OPEN ACCESS

ISAR Journal of Science and Technology

Volume 3, Issue 2, 2025, Page: 16-22

Abbriviate Title- ISAR J Sci Tech

ISSN *(Online)*- 2584-2056

https://isarpublisher.com/journal/isarjst

OVERVIEW OF PARAMETER REDUCTION AND DECISION-MAKING ALGORITHMS FOR FUZZY SOFT SETS

Abdulmajid Babangida Umar[1*], Sani Danjuma[2], Shamsuddeen Muhammad Abubakar[3]

[1,2,3] Faculty of Computing, Northwest University, Ado Bayero House Kofar Nassarawa, Kano - Nigeria.

[3] Faculty of Computing, Federal University Dutse- Jigawa State.

***Corresponding Author**

Abdulmajid Babangida Umar

Faculty of Computing, Northwest University, Ado Bayero House Kofar Nassarawa, Kano - Nigeria.

Article History

Received: 19.11.2024
Accepted: 29.01.2025
Published: 25.02.2025

Abstract: *Fuzzy soft sets is an extension of traditional soft sets that incorporates fuzzy logic, which provides a robust framework to handle imprecision and uncertainty in decision-making systems. Within the framework, several algorithms have been developed to handle parameter reduction and decision-making challenges, with each providing unique methodologies and applications. Despite this prominence, systematic review which covers the different aspects of studies on parameter reduction and decision-making algorithms is largely lacking. Hence, this study presents a comprehensive analysis of existing algorithms for parameter reduction and decision-making in fuzzy soft sets. The paper classifies, discuss, and evaluates key methods in terms of their computational efficiency, applicability, and effectiveness in decision support. Our findings underscore the strengths and limitations of current approaches and suggest avenues for future research aimed at enhancing the efficiency and applicability of fuzzy soft set algorithms in complex decision-making scenarios.*

Keywords: Fuzzy soft sets, Parameter reduction, Decision-making algorithms, Fuzzy logic.

Cite this article:

Umar, A. B., Danjuma, S., Abubakar, S. M., (2025). OVERVIEW OF PARAMETER REDUCTION AND DECISION-MAKING ALGORITHMS FOR FUZZY SOFT SETS. *ISAR Journal of Science and Technology*, *3*(2), 16-22.

1. Introduction

In data analysis and decision-making, uncertainty and vagueness are often inherent (Abdu et al., 2024). Traditional models struggle to address these ambiguities effectively, prompting the development of a hybrid approach combining soft set theory with fuzzy logic called *fuzzy soft sets* (Denoeux, 2023). Fuzzy soft sets allow for greater flexibility and are widely used in decision support systems, where accurate data representation and optimal decision-making are crucial (Beyza EKEN, 2022). Two major challenges in the practical application of fuzzy soft sets are parameter reduction and decision-making (Hassouneh, 2022). Parameter reduction simplifies the data, removing redundancy and focusing on essential parameters, while decision-making algorithms help identify optimal solutions based on multiple criteria (He, 2023).

Molodtsov defines the soft set theory as a tuple which is associated with a set of parameters and a mapping from a parameter set onto the power set of a universal set, unlike existing mathematical theories for dealing with uncertainties such as probability theory (Kolmogorov, 1950), Fuzzy set theory (Zadeh, 1965), intuitionistic fuzzy set theory (K. T. Atanassov, 1986), Rough set theory (Pawlak, 1982), vague set (Gau & Buehrer, 1993), grey set theory (Julong, 1989). Although the soft set has parameterization tool, it requires hybridization in cases involving non-Boolean datasets which could establish larger paradigms, so that any parameter can be chosen. This explain the decision-making process and compose the procedure more proficient from available data. A major advantage of fuzzy soft set theory is its lack of requirement for additional data information (e.g. the probability in statistic or possibility value in fuzzy set theory).

Parameter reduction and decision-making problems in soft set theory are interesting areas been explored by researchers in recent times. However, a review that summarizes advances in the applications of soft set theory in parameter reduction is scarce in the literature.

2. Parameter Reduction in Fuzzy Soft Sets

Parameter reduction aims to eliminate redundant or insignificant parameters, thus simplifying the decision-making process while preserving the quality of the original data (Huang, 2022). Key algorithms in this area include:

2.1. Greedy Algorithms for Parameter Reduction

Greedy algorithms are a class of algorithms that make locally optimal choices at each stage with the hope of finding a global

optimum (Mesquita, 2022). In the context of parameter reduction, greedy algorithms play a crucial role in simplifying models, improving computational efficiency, and enhancing interpretability without significantly compromising performance. Parameter reduction aims to minimize the number of parameters in a system or model while retaining its essential properties or performance (GlobalNewswire, 2018). Greedy algorithms achieve this by iteratively selecting or removing parameters based on predefined criteria, such as importance, correlation, or contribution to a specific objective function (Kohavi & John, 1997). The key characteristic of greedy algorithms is their focus on short-term gains, which may or may not lead to globally optimal solutions.

