Systematic Review of Bespoke Techniques of Software Fault Prediction: Machine Learning and Conventional Techniques

[1]Tajinder Pal Singh, [2]Dr. Tushar Patil, [3]Mr. Madhur Chauhan, [4] Aarti Rana Chauhan. [5]Mahesh Bhat, [6]Mr. Adnan Khan

[1] *Chitkara University Institute of Engineering and Technology, Chitkara University, Rajpura, Punjab, India 140401*

[2,3,5,6] *G H Raisoni College of Engineering and Management, Jalgaon, Maharashtra, India 425002*

[4]*University Institute of Architecture at Chandigarh University, NH-95 Chandigarh-Ludhiana Highway, Mohali, Punjab, India*

Abstract : Software fault prediction is the practice of creating frameworks that software professionals are able to employ in the early stages of the software development life cycle to detect troubling constructs related to modules or classes. Several techniques have been suggested in the past for this purpose, the majority of which relied on lengthy mathematical models that did not appear to be suited for the current challenge. With technological advances, artificial intelligence (AI) modeling, also known as soft computing approaches, unlike conventional models, is gaining attention in research as it requires less processing and anticipates results quickly. In this paper, the author attempts to review conventional and machine learning techniques used previously for software fault prediction. For this review, data has been collected from various reputed library, such as IEEE Xplore. The present article has analyzed the various papers published from 2018 onward on the use of various conventional and non-conventional techniques. The extensive study's analysis given in this article will be of great interest to academicians and professionals working in the field of software fault prediction in determining in which situations they need to apply which technique for the best results.

Keywords : Bespoke Techniques, Conventional Techniques

1. Introduction

Despite the fact that a team has meticulously followed development processes, unforeseen flaws and unknown problems may be revealed. It is critical to anticipate potential software flaws and improve the way projects are organized and managed for testing and maintenance. The increasing level of complexity and reliance on software has led to a demand for superior, sustainable software at affordable prices. Software defect prediction is a critical activity for improving software quality and reducing operational effort before the system as a whole is deployed [1]. Early recognition of faults may result in quick rectification and the release of maintainable software. In the literature on the subject, there are numerous software metrics. These software metrics and information about faults can be used to build models for anticipating problematic modules or classes at an early stage of the course of the software development cycle.

The goal of software fault prediction is to forecast fault-prone software modules by utilizing some fundamental software project attributes. It is normally carried out by training a prediction model for a well-known project, utilizing project attributes enhanced with fault details, and then using the prediction model to anticipate faults for unfamiliar projects. Software fault prediction is built on the concept that if a project generated in a particular environment results in faults, then any part of it developed in the same environment with the same features would also result in faults [2]. Early diagnosis of problematic modules can help to streamline efforts in later stages of software development by effectively focusing quality assurance efforts on particular modules.

Figure 1 depicts the software fault prediction approach. The graphic depicts three of the most important parts of the software fault prediction (SFP) procedure: the software fault dataset, software fault prediction methodologies, and performance analysis metrics. First, software fault data is gathered from software development project repositories that contain data relating to the software project's production cycle, which can include source code and change logs, and fault data is then gathered from the appropriate fault repositories. Following that, the values of a variety of software metrics are taken into account to serve as independent variables, while the amount of fault information required for determining the fault (e.g., the total number of faults, faulty and non-faulty) serves as the dependent variable. Finally, the developed fault-predictive system is tested using various performance evaluation metrics such as accuracy, precision, recall, and AUC (Area Under the Curve).

