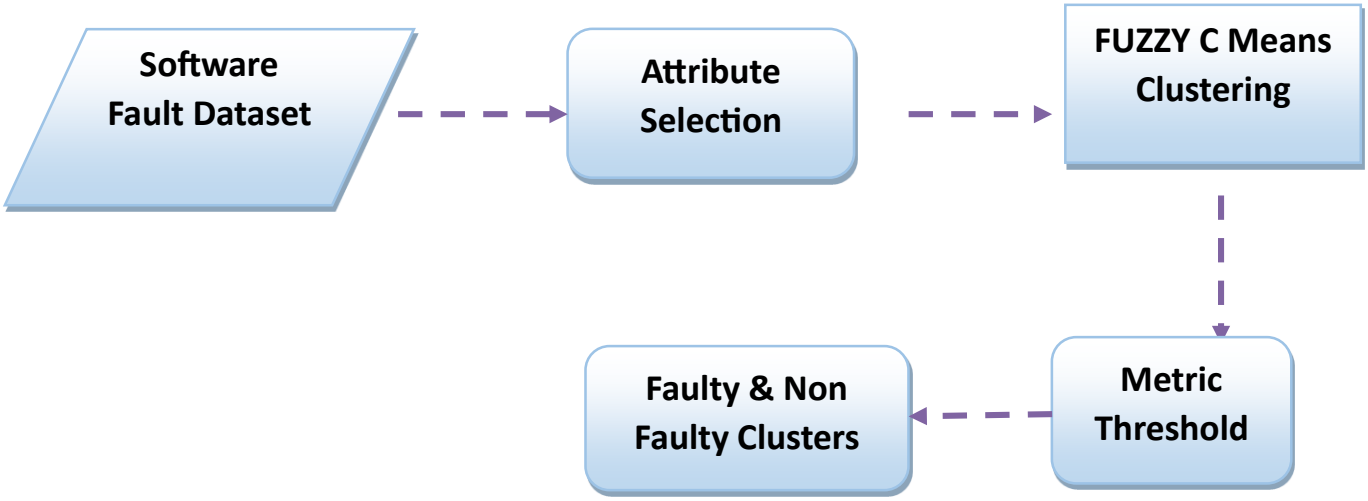

Fig.2 System Architecture

Modules

- Fault Dataset
- Attribute Selection
- FUZZY C-Means clustering
- Metric Threshold
- Detect Fault

In proposed system, present an f-fuzzy c clustering technique for calculation of software quality. First, we select the input data and apply a character selection technique. Attribute selection technology

Input Dataset Selection: selecting the input dataset, which likely contains various attributes (features) related to software modules. This is the dataset containing information about the modules, their attributes (features), and their corresponding labels indicating whether they are faulty or not. This dataset is crucial for training and evaluating fault prediction model.

Journal of East-West Thought
ISSN: 2168-2259 (online)
(https://jetjournal.us/)
Volume 14, Issue 4 – Dec 2024
Impact Factor : 7.665
JOURNAL OF EAST-WEST THOUGHT
UGC CARE I

Attribute Selection Technique: To reduce the number of attributes and focus on the most relevant ones, apply an attribute selection technique. This is important to enhance the efficiency and effectiveness of the subsequent steps. In this step, identify and select the most relevant attributes (features) from dataset. This process helps streamline the analysis by focusing on the attributes that are most likely to contribute to the detection of faults.

Attribute Evaluation and Weighting: Attribute Evaluation is used to assess the importance of each attribute in predicting software faults. The attributes are assigned weights based on their significance in fault prediction.

Ranking Method: After attribute evaluation, a ranking method is used to sort the attributes based on their weights, ensuring that the most important attributes are considered for further analysis.

Clustering Approach: Fuzzy C-Means is a clustering algorithm that groups similar data points into clusters while allowing each data point to have a degree of membership in multiple clusters. This approach can help identify patterns or groups within dataset, which can aid in understanding the distribution of potential faults.

Using the reduced set of attributes obtained from the ranking, apply the Fuzzy C-Means clustering algorithm to group the software modules into clusters. Fuzzy C-Means is a partitional clustering approach that allows data points to belong to multiple clusters with varying degrees of membership.

Cluster Centroids: After clustering, each cluster has a centroid that represents the center of the cluster in the feature space.