In machine learning, greedy algorithms are commonly used for feature selection, where irrelevant or redundant features are removed to improve model performance (Jiarpakdee et al., 2021). Algorithms like forward selection, backward elimination, and stepwise regression are examples of greedy approaches that evaluate parameters one at a time based on metrics like accuracy, information gain, or mutual information (Guyon & Elisseeff, 2003). Greedy approaches are also used in dimensionality reduction techniques like Principal Component Analysis (PCA) and feature subset selection (Cai et Al, 2023). By iteratively removing or combining dimensions, these methods retain the most significant features while reducing computational overhead (Jolliffe, 2002). Kohavi and John (1997) demonstrated the effectiveness of greedy algorithms in feature subset selection, leading to significant improvements in classification accuracy while reducing the computational cost.

2.2. Entropy-Based Parameter Reduction

Entropy-based parameter reduction is a technique that applies concepts from information theory to minimize the complexity of data representations, particularly in systems or models requiring parameter optimization (Al-Jamimi., 2024). Entropy, a measure of uncertainty or randomness in a dataset, provides a quantitative basis for identifying and discarding irrelevant or redundant parameters. Introduced by Shannon (1948), entropy quantifies the amount of information contained in a dataset. High entropy implies greater uncertainty or variability, whereas low entropy suggests redundancy or predictability in the data. In parameter reduction, the goal is to identify parameters with low entropy that contribute minimal unique information to the system, allowing for their removal without significant loss of predictive power or performance (Tantithamthavorn, 2018).

Advancements in entropy-based methods are focused on integrating non-linear measures of dependency and developing adaptive thresholding techniques to improve the accuracy and efficiency of parameter reduction processes. Additionally, hybrid approaches that combine entropy-based methods with machine learning algorithms are gaining traction for their ability to balance computational efficiency with predictive accuracy. Figure 1 depicts the trend of research that employed entropy-based reduction approach covered from 2014 to 2024.

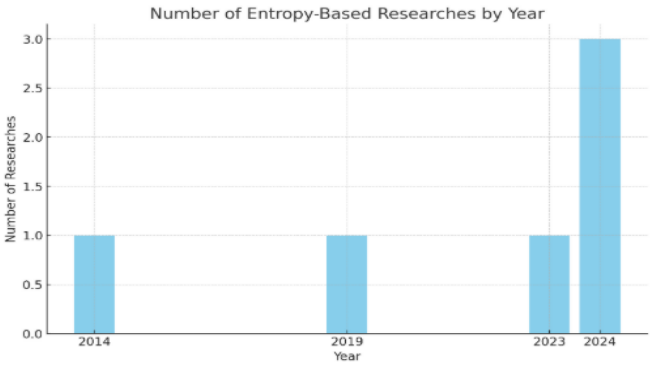

Figure 1: Entropy-based approach

The bar chart illustrating the number of entropy-based research studies conducted each year. The chart shows a spike in research in 2024, indicating increased activity in this area during that period.

2.3. Correlation-Based Reduction

Correlation-Based Reduction (CBR) is a dimensionality reduction technique aimed at simplifying data while preserving essential relationships among variables (Khurma et al., 2021). This method leverages the statistical property of correlation to identify redundant or irrelevant features in datasets, which often impede the performance of machine learning algorithms (Ayon, 2019). By eliminating or merging highly correlated variables, CBR enhances computational efficiency, improves model interpretability, and reduces overfitting risks *(Alsaeedi & Khan, 2019)*.

Correlation quantifies the linear relationship between two variables, often measured using Pearson's correlation coefficient, Spearman's rank correlation, or Kendall's tau (Agrawal, 2020). Features with high correlation share significant information, leading to redundancy in the dataset. For instance, in software defect prediction datasets, metrics such as lines of code (LOC) and cyclomatic complexity often exhibit high correlation, making one of these metrics redundant for modeling purposes (Zhang et al., 2022).