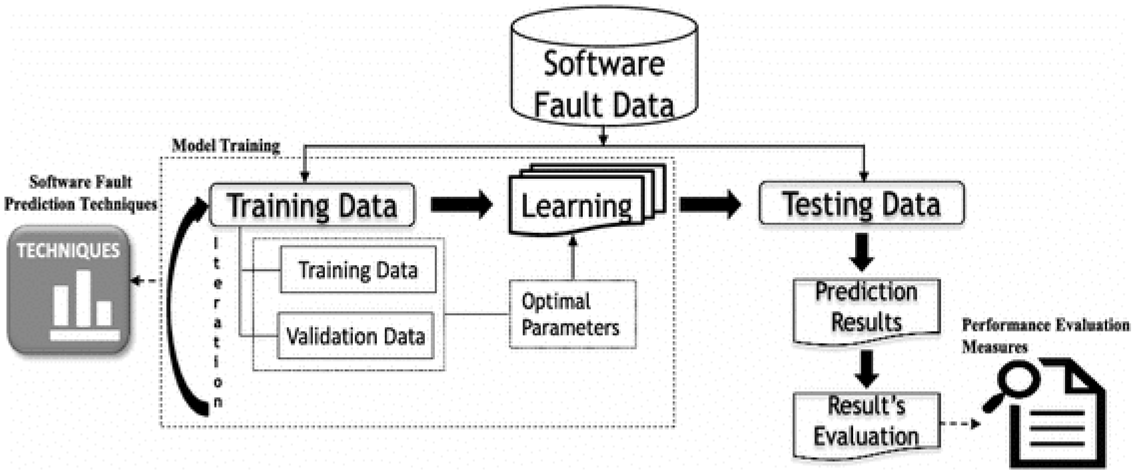

Figure 1: Approach to Software fault prediction [3]

1.2 Conventional technique for SFP

In SFP, most of the conventional STs employed, such as logistic regression (LR), linear regression (LIR), univariate regression (UR), and multivariate regression (MR) [4], are now trustworthy for research in detecting flaws, but such techniques are not beneficial for the original study. In response, scientists developed the application of artificial intelligence (AI) [5] and associated methodologies [6]. Researchers are also looking into other facets of AI to address SFP [7]. The ST is highly effective at identifying known flaws [8] since they make their own money [9], while the statistical technique for SFP [10] is less useful because there aren't many false positives. The types of ST that have been proposed in the literature are depicted in Figure 2.

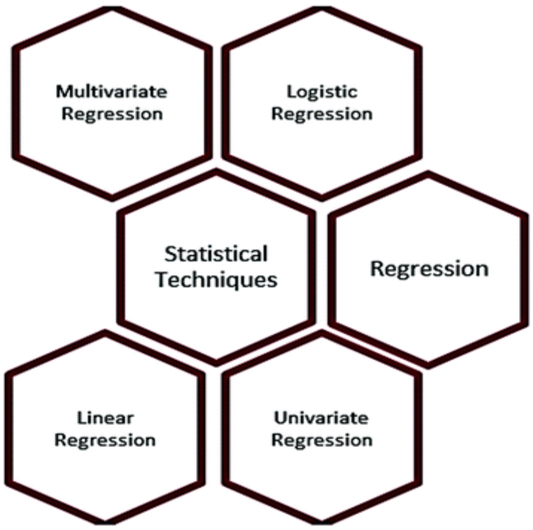

Figure 2: Conventional techniques to SFP [11]

Software quality estimation [12] was proven to be a difficult task in the field of software engineering [13]. According to a recent study on MLT-based SFP, it's essential to conduct studies and in-depth analysis in the SFP to get better outcomes for the SQ. The SFP refers to the direction of characteristics like dependability, tolerance for errors, compliance, complexity of time and effort estimate, etc. in this context. Researchers have also claimed that it is difficult to predict errors and flaws without using machine learning techniques at different stages of the software development life cycle. It is meant to draw attention to the machine learning technique-based SFP experiments that were conducted.

1.3. Machine learning based SFP

Machine learning techniques enhance the fault prediction and software quality. Rule-based learning (RBL), decision trees (DT), Bayesian learners (BL), supervised learning algorithms (SLA), neural networks (NN), support vector machines (SVMs), and evolutionary algorithms (EA) with their sub-techniques are all included in machine learning. The goal of employing machine learning in SFP is to improve software reliability and reach the goal of fault prediction through the use of artificial intelligence [14].

Initially, for SFP, one must not only anticipate failures but also the frequency with which they will occur in that specific program [15]. Previously, many research has employed different machine learning algorithms to predict software faults. For example, random forest has been employed for software faucet prediction by [16][17], support vector machines utilized for the same by [18] and [19], neural network has been employed by [20], decision tree has been tested for the same in [21][22][23].\

For the present literature review, a critical analysis of articles published between 2018 and 2023 has been conducted to present a systematic literature review that highlights the most recent research trends in the field of SDP.

The articles for this review have both conventional and machine learning techniques under study. The chosen library for the selection of the article is IEEE. In total, 213 articles have been studied under this review. In the next section, a metadata analysis of the articles studied under this review will be presented.

2. Metadata analysis of reviewed articles

In this section, the metadata analysis of reviewed articles has been presented in terms of study period, type of publication (conference or journal), citation score of the article, etc.

2.1 Distribution of articles per year throughout the study period of literature review

A literature review, as described by [24], is a methodical, comprehensible, and repeatable design for locating, analyzing, and interpreting the existing corpus of recorded materials. Literature reviews can be thought of as an evaluation of content from a methodological standpoint, where a variety of qualitative and quantitative features are combined to access structural (descriptive) as well as content requirements.