Metric Threshold Definition: establish a metric threshold, which is a value used to determine whether a cluster should be labeled as "faulty" or "non-faulty." This threshold is likely based on specific metrics related to software quality. A metric threshold is a predetermined value used to classify clusters or modules as "faulty" or "non-faulty." This threshold is based on certain metrics or features that are indicative of faults. Modules whose metric values exceed the threshold may be considered faulty.

Labeling Clusters: For each cluster, compare the metric values of its centroid with the defined metric threshold. If the metric value exceeds the threshold, the cluster is labeled as "faulty"; otherwise, it's labeled as "non-faulty."

Modules: These refer to the individual components or units of software that are analyzing for potential faults. These could be source code files, functions, classes, or other structural elements of the software.

Collect Software Fault Dataset (AR3, AR4, and AR5): begin by collecting a comprehensive software fault dataset that spans different versions, denoted as AR3, AR4, and AR5. These versions likely represent distinct stages of software development, capturing changes and improvements over time. The dataset includes information about software modules, their attributes, and their corresponding fault labels.[13-15]

Journal of East-West Thought
ISSN: 2168-2259 (online)
(https://jetjournal.us/)
Volume 14, Issue 4 – Dec 2024
Impact Factor : 7.665
JOURNAL OF EAST-WEST THOUGHT
UGC CARE I

Read the Data and Store it in the Database: Once gathered the fault dataset, we read and preprocess the data. This may involve cleaning the data, dealing with missing values, and ensuring the dataset is properly structured for analysis. Furthermore, we store this cleaned and organized data into a database. Storing the data in a database facilitates efficient retrieval and manipulation for subsequent stages of our approach.

Attribute Selection: Attributes are the various features or metrics associated with software modules that may contribute to fault prediction. In this step, we extract the names of all attributes present in the dataset. These attributes could encompass a range of software-related metrics, such as Lines of Code (LoC), Cyclomatic Complexity (CC), and more. This attribute information is essential for the subsequent process of attribute selection.

Software Requirements: The software requirements for the proposed system encompass an operating system such as Windows XP, utilizing Core Java version JDK 1.5 as the programming language, and the NetBeans IDE 6.9.1 for development. [16-18]

IV SIMULATION RESULT

The software indicators and error data that belong to the previous software version are used to construct a software error calculation model for the next software version. In some cases, though, the last error data does not exist. In other words, when the error-tag for the module is not available, it is a difficult task that often occurs in software manufacturing to predict a trend towards a program module. It is necessary to develop some methods to construct prediction models for software errors based on unsupervised learning, which will help expect failure tendency of program modules when the module error mark is not found. One of the processes is to use cluster techniques. Unsupervised methods such as clusters can predict errors in software modules, especially when error tags are not available. This research proposes a method for predicting software errors based on fuzzy c-means clustering for this challenging problem.