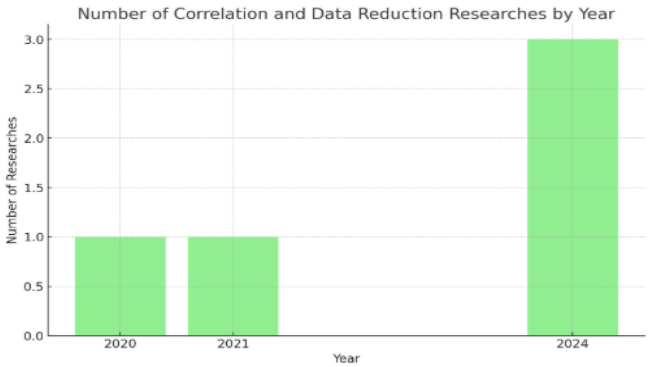

Figure 2: Number of Correlation and Data reduction on yearly Distribution

The bar chart illustrating the number of correlation and data reduction research studies conducted each year. The chart shows a noticeable increase in 2024, indicating heightened research activity in this area during that period.

Correlation-Based Reduction is a powerful tool in data preprocessing, particularly in high-dimensional datasets. Its ability to streamline features while preserving critical information makes

it indispensable in various fields, from software engineering to healthcare analytics. However, its success depends on careful implementation, including appropriate threshold selection and validation to ensure model performance remains robust.

2.4. Hybrid Reduction Methods

Hybrid reduction methods integrate multiple dimensionality reduction techniques to effectively reduce the complexity of high-dimensional data while retaining critical features *(Anju & Judith, 2023)*. These methods address the limitations of single-method approaches, which might overlook important data characteristics or fail to handle diverse data types efficiently (Parashar et al., 2022). By combining the strengths of different techniques, hybrid reduction methods improve performance in tasks such as classification, clustering, and prediction.

Hybrid methods are widely used in gene expression analysis, where datasets are often high-dimensional and sparse. Combining feature selection with PCA or LDA helps to identify biomarkers while maintaining computational efficiency (Feng, 2022). In software prediction, hybrid methods improve model accuracy by reducing redundant and irrelevant software metrics while preserving critical defect-related features (Abdu et al., 2024). For instance, a combination of mutual information for feature selection and autoencoders for feature extraction has shown promising results.

Hybrid methods are also positive in handling textual and visual data, where dimensionality is a significant challenge. Combining term frequency-inverse document frequency (TF-IDF) with deep learning-based embeddings can enhance classification accuracy (Mustaqeem et al., 2024).

3. Decision-Making in Fuzzy Soft Sets

Decision-making in fuzzy soft sets often involves scenarios where data uncertainty and vagueness necessitate systematic approaches to aggregate and evaluate the information (Denoeux, 2023). Weighted aggregation methods provide an essential mechanism in this process, offering a structured way to combine multiple criteria, each with a corresponding weight, to derive meaningful decisions. Decision-making in fuzzy soft sets focuses on identifying the best option based on selected parameters. Prominent algorithms include:

3.1. Weighted Aggregation Methods

Weighted aggregation methods are a class of techniques used to merge individual criteria evaluations into a single, comprehensive assessment (Osman et al., 2023). These methods account for the relative importance (weights) assigned to each criterion, reflecting its contribution to the overall decision. In fuzzy soft sets, weighted aggregation methods deal with fuzzy parameters, which are represented as fuzzy values characterized by a degree of membership ranging between 0 and 1. Each parameter in a fuzzy soft set is associated with a membership function that maps elements to values in [0,1] (Beyza EKEN, 2022). The membership value indicates the degree to which an element satisfies a particular criterion. Weights are assigned to parameters to reflect their relative importance in the decision-making process (Mafarja, 2023). Weight values are usually normalized such that the sum of all weights equals 1:

$$\sum_{i=1}^{n} w_i = 1$$

Where:

wi: The weight assigned to the iii-th parameter.

n: The total number of parameters.

The weights *wi* are normalized, ensuring that their sum equals 1. This reflects to their relative importance in the decision-making processes.Weighted aggregation methods rely on specific operators to combine parameter values based on their weights. Commonly used operators include:

$$A(x) = \sum_{i=1}^{n} w_i \cdot \mu_i(x)$$

Where:

wi: is the weight of the *i*-th criterion.

μi(x): is the membership value of element *x* for the *i*-th criterion

A(x): represents the aggregated value for *x*

The equation calculates a weighted sum of the membership values for each criterion, using their respective weights. Overall performance score is then calculated as the product of the criteria values raised to the power of their respective weights, capturing the multiplicative interaction among the criteria.

$$A(x) = \prod_{i=1}^{n} (\mu_i(x))^{w_i}$$

$$A(x) = \left(\prod_{i=1}^{n} \mu_i(x)^{w_i} \right)^{\frac{1}{\sum w_i}}$$

3.2. Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS)

The Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS) is a widely recognized multi-criteria decision-making (MCDM) method (Zhou Xu, 2021). It was first introduced by Hwang and Yoon in 1981 to rank alternatives based on their geometric distance from an ideal solution and a nadir (anti-ideal) solution. TOPSIS is known for its simplicity, logical approach, and ability to provide a clear ranking of alternatives, making it a popular tool in fields such as engineering, finance, healthcare, and supply chain management (Hwang & Yoon, 1981). The technique is based on the principle that the chosen alternative should have the shortest distance from the positive ideal solution (PIS) and the longest distance from the negative ideal solution (NIS).