In this section, the distribution of publications per year across the study period has been presented for review. In Table 1, the year-wise count of the publications has been given for the conducted study period. It has been observed that the maximum of the article is from 2019 and 2020, whereas for other years the count is negligibly unequal.

Table 1: Year wise count of the publication for the conducted study period

Year of Publication	Count
2018	31
2019	42
2020	40
2021	36
2022	43
2023	20

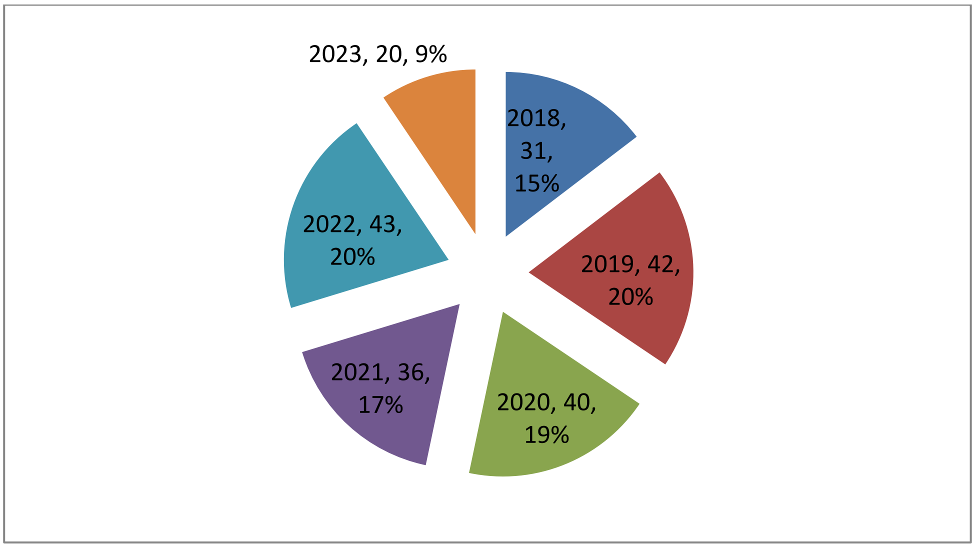

Figure 3: Distribution of publications per year across the study period

2.2 Type of publication

The goal of this comprehensive literature review is to represent the most recent advances that have been made on detecting defect-prone software modules. A thorough research procedure was used to extract pertinent research papers. In the beginning, 213 studies from the well-known online libraries were extracted: IEEE Xplore. The 117 most relevant research publications were chosen for detailed review of IEEE conferences, 35 IEEE journal articles, 3 other (early access etc). The pie form for the type of publication analysis has been shown in figure 4 below.