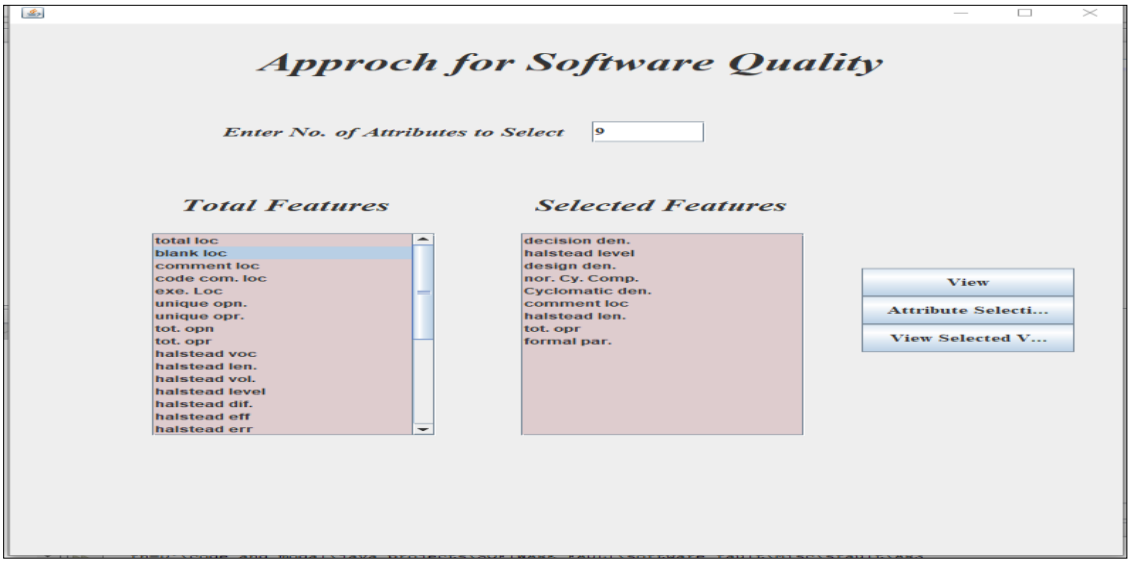

Fig.3 Number of features and Number of attributes

Journal of East-West Thought
ISSN: 2168-2259 (online)
(https://jetjournal.us/)
Volume 14, Issue 4 – Dec 2024
Impact Factor : 7.665
JOURNAL OF EAST-WEST THOUGHT
UGC CARE I

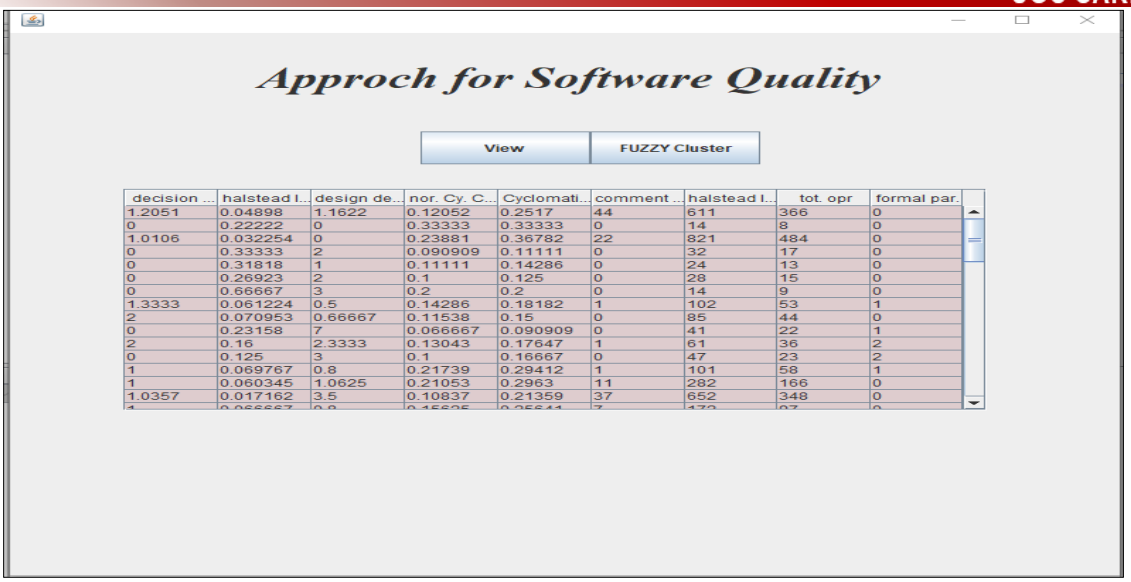

Fig.4 fuzzy cluster approach

FUZZY C-Means clustering

• A cluster is a collection of objects that are "similar" between objects and that are "different" from things in other groups.

• Fuzzy c-mean (FCM) is a clustering method that allows a single data set for two or more groups.

• Add the data first and then execute the fuzzy c-mean algorithm.

• After clustering, we separate the cluster data from the centroid cluster.