TOPSIS is a robust and versatile MCDM tool that balances proximity to the ideal solution with separation from the worst-case scenario. Its adaptability and logical foundation make it a preferred choice for decision-making across diverse fields. However, careful consideration of criteria weights and potential biases is essential for accurate application (Watanabe, 2022).

3.3. Fuzzy Rough Set Approach

The **Fuzzy Rough Set Approach (FRSA)** is a hybrid methodology that combines the strengths of rough set theory and fuzzy logic to handle uncertainty and vagueness in data analysis (Xiao-Yuan Jing, 2022). It is particularly effective in situations where the boundaries of concepts are not sharply defined, and attributes have overlapping memberships in multiple sets. This approach has gained significant attention in fields such as decision-making, pattern recognition, data mining, and software defect prediction.

1. **Conceptual Foundations**

Rough set theory, introduced by Pawlak in the early 1980s, deals with vagueness by approximating sets using lower and upper bounds, capturing the indiscernibility of elements within dataogic, introduced by Zadeh, enhances this by addressing the imprecision of attributes through the use of membership functions (DAmbros, 2020). The n of rough sets and fuzzy logic creates a framework where fuzzy sets are used to approximate the indiscernibility relations of rough sets. This combination allows the FRSA to model uncertainty at multiple levels: indiscernibility at the structural level (via rough sets) and imprecision in attribute values (Jaechang, 2022).

2. Sentation

In FRSA, the fuzzy equivalence relation RRR replaces the classical equivalence relation used in traditional rough sets (Rahman, 2021). This relation is defined by a membership function μR(x,y)\mu_{R}(x, y)μR(x,y) that quantifies the degree of similarity between elements xxx and yyy. The lower and upper approximations of a fuzzy set FFF are given as follows:

Lower Approximation:

$$\mu_{\underline{R}(F)}(x) = \inf_{y \in U} \{\max(1 - \mu_R(x, y), \mu_F(y))\}$$

Upper Approximation

$$\mu^{\overline{R}(F)}(x) = \sup_{y \in U} \{\min(\mu_R(x, y), \mu_F(y))\}$$

FRSA is extensively applied in feature selection tasks to reduce dimensionality without losing critical information. It identifies the most relevant attributes while handling overlapping and noisy data, making it suitable for high-dimensional datasets. The approach has been used in software defect prediction to enhance the interpretability of prediction models by combining the data-driven nature of rough sets with the uncertainty modeling of fuzzy logic (Ghotra, 2019). This improves the reliability of predictions in the presence of noisy or incomplete datasets.

3.4. Multi-Attribute Decision-Making (MADM) Methods

Multi-Attribute Decision-Making (MADM) methods are a subset of decision-making techniques used to evaluate and prioritize alternatives when multiple, often conflicting, attributes or criteria are involved (Menzies, 2022). These methods are particularly useful in scenarios where decision-makers must consider various quantitative and qualitative factors to achieve an optimal or near-optimal solution. MADM methods are distinguished by the following features:

i. Alternatives: A finite number of discrete options or choices.

ii. Attributes: Multiple criteria or attributes, often with different units of measurement that must be evaluated for each alternative.

iii. Preference Modeling: Techniques to model and quantify preferences to guide decision-making.

Several MADM techniques have been developed, each with unique characteristics and applications. Simple Additive Weighting (SAW) also known as weighted linear combination, is one of the simplest MADM methods (Swapnil Shukla, 2019). It involves multiplying attribute values by their respective weights and summing them to compute a total score for each alternative. It is also suitable for straightforward decision problems where criteria are independent. Analytic Hierarchy Process (AHP) is another MADM approach that employs pairwise comparisons to establish criteria weights and rank alternatives. It combines qualitative and quantitative data, making it useful for complex problems (Balogun et al., 2021). AHP ensures consistency in comparisons by using a consistency ratio as a validation metric.