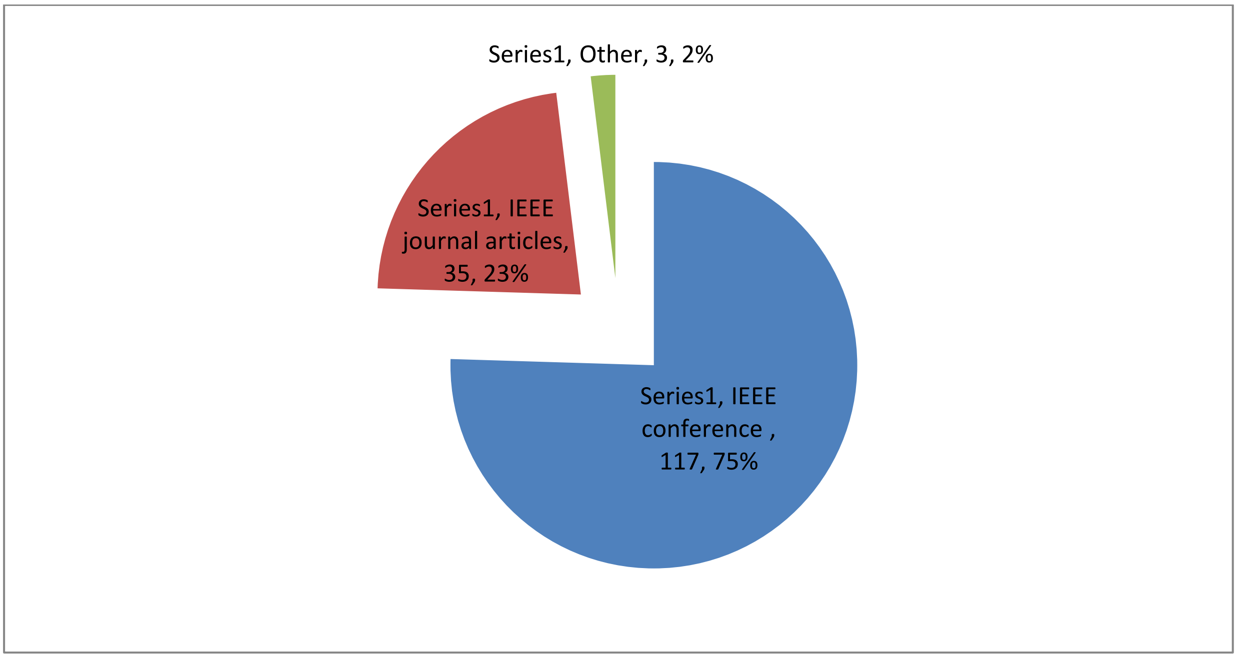

Figure 4: Pie analysis of type of publication

2.3 Quality analysis of reviewed articles

Techniques are not essential or stated unless it is a comprehensive review or meta-analysis. The evaluation of an article's quality can be determined by factors such as its timeliness, the depth and clarity of the topic, whether it indicates the best routes for further research, and how many researchers have been referring to it. In this section, the quality analysis of articles has been presented in terms of their citation scores. The citation scores of the articles have varied from 1 to 89. One is the lowest score of citation, and 89 are the highest score in this regard. In Table 2, given below, the citation-wise score of revised articles has been given.

Table 2: Citation score of reviewed articles

Citation level	Publication Types		Count
	Conference	Journal	
1-20	81	23	104
20-40	5	5	10
40-60	0	3	3
60-80	0	1	1
>80	1	0	1

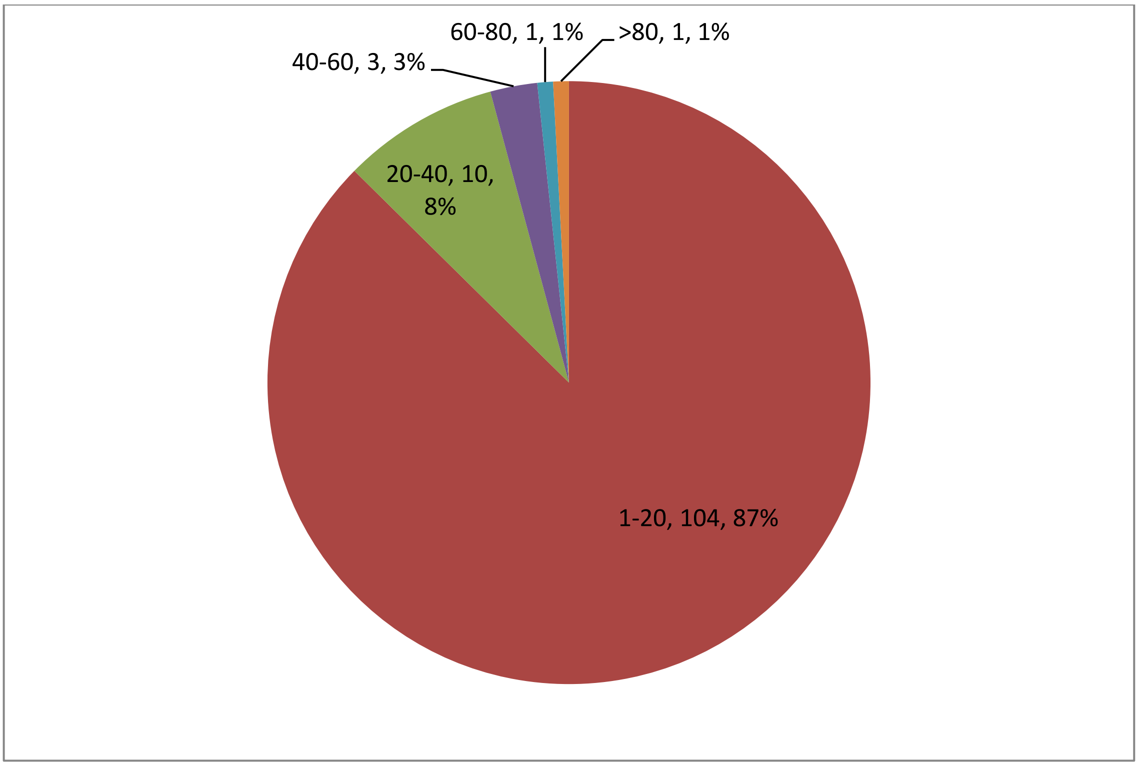

Figure 5: Pie analysis of citation score

From Table 2 and Figure 5, it has been observed that the majority of reviewed articles have been cited in the range of 1–20, and 81 of these articles have been published in IEEE conferences and 23 in IEEE journals. The count of articles among the citation scores in the range of 20–40 is 10, of which 5 have been published at an IEEE conference and 5 in an IEEE journal. Only one article has a citation score of more than 80, and it is a conference article.