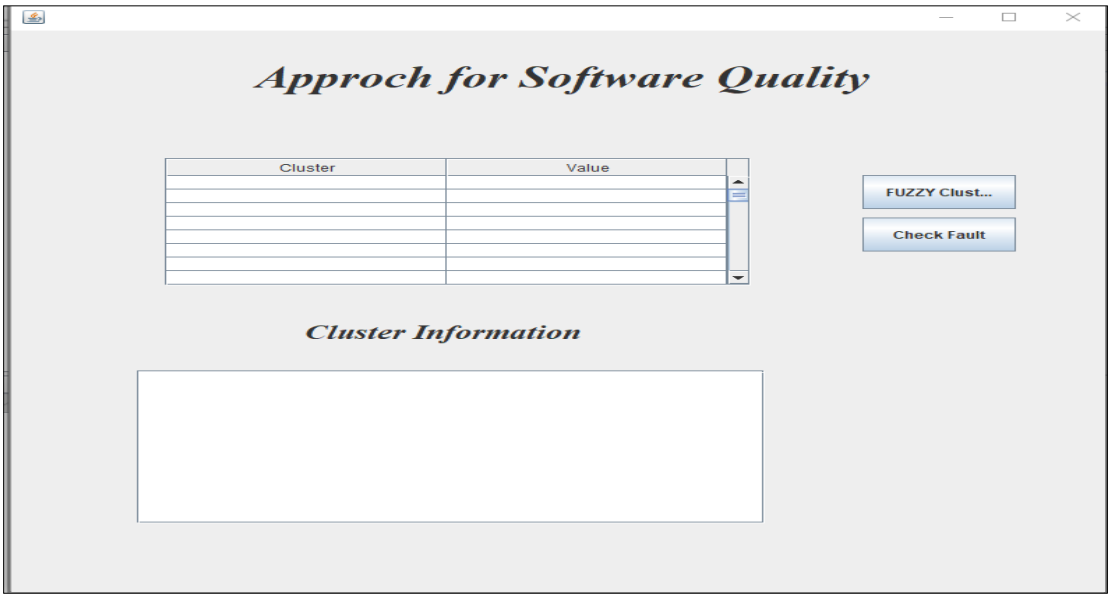

Fig.5 fuzzy cluster information for checking fault

Journal of East-West Thought
ISSN: 2168-2259 (online)
(https://jetjournal.us/)
Volume 14, Issue 4 – Dec 2024
Impact Factor : 7.665
JOURNAL OF EAST-WEST THOUGHT
UGC CARE I