TOPSIS on the other hand ranks alternatives by their closeness to an ideal solution (maximum benefit) and distance from a nadir solution (maximum cost). It is widely used for problems requiring a balance between conflicting criteria (Jiarpakdee et al., 2021). WPM is similar to SAW but uses the product of attribute values raised to the power of their weights. It is particularly effective for multiplicative relationships between criteria (Tantithamthavorn, 2022). MAUT constructs a utility function to evaluate the overall satisfaction of decision-makers with respect to each alternative. This approach is ideal for problems requiring precise utility representation for complex attributes (Suresh Kumar et al., 2021).

Elimination and Choice Expressing Reality (ELECTRE) is a family of methods based on outranking relationships between alternatives. It uses thresholds to determine whether one alternative is preferred, indifferent, or incomparable to another. Commonly used for group decision-making (Palma et al., 2022).from a set of alternatives while providing a compromise solution. This method emphasizes proximity to an ideal solution and can handle conflicting objectives effectively (Alsaeedi & Khan, 2019).

4. Comparative Analysis of Algorithms

To better understand the trade-offs involved in parameter reduction and decision-making algorithms for fuzzy soft sets, we evaluate them across key dimensions, such as computational efficiency, accuracy, and applicability. Table 1 provides a summary of this comparison.

Table 1: Summary of Comparative Analysis

Algorithm	Computational Efficiency	Accuracy	Applicability	Notes
Greedy Parameter Reduction	High	Moderate	Large datasets	Fast but may lack global optimality
Entropy-Based Reduction	Moderate	High	Systems with probabilistic data	Accurate but computationally intensive
Correlation-Based Reduction	High	Moderate	Systems with redundant data	Good for redundancy removal
Hybrid Reduction Methods	Moderate	High	Complex decision environments	More accurate but complex
Weighted Aggregation	Moderate	High	Customizable for expert systems	Requires expert knowledge
TOPSIS	High	High	Multi-criteria fuzzy systems	Sensitive to scaling
Fuzzy Rough Set	Low	High	High-uncertainty environments	Effective but computationally heavy
MADM Methods	Moderate	High	Structured decision environments	Complex for large systems

According to the table, high computational efficiency in the greedy parameter reduction algorithms makes it suitable for large datasets. Moderate accuracy was presented due to its heuristic nature, which may not guarantee the best global solution. In terms of applicability, it is ideal for quick reductions in large datasets. Entropy-Based Reduction is a moderate efficiency is recorded in entropy-based reduction approach, as the calculations are more computationally demanding. High accuracy was recorded, making it suitable for systems with probabilistic data. This approach is most effective in applicability when dealing with scenarios where precision is crucial. For correlation-based reduction, high efficiency was recorded in the approach due to the straightforward removal of redundant data. While, Moderate accuracy since it focuses on redundancy rather than overall data significance. It is more Applicable for systems where data redundancy is a concern.

Figure 3 presents a bar chart comparing computational efficiency across different algorithms and a bar chart comparing accuracy across different algorithms.