3. Methodology

This research is designed to use the systematic literature review (SLR) approach. This method was chosen to conduct a review of the works on software faculties prediction, and SLR is a renowned review process that consists of locating, analyzing, and comprehending the available research data in order to determine the answers to the set research questions.

3.1 Research questions

Research questions have been formulated to assist us through the review and evaluation of previous studies. The goal of this study is to present and assess the results of experiments from previous works on the use of machine learning techniques and conventional techniques for faults prediction models. The following are the sorts of issues that will be addressed in this SLR.

RQ1: Which datasets are commonly used for predicting software bugs?

RQ2 - What machine learning approaches have been chosen for the prediction model?

RQ3: What metrics are commonly utilized for software faults prediction?

RQ4 - What performance metrics are utilized to predict software bugs?

3.2 Keyword Selection

This stage involves selecting specific keywords/words as well as their synonyms while maintaining the study objectives in mind. A search string is formed by combining keywords and their synonyms and IEEE Terms. A list of IEEE terms utilized for selecting the article from IEEE database has been give below in table 3.

Table 3: IEEE terms utilized for article selection

Keyword or IEEE terms	Name
General	Software design ,Software testing, Software quality, Software systems, Software algorithms, Software metrics, Computer bugs Source coding
Objective	Object oriented modeling, Predictive models, Fault diagnosis, Fault tolerance, Computational modeling
Technique	Feature extraction, classification , Error analysis, Correlation, Self-organizing feature maps
Tool	Conventional technique, Machine learning, Neural Network , Fuzzy logic, Regression, Clustering algorithms

3.3 Outlining of the Selection Criteria

This phase defines the boundaries of the study's objectives by providing the selection criteria that were used for the selection of the gathered research papers. The goal of this part of the process is to choose the most suitable research papers for review. This phase can be divided into two parts: defining the criteria for selection and defining the criteria for exclusion.

3.3.1 Criteria for selection

1. Published research articles spanning 2018 to 2023.
2. Research articles that have been presented or published in journals, conferences, or conference proceedings.
3. Research publications that used conventional and machine learning algorithms to perform SFP.
4. Research papers describing actual trials, case studies, comparative research, or a novel method, technique, or framework for predicting software defects.
5. Research publications that provided results from a dataset using a used, proposed, or modified artificial intelligence (AI) algorithm, technique, or framework.

3.3.2 *Criteria for exclusion*

1. Prior to 2016, research articles were published.
2. Research articles that are not written in English.
3. Research papers that don't use SDP.
4. Research articles that don't employ the given technique in their SFP method, technique, or framework
5. Research articles in which the defect prediction method, technique, or framework is not evaluated on any dataset.

4. Results and Discussion

In this section, analysis based on the techniques and tools used in the articles reviewed in this paper has been presented.

4.1 Review of techniques and tools for SFP

Here in this section, the articles have been summarized based on the techniques and tools used. Table 1 summarizes the references, and tools used in the reviewed articles.

Table 1: Analysis based on tools used in the reviewed articles

References	Tool
[28][30][31][36][37][38][43][45][48] [49][50][51][52][53][54][56][58] [60][62][63][70][72][74][75][77][78] [84][87][88][90][92][93][95][98] [103][104][105][109][110][111][113] [114][117][118][119][121][128][129] [130][131][137][139][140][142][144] [145][146][147][148][151][153][154] [156][158][159][161][162] [164][165][166][171][172][174][175][177] [178][182][183][184][186][188][191][192] [193][195][200][202][204][205][206] [207][211][214][216][219][220] [221][227]	Conventional technique
[25][26][27][29][32][33][34][39][40] [41][46][59][61][68][76][78][79][80][81] [82][86][89][91][97][99][100][101][102] [106][107][108][112][116][117][122] [126][127][132][135][138][141] [152][157][163][168][169] [170][180][181][184][187][190][201][203] [208][210][215][217][225][226]	Machine learning
[35] [44][65][66][67][69][85][95][96][102] [120][133][134][149][150][137][175] [189][197][198][212][218][223][224]	Deep learning
[47] [55][66][67][79][83][89][91] [123][125][143]	Neural Network
[44] [64] [71][76][100][137][179]	Fuzzy logic
[42][209]	Regression
[64][73][159]	Clustering algorithms
[67][108][115][155][173] [194][199][219][222][228]	Genetic or Heuristic algorithms

The articles reviewed in the paper have been listed for their references and the tools or methods employed for effective results. It has been found that conventional techniques have been employed in more articles than machine learning and deep learning techniques. The use of fuzzy logic is also very limited. The current trends show that the use of genetic or heuristic algorithms is also very common.