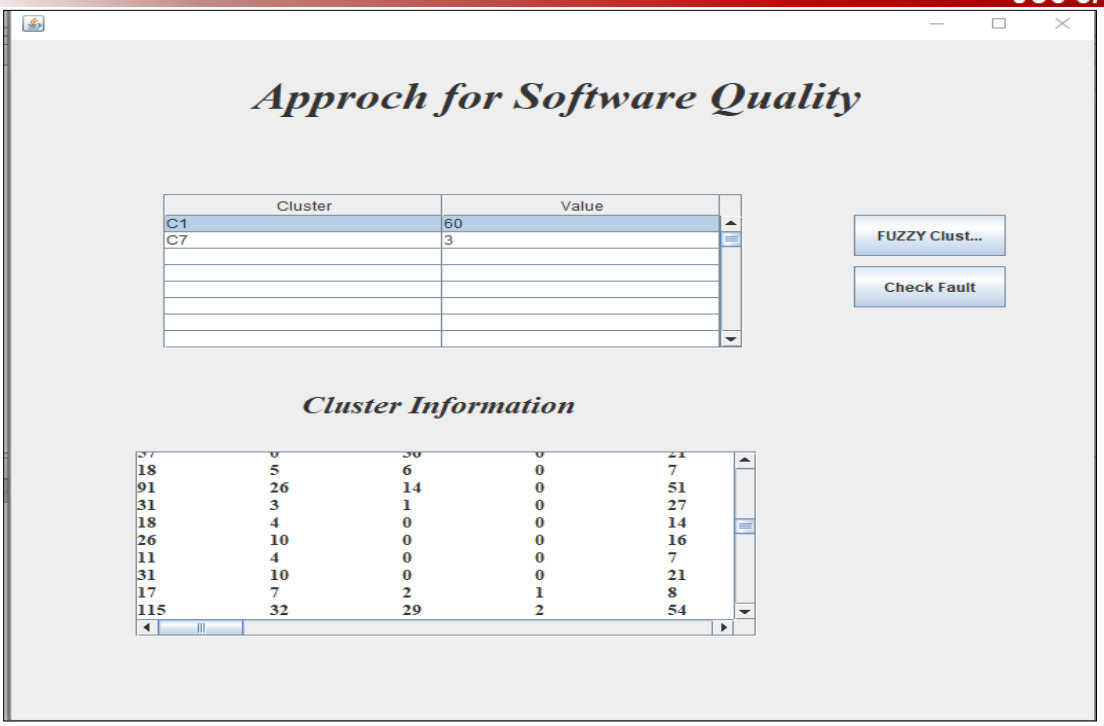

Fig.6 showing cluster information

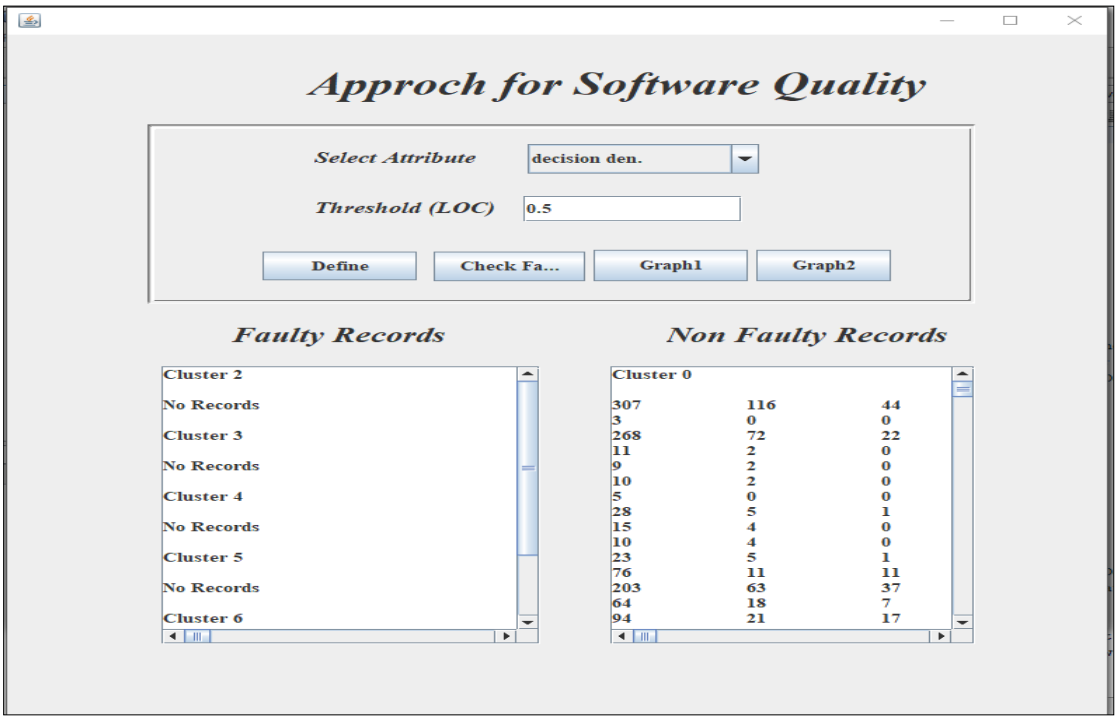

Fig.7 faulty and non-faulty records classified.

Journal of East-West Thought
ISSN: 2168-2259 (online)
(https://jetjournal.us/)
Volume 14, Issue 4 – Dec 2024
Impact Factor : 7.665
JOURNAL OF EAST-WEST THOUGHT
UGC CARE I

Metric Threshold

Determine acceptable metrics thresholds using some parameters.
The Parameters are,

- Lines of Code (LoC),
- Cyclomatic Complexity (CC),
- Unique Operator (UOp),
- Unique Operand (UOpnd),
- Total Operator (TOp),
- Total Operand (TOpnd).

Finally, we have threshold vector [LoC, CC, UOp, UOpnd, TOp, TOpnd]

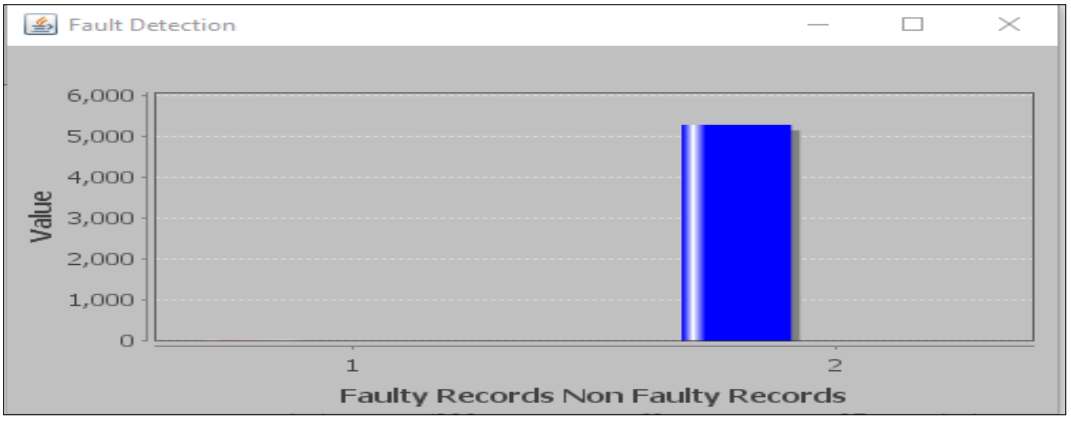

Fig.8 faulty record showing

Detect Fault

When the clustering process is complete, each cluster will hold a data point.