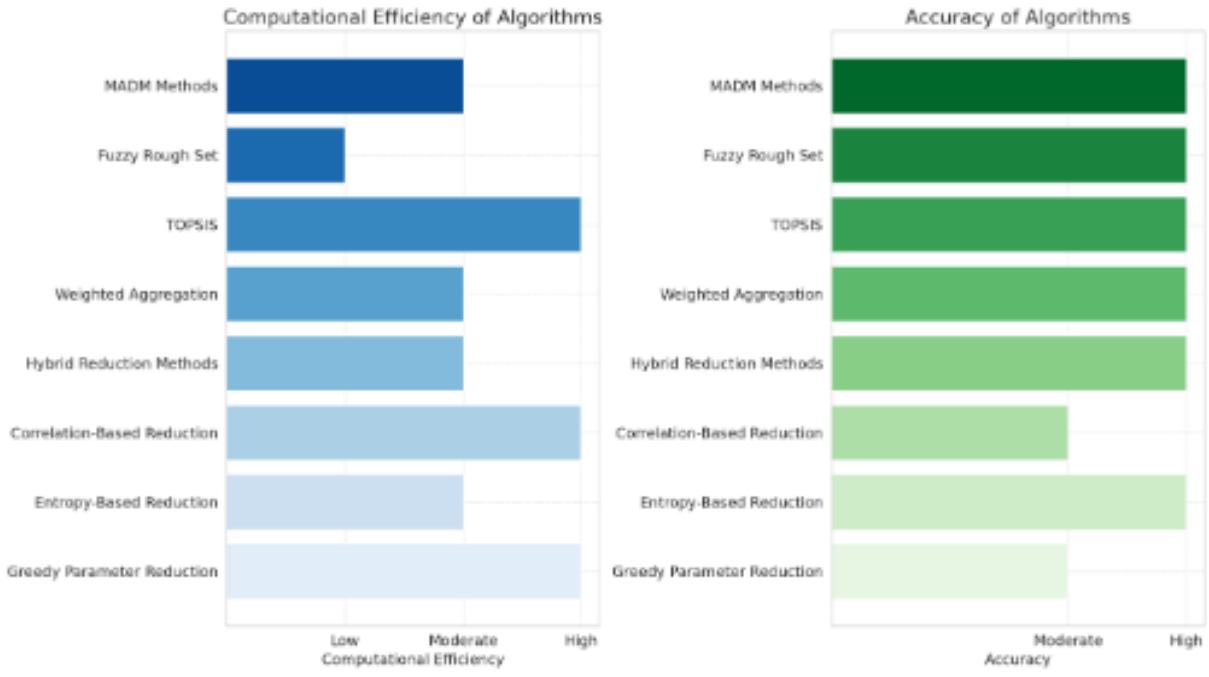

Figure 3: Computational Efficiency and Accuracy of Algorithms

The bar charts provide a visual comparison of the computational efficiency and accuracy of different data reduction algorithms. From the bar charts following has been deduced.

5. Conclusion

This paper presents a structured analysis of existing algorithms for parameter reduction and decision-making in fuzzy soft sets. While many methods provide efficient and accurate solutions, the choice of algorithm often depends on the specific context, data characteristics, and computational constraints. Our findings indicate that while parameter reduction algorithms like hybrid methods balance accuracy with efficiency, decision-making methods like TOPSIS and fuzzy rough sets excel in handling uncertainty and complex criteria. Further research in adaptive and hybrid models may lead to more efficient algorithms suitable for a broader range of applications, particularly in real-time decision support systems.