Further analysis based on techniques used in revised articles is presented in Table 2. From Table 2, it has been observed that feature extraction and classification have been used most for fault prediction in software systems.

The use of correlation has been limited to empirical investigation. The well-organized map is also very limited in the research articles.

Table 2: Analysis based on techniques used in the reviewed articles

References	Technique
[27] [29][30][33][39][42][46][51] [58][59][61][72][74][79] [81][82][91][107][108][115][117][126][152] [132][135][137][141][143][150] [155][159][163][169][170] [173][175][179][180][181] [184][187][203][209][210][216][217][222][225]	Feature extraction
[25][26][32][34][40][41][55][65][66] [67][69][70][71][74][80][81][89][100] [101][102][112][120][123][125][127][133][134] [138][149][165][167][189][194][197][198][199][201] [218][219][220][223]	classification
[28][31][35][36][37][38][45][60] [63][68][76][111][119][128][139][140] [147][154][159][161][168][174][202][206]	Error or fault analysis
[43][48][54][62][65][67][80][84][90][177][195][204]	Correlation
[47][73]	Self-organizing feature maps

4.2 Evolution criteria

The evolution criteria used in most of the research articles has been listed below. The following is a quick overview of how the performance and efficiency of the reported models were assessed. Knowing how frequently the model under examination is erroneous is insufficient in many circumstances. It is more important to comprehend how frequently it fails to precisely anticipate a certain outcome. The confusion matrix provides useful information about the model's ability to forecast a certain outcome. The confusion matrix consists of four real-world scenarios. Based on all such aspects the evolution criteria used previously has been elaborated below.

1. Accuracy
2. True-positive rate
3. False-positive rate
4. Sensitivity
5. Specificity
6. Precision
7. Recall
8. F1 Score

Accuracy is defining **as** the ratio of correct predicted instances or samples to all instances or

$$\textbf{Accuracy} = \frac{True\ Positive + True\ Negative}{True\ Positive + True\ Negative + False\ Postive + False\ Nagative} \quad (1)$$

Error rate is defining **as** the ratio of incorrectly predicted instances or samples to all instances.

True-positive rate or positive predictive value is defining **as** the ratio of instances predicted correctly as positive to all positive instances.

False-positive rate or negative predictive value is defining **as** the ratio of instances predicted incorrectly as positive to all negative instances.

Sensitivity is defining **as** the ratio of instances correctly predicted as positive to all instances predicted as positive.

Specificity is defining **as** the ratio of instances correctly predicted as negative to all instances predicted as negative.

Precision is defining **as**

$$\boldsymbol{Precision} = \frac{True\ Positive}{True\ Positive + False\ Positive} \tag{2}$$

(In binary classification **Precision** is **Specificity)**

Recall is defining **as**

$$\boldsymbol{Recall} = \frac{True\ Positive}{True\ Positive + False\ Negative} \tag{3}$$

(Recall is also known as **sensitivity** in binary classification).

F1 score is the final metric. The F1 score is calculated by taking the harmonic mean of precision and recall.

$$\boldsymbol{F1\ Score} = \frac{2}{Recall^{-1} + Precision^{-1}} \tag{4}$$

Or

$$\boldsymbol{F1\ Score} = 2\frac{Recall\ X\ Precision}{Recall + Precision} \tag{5}$$

Or

$$\boldsymbol{F1\ Score} = \frac{True\ Positive}{True\ Positive + \frac{1}{2}(False\ Positive + False\ Negative)} \tag{6}$$

5. Conclusion

To detect problematic modules when the testing phase is complete, the software defect prediction (SDP) procedure can be adapted to serve as a form of quality control activity throughout the entire software development cycle. This prognostication can be used to build a high-quality product at a lower cost because only modules identified as defective will be examined during the testing stage. Many academics have worked over the past decade to improve the quality of SDP. Despite the fact that academics have done reviews and published survey articles in this sector, there is still a lack of up-to-date understanding of research trends. This study fills the void by conducting a systematic review of research publications published between 2018 and 2023. For this review, data has been collected from various reputed libraries, such as IEEE Xplore. Researchers attempted to increase prediction accuracy by adopting unique strategies in data preparation and merging several classifiers using meta-learners. Some academics have developed unique frameworks for multiple processes by combining multiple approaches. In addition, many researchers have employed machine learning classifiers across many datasets in order to discover the few strategies that performed best across all datasets.