Take the metric Threshold for each cluster.

If the cluster's metric value's data point is larger than the Threshold, the group is marked as false, or else it is marked as not faulty.

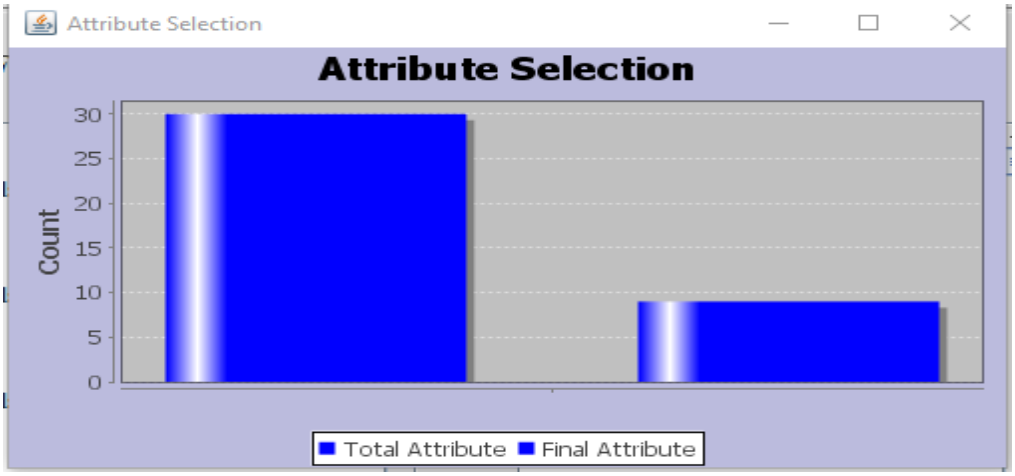

Fig.9 total number of attributes showing

Journal of East-West Thought
ISSN: 2168-2259 (online)
(https://jetjournal.us/)
Volume 14, Issue 4 – Dec 2024
Impact Factor : 7.665
JOURNAL OF EAST-WEST THOUGHT
UGC CARE I

Attribute Selection

- Receive all attributes in the data
- The total data contains 30 attributes.
- Character selection is a technique for selecting a subset of related tasks to construct a reliable learning model.
- include in the process all the properties, which require a lot of time to load and increase the load.
- Therefore, we reduced the number of traits, and we considered self-esteem traits.
- Use "trait assessment" to calculate trait relationships.
- use Weka tools to select attributes and numbers.

V CONCLUSION

In this study, we introduced an innovative approach for software fault prediction when prior fault data is unavailable. The proposed method combines the power of Fuzzy C-Means clustering with a Metrics Threshold technique, presenting a comprehensive solution to identify potential software faults. A critical step in our approach is the Attribute Selection method, where we strategically select a subset of attributes with significant relevance. By leveraging Attribute Evaluation techniques and employing the Weka tool's ranker, we optimize the process, reducing complexity and enhancing efficiency.The compassion of our strategy lies in the FUZZY C-Means Clustering algorithm. This method not only clusters data points based on similarity but also accounts for varying degrees of membership in multiple clusters. After clustering, each cluster is characterized by its own centroid value, representing a central point within the cluster. Central to our approach is the concept of Metric Thresholds. By defining acceptable threshold values, utilizing parameters such as Lines of Code (LoC), Cyclomatic Complexity (CC), Unique Operator (UOp), Unique Operand (UOpnd), Total Operator (TOp), and Total Operand (TOpnd), we establish a metric-based criterion for fault detection. In conclusion, our novel approach tackles the challenge of software fault prediction without prior fault data. By adeptly integrating Attribute Selection, Fuzzy C-Means clustering, and Metric Thresholds, we provide a robust system capable of classifying data points as faulty or non-faulty. This system not only streamlines the process but also offers valuable insights into potential software defects, enabling proactive measures and efficient resource allocation. Through these innovations, we contribute to the advancement of software quality assurance and fault prediction techniques in an environment where prior fault data is not readily available.