References

1. Abdu, A., Zhai, Z., Abdo, H. A., Algabri, R., Al-masni, M. A., Muhammad, M. S., & Gu, Y. H. (2024). Semantic and traditional feature fusion for software defect prediction using hybrid deep learning model. *Scientific Reports*, *14*(1). https://doi.org/10.1038/s41598-024-65639-4
2. Agrawal, T. (2020). Hyperparameter Optimization in Machine Learning: Make Your Machine Learning and Deep Learning Models More Efficient. In *Hyperparameter Optimization in Machine Learning: Make Your Machine Learning and Deep Learning Models More Efficient*. Apress Media LLC. https://doi.org/10.1007/978-1-4842-6579-6
3. Alsaeedi, A., & Khan, M. Z. (2019). Software Defect Prediction Using Supervised Machine Learning and Ensemble Techniques: A Comparative Study. *Journal of Software Engineering and Applications*, *12*(05), 85–100. https://doi.org/10.4236/jsea.2019.125007
4. Anju, A. J., & Judith, J. E. (2023). Adaptive recurrent neural network for software defect prediction with the aid of quantum theory- particle swarm optimization. *Multimedia Tools and Applications*, *82*(11), 16257–16278. https://doi.org/10.1007/s11042-022-14065-7
5. Arasteh, B., Arasteh, K., Ghaffari, A., & Ghanbarzadeh, R. (2024). A new binary chaos-based metaheuristic algorithm for software defect prediction. *Cluster Computing*. https://doi.org/10.1007/s10586-024-04486-4
6. Associate Professor, K. N., & Uthariaraj Professor, R. V. (2013). Improving Fault Prediction using ANN-PSO in Object Oriented Systems. In *International Journal of Computer Applications* (Vol. 73, Issue 3).
7. Ayon, S. I. (2019, May 1). Neural Network based Software Defect Prediction using Genetic Algorithm and Particle Swarm Optimization. *1st International Conference on Advances in Science, Engineering and Robotics Technology 2019, ICASERT 2019*. https://doi.org/10.1109/ICASERT.2019.8934642
8. Baljinder Ghotra, S. M. and A. E. Hassan. (n.d.). *Revisiting the impact of classification techniques on the performance of defect prediction models.*
9. Balogun, A. O., Basri, S., Capretz, L. F., Mahamad, S., Imam, A. A., Almomani, M. A., Adeyemo, V. E., Alazzawi, A. K., Bajeh, A. O., & Kumar, G. (2021). Software defect prediction using wrapper feature selection based on dynamic re-ranking strategy. *Symmetry*, *13*(11). https://doi.org/10.3390/sym13112166
10. Beyza EKEN, T. (2022). *SOFTWARE DEFECT PREDICTION WITH A PERSONALIZATION FOCUS AND CHALLENGES DURING DEPLOYMENT*.
11. Bibi S, Tsoumakas G, & Stamelos I. (n.d.). *Software Defect Prediction Using Regression via Classification*.
12. Chiu, N. H. (2011). Combining techniques for software quality classification: An integrated decision network approach. *Expert Systems with Applications*, *38*(4), 4618–4625. https://doi.org/10.1016/j.eswa.2010.09.136 Press.
13. Denoeux, T. (2023). Quantifying Prediction Uncertainty in Regression Using Random Fuzzy Sets: The ENNreg Model. *IEEE TRANSACTIONS ON FUZZY SYSTEMS*, *31*(10). https://doi.org/10.1007/978-3-031-17801
14. Diego PP Mesquita, L. S. R. J. P. P. G. and A. R. R. Neto. (n.d.). *Classification with reject option for software defect prediction.*
15. Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). *A fast and elitist multiobjective genetic algorithm: NSGA-II*. IEEE Transactions on Evolutionary Computation, 6(2), 182–197.
16. Foyzur Rahman, D. P. and P. Devanbu. (n.d.). *Recalling the imprecision of cross-project defect prediction.*
17. Ghotra, B., McIntosh, S., & Hassan, A. E. (2015). Revisiting the impact of classification techniques on the performance of defect prediction models. *Proceedings - International Conference on Software Engineering*, *1*, 789–800. https://doi.org/10.1109/ICSE.2015.91
18. GlobalNewswire. (2018, January 24). *Software Failures, defects and vulnarabilities*. 3–7.
19. H. A. Al-Jamimi. (n.d.). *Toward comprehensible software defect prediction models using fuzzy logic.*
20. Hassouneh, Y. et al. (n.d.). *Boosted whale optimization algorithm with natural selection operators for software fault prediction.*
21. Han, J., Kamber, M., & Pei, J. (2011). *Data mining: Concepts and techniques*. Morgan Kaufmann.
22. Han, S., Pool, J., Tran, J., & Dally, W. J. (2015). *Learning both weights and connections for efficient neural networks*. Advances in Neural Information Processing Systems, 28, 1–9.
23. Jiarpakdee, J., Tantithamthavorn, C., & Hassan, A. E. (2021). The Impact of Correlated Metrics on the Interpretation of Defect Models. *IEEE Transactions on Software Engineering*, *47*(2), 320–331. https://doi.org/10.1109/TSE.2019.2891758
24. Jin, C. (2021). Cross-project software defect prediction based on domain adaptation learning and optimization. *Expert Systems with Applications*, *171*. https://doi.org/10.1016/j.eswa.2021.114637
25. Jiang, J., Chen, L., & Gao, W. (2016). *Feature selection in intrusion detection systems using entropy-based methods*. Journal of Network and Computer Applications, 64, 125–132. https://doi.org/10.1016/j.jnca.2016.02.001
26. Kassaymeh, S., Abdullah, S., Al-Betar, M. A., & Alweshah, M. (2022). Salp swarm optimizer for modeling the software fault prediction problem. *Journal of King Saud University - Computer and Information Sciences*, *34*(6), 3365–3378. https://doi.org/10.1016/j.jksuci.2021.01.015