This strategy may encourage us to focus solely on high-performing classifiers while developing novel SDP models and frameworks. Analyzing all such research articles revealed that conventional techniques have been employed in more articles than machine learning and deep learning techniques. The use of fuzzy logic is also very limited. The current trends show that the use of genetic or heuristic algorithms is also very common nowadays. Further the use of machine learning is found to be more effective than conventional techniques

References

1. Radjenović, D., Heričko, M., Torkar, R. and Živkovič, A., 2013. Software fault prediction metrics: A systematic literature review. *Information and software technology*, *55*(8), pp.1397-1418.
2. Jiang, Y., Cukic, B. and Ma, Y., 2008. Techniques for evaluating fault prediction models. *Empirical Software Engineering*, *13*, pp.561-595.
3. Rathore, S.S. and Kumar, S., 2019. A study on software fault prediction techniques. *Artificial Intelligence Review*, *51*, pp.255-327.

4. e Abreu, F.B. and Melo, W., 1996, March. Evaluating the impact of object-oriented design on software quality. In *Proceedings of the 3rd international software metrics symposium* (pp. 90-99). IEEE.
5. Abubakar, A. and Alghamdi, J., 2006. Can cohesion predict fault density?. In *IEEE International Conference on Computer Systems and Applications, 2006* (pp. 890-893). IEEE Computer Society.
6. Al Dallal, J., 2011. Improving the applicability of object-oriented class cohesion metrics. *Information and software technology*, *53*(9), pp.914-928.
7. Briand, L.C., Wüst, J., Ikonomovski, S.V. and Lounis, H., 1999, May. Investigating quality factors in object-oriented designs: an industrial case study. In *Proceedings of the 21st international conference on Software engineering* (pp. 345-354).
8. Aggarwal, K.K., Singh, Y., Kaur, A. and Malhotra, R., 2009. Empirical analysis for investigating the effect of object-oriented metrics on fault proneness: a replicated case study. *Software process: Improvement and practice*, *14*(1), pp.39-62.
9. Aggarwal, K.K., Singh, Y., Kaur, A. and Malhotra, R., 2007. Investigating effect of Design Metrics on Fault Proneness in Object-Oriented Systems. *J. Object Technol.*, *6*(10), pp.127-141.
10. Al Dallal, J. and Briand, L.C., 2010. An object-oriented high-level design-based class cohesion metric. *Information and software technology*, *52*(12), pp.1346-1361.
11. Sharma, D. and Chandra, P., 2018. Software fault prediction using machine-learning techniques. In *Smart Computing and Informatics: Proceedings of the First International Conference on SCI 2016, Volume 2* (pp. 541-549). Springer Singapore.
12. D'Ambros, M., Lanza, M. and Robbes, R., 2012. Evaluating defect prediction approaches: a benchmark and an extensive comparison. *Empirical Software Engineering*, *17*, pp.531-577.
13. El Emam, K., Melo, W. and Machado, J.C., 2001. The prediction of faulty classes using object-oriented design metrics. *Journal of systems and software*, *56*(1), pp.63-75.
14. D'Ambros, M., Lanza, M. and Robbes, R., 2009, October. On the relationship between change coupling and software defects. In *2009 16th Working Conference on Reverse Engineering* (pp. 135-144). IEEE.
15. Andersson, C. and Runeson, P., 2007. A replicated quantitative analysis of fault distributions in complex software systems. *IEEE transactions on software engineering*, *33*(5), pp.273-286.
16. Kaur, I. and Kaur, A., 2021. A novel four-way approach designed with ensemble feature selection for code smell detection. *IEEE Access*, *9*, pp.8695-8707.
17. Sohn, J. and Yoo, S., 2019. Empirical evaluation of fault localisation using code and change metrics. *IEEE Transactions on Software Engineering*, *47*(8), pp.1605-1625.
18. Bal, P.R. and Kumar, S., 2020. WR-ELM: Weighted regularization extreme learning machine for imbalance learning in software fault prediction. *IEEE Transactions on Reliability*, *69*(4), pp.1355-1375.
19. Prabha, C.L. and Shivakumar, N., 2020, June. Software defect prediction using machine learning techniques. In *2020 4th International Conference on Trends in Electronics and Informatics (ICOEI)(48184)* (pp. 728-733). IEEE.
20. Pimentel, J., McEwan, A.A. and Yu, H.Q., 2022, August. Towards a Real-Time Smart Prognostics and Health Management (PHM) of Safety Critical Embedded Systems. In *2022 25th Euromicro Conference on Digital System Design (DSD)* (pp. 696-703). IEEE.
21. ALSANGARI, B. and BİRCİK, G., 2023, June. Performance Evaluation of various ML techniques for Software Fault Prediction using NASA dataset. In *2023 5th International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)* (pp. 1-7). IEEE.
22. Faisal, M.I., Bashir, S., Khan, Z.S. and Khan, F.H., 2018, December. An evaluation of machine learning classifiers and ensembles for early stage prediction of lung cancer. In *2018 3rd international conference on emerging trends in engineering, sciences and technology (ICEEST)* (pp. 1-4). IEEE.
23. Yin, W., Gumabay, M.V.N., Lin, H., Tu, C. and Ao, C., 2022, April. Overhead Transmission Lines Early Warning and Decision Support System with Predictive Analytics. In *2022 5th International Conference on Advanced Electronic Materials, Computers and Software Engineering (AEMCSE)* (pp. 310-314). IEEE.
24. Fink, A., 2019. *Conducting research literature reviews: From the internet to paper*. Sage publications.