REFERENCES

1. Yucalar F, Ozcift A, Borandag E, Kilinc D (2020) Multiple-classifiers in software quality engineering: combining predictors to improve software fault prediction ability. Eng Sci Technol Int J 23(4):938–950
2. Bal PR, Kumar S (2020) WR-ELM: weighted regularization extreme learning machine

for imbalance learning in software fault prediction. IEEE Trans Reliab
3. Tumar I, Hassouneh Y, Turabieh H, Thaher T (2020) Enhanced binary moth flame optimization as a feature selection algorithm to predict software fault prediction. IEEE Access 8:8041–8055
4. Al O, Akour M, Alenezi M (2020) The influence of deep learning algorithms factors in software fault prediction. IEEE Access 8:63945–63960
5. Alsghaier H, Akour M (2020) Software fault prediction using particle swarm algorithm with genetic algorithm and support vector machine classifier. Softw Pract Exp 50(4):407–427
6. Xiao H, Cao M, Peng R (2020) Artificial neural network based software fault detection and correction prediction models considering testing effort. Appl Soft Comput 94:106491
7. Myung-Sun Baek; Wonjoo Park; Jaehong Park; Kwang-Ho Jang; Yong-Tae Lee Smart Policing Technique With Crime Type and Risk Score Prediction Based on Machine Learning for Early 1.AwarenessofRiskSituationDOI: 10.1109/ACCESS.2021.3112682
8. Hussain Mahfoodh; Qasem Obediat Software Risk Estimation Through Bug Reports Analysis and Bug-fix Time Predictions DOI: 10.1109/3ICT51146.2020.9312003
9. Chuanbao Du; Wanzhi Ma; Congguang Mao General platform designed for E3 risk assessment based on software-defined radio (SDR) principles DOI: 10.1109/APEMC53576.2022.9888746
10. Mohamed Najah Mahdi; Mohamed Zabil M.H.; Azlan Yusof; Lim Kok Cheng; Muhammad Sufyian Mohd Azmi; Abdul Rahim Ahmad Design and Development of Machine Learning Technique for Software Project Risk Assessment - A Review DOI: 10.1109/ICIMU49871.2020.9243459
11. Bilal Khan; Rashid Naseem; Iftikhar Alam; Inayat Khan; Hisham Alasmary; Taj Rahman Analysis of Tree-Family Machine Learning Techniques for Risk Prediction in Software Requirements DOI: 10.1109/ACCESS.2022.3206382 Thaher T, Arman N (2020) Efficient multi-swarm binary harris hawks optimization as a feature selection approach for software fault prediction. In: 2020 11th International conference on information and communication systems (ICICS). IEEE, pp 249–254
12. Pandey SK, Mishra RB, Tripathi AK (2020) BPDET: an effective software bug prediction model using deep representation and ensemble learning techniques. Expert Syst Appl 144:113085
13. Majd A, Vahidi-Asl M, Khalilian A, Poorsarvi P, Haghighi H (2020) SLDeep: Statement-level software defect prediction using deep-learning model on static code features. Expert Syst Appl 147:113156
14. Kalsoom A, Maqsood M, Ghazanfar MA, Aadil F, Rho S (2018) A dimensionality reduction-based efficient software fault prediction using Fisher linear discriminant analysis (FLDA). J Supercomput 74(9):4568–4602
15. Bhandari GP, Gupta R (2018) Measuring the fault predictability of software using deep learning techniques with software metrics. In: 2018 5th IEEE Uttar Pradesh section international conference on electrical, electronics and computer engineering (UPCON). IEEE, pp 1–6
16. Turabieh H, Mafarja M, Li X (2019) Iterated feature selection algorithms with layered recurrent neural network for software fault prediction. Expert Syst Appl 122:27–42
17. Geng W (2018) Cognitive deep neural networks prediction method for software fault tendency module based on bound particle swarm optimization. Cogn Syst Res 52:12–

20
18. Manjula C, Florence L (2019) Deep neural network based hybrid approach for software defect prediction using software metrics. Clust Comput 22(4):9847–9863
19. Juneja K (2019) A fuzzy-filtered neuro-fuzzy framework for software fault prediction for inter-version and inter-project evaluation. Appl Soft Comput 77:696–713
20. Kaur R, Sharma S (2018) An ANN based approach for software fault prediction using object oriented metrics. In: International conference on advanced informatics for computing research. Springer, Singapore, pp 341–354