27. Khurma, R. A., Alsawalqah, H., Aljarah, I., Elaziz, M. A., & Damaševičius, R. (2021). An enhanced evolutionary software defect prediction method using island moth flame optimization. *Mathematics*, *9*(15). https://doi.org/10.3390/math9151722
28. Laradji, I. H., Alshayeb, M., & Ghouti, L. (2015). Software defect prediction using ensemble learning on selected features. *Information and Software Technology*, *58*, 388–402. https://doi.org/10.1016/j.infsof.2014.07.005
29. Li, Z., Li, T., Wu, Y., Yang, L., Miao, H., & Wang, D. (2021). Software Defect Prediction Based on Hybrid Swarm Intelligence and Deep Learning. *Computational Intelligence and Neuroscience*, *2021*. https://doi.org/10.1155/2021/4997459
30. Mafarja, M. et al. (n.d.-a). *Classification framework for faulty-software using enhanced exploratory whale optimizer-based feature selection scheme and random forest ensemble learning*. .
31. Mafarja, M. et al. (n.d.-b). *Classification framework for faulty-software using enhanced exploratory whale optimizer-based feature selection scheme and random forest ensemble learning. Appl.* .
32. Malhotra, R. (2018). An extensive analysis of search-based techniques for predicting defective classes. *Computers and Electrical Engineering*, *71*, 611–626. https://doi.org/10.1016/j.compeleceng.2018.08.017
33. Marco DAmbros, M. L. and R. Robbes. (n.d.). *Evaluating defect prediction approaches: a benchmark and an extensive comparison.*
34. Martin Shepperd, Q. S. Z. S. and C. Mair. (n.d.). *Data quality: Some comments on the nasa software defect datasets.*
35. Mccabe, T. J. (1976). A Complexi-ty Measure. In *IEEE TRANSACTIONS ON SOFTWARE ENGINEERING* (Issue 4).
36. Muhammad, R., Nadeem, A., & Sindhu, M. A. (2021). Vovel Metrics—novel Coupling Metrics for Improved Software Fault Prediction. *PeerJ Computer Science*, *7*, 1–27. https://doi.org/10.7717/peerj-cs.590
37. Mustaqeem, M., Mustajab, S., Alam, M., Jeribi, F., Alam, S., & Shuaib, M. (2024). A trustworthy hybrid model for transparent software defect prediction: SPAM-XAI. *PLoS ONE*, *19*(7 July). https://doi.org/10.1371/journal.pone.0307112
38. Olatunji, B. L., Olabiyisi, S. O., Oyeleye, C. A., Sanusi, B. A., Olowoye, A. O., & Ofem, O. A. (2020). Development of software defect prediction system using artificial neural network. *International Journal of Advances in Applied Sciences*, *9*(4), 284–293. https://doi.org/10.11591/ijaas.v9.i4.pp284-293
39. Osman, H., Ghafari, M., & Nierstrasz, O. (n.d.). *Hyperparameter Optimization to Improve Bug Prediction Accuracy*. https://github.com/fracpete/multisearch-weka-package
40. Palma, S. D., Di Nucci, D., Palomba, F., & Tamburri, D. A. (2022). Within-Project Defect Prediction of Infrastructure-as-Code Using Product and Process Metrics. *IEEE Transactions on Software Engineering*, *48*(6), 2086–2104. https://doi.org/10.1109/TSE.2021.3051492
41. Pandey, S. K., Mishra, R. B., & Tripathi, A. K. (2021). Machine learning based methods for software fault prediction: A survey. In *Expert Systems with Applications* (Vol. 172). Elsevier Ltd. https://doi.org/10.1016/j.eswa.2021.114595
42. Parashar, A., Kumar Goyal, R., Kaushal, S., & Kumar Sahana, S. (2022). Machine learning approach for software defect prediction using multi-core parallel computing. *Automated Software Engineering*, *29*(2). https://doi.org/10.1007/s10515-022-00340-2
43. Perera, A. (n.d.). *Using Defect Prediction to Improve the Bug Detection Capability of Search-Based Software Testing*.
44. Patel, R., et al. (2022). *Applications of correlation-based methods in image analysis*. Computer Vision Advances, 20(4), 312–328.
45. Qiao Huang, X. X. and D. Lo. (n.d.). *Revisiting supervised and unsupervised models for effort-aware just-in-time defect prediction.*
46. Rahman, A. (2019). *Software Defect Prediction Using Rich Contextualized Language Use Vectors*.
47. Rajapaksha, D., Tantithamthavorn, C., Jiarpakdee, J., Bergmeir, C., Grundy, J., & Buntine, W. (2021). *SQAPlanner: Generating Data-Informed Software Quality Improvement Plans*. http://arxiv.org/abs/2102.09687
48. Rhmann, W. (2018). Application of Hybrid Search Based Algorithms for Software Defect Prediction. *International Journal of Modern Education and Computer Science*, *10*(4), 51–62. https://doi.org/10.5815/ijmecs.2018.04.07
49. Shinya Watanabe, H. K. and K. Kaijiri. (n.d.). *Adapting a fault prediction model to allow inter language reuse.*
50. Stefan Lessmann, B. B. C. M. and S. Pietsch. (n.d.). *Benchmarking classification models for software defect prediction: A proposed 151 framework and novel findings*. .
51. Suresh Kumar, P., Behera, H. S., Nayak, J., & Naik, B. (2021). Bootstrap aggregation ensemble learning-based reliable approach for software defect prediction by using characterized code feature. *Innovations in Systems and Software Engineering*, *17*(4), 355–379. https://doi.org/10.1007/s11334-021-00399-2.
52. Zhimin He, F. S. Y. Y. M. L. and QingWang. (n.d.). *An investigation on the feasibility of cross-project defect prediction.*
53. Zhongguo nong ye da xue, & Institute of Electrical and Electronics Engineers. (n.d.). *International Conference on Big Data and Artificial Intelligence (BDAI 2018) : June 22-24, 2018, Beijing, China.*
54. Zhou Xu, J. L. X. L. and T. Zhang. (n.d.). *Cross-version defect prediction via hybrid active learning with kernel principal component analysis.*