25. Mohammed, M.Z. and Saleh, I.A., 2022, September. Predicted of Software Fault Based on Random Forest and K-Nearest Neighbor. In *2022 4th International Conference on Advanced Science and Engineering (ICOASE)* (pp. 43-48). IEEE.
26. Kumar, A. and Bansal, A., 2019, April. Software fault proneness prediction using genetic based machine learning techniques. In *2019 4th International Conference on Internet of Things: Smart Innovation and Usages (IoT-SIU)* (pp. 1-5). IEEE.
27. Kumar, T.S. and Booba, B., 2021, October. A Systematic Study on Machine Learning Techniques for Predicting Software Faults. In *2021 IEEE Mysore Sub Section International Conference (MysuruCon)* (pp. 133-136). IEEE.
28. Saini, N., Bhandari, K. and Kumar, K., 2021, December. Various Aspects of Software Fault Prediction: A Review. In *2021 3rd International Conference on Advances in Computing, Communication Control and Networking (ICAC3N)* (pp. 1625-1629). IEEE.
29. Jothi, R., 2018, June. A comparative study of unsupervised learning algorithms for software fault prediction. In *2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS)* (pp. 741-745). IEEE.
30. Sharma, T. and Sangwan, O.P., 2021, September. Sine-Cosine Algorithm for Software Fault Prediction. In *2021 IEEE International Conference on Software Maintenance and Evolution (ICSME)* (pp. 701-706). IEEE.
31. Joon, A., Tyagi, R.K. and Kumar, K., 2020, June. Noise filtering and imbalance class distribution removal for optimizing software fault prediction using best software metrics suite. In *2020 5th International Conference on Communication and Electronics Systems (ICCES)* (pp. 1381-1389). IEEE.
32. Elahi, E., Kanwal, S. and Asif, A.N., 2020, January. A new ensemble approach for software fault prediction. In *2020 17th international Bhurban conference on applied sciences and technology (IBCAST)* (pp. 407-412). IEEE.
33. Bhandari, G.P. and Gupta, R., 2018, October. Machine learning based software fault prediction utilizing source code metrics. In *2018 IEEE 3rd International Conference on Computing, Communication and Security (ICCCS)* (pp. 40-45). IEEE.
34. Prabha, C.L. and Shivakumar, N., 2020, June. Software defect prediction using machine learning techniques. In *2020 4th International Conference on Trends in Electronics and Informatics (ICOEI)(48184)* (pp. 728-733). IEEE.
35. Oberoi, A., Vats, A., Sivasangari, A. and Siwach, K., 2022, October. Research on Efficient Software Defect Prediction Using Deep Learning Approaches. In *2022 2nd International Conference on Technological Advancements in Computational Sciences (ICTACS)* (pp. 549-554). IEEE.
36. Kumar, R., Chaturvedi, A. and Kailasam, L., 2022. An unsupervised software fault prediction approach using threshold derivation. *IEEE Transactions on Reliability*, *71*(2), pp.911-932.
37. Phung, K., Ogunshile, E. and Aydin, M., 2023. Error-Type—A Novel Set of Software Metrics for Software Fault Prediction. *IEEE Access*, *11*, pp.30562-30574.
38. Bhandari, K., Kumar, K. and Sangal, A.L., 2021, May. A study on modeling techniques in software fault prediction. In *2021 2nd International Conference on Secure Cyber Computing and Communications (ICSCCC)* (pp. 6-11). IEEE.
39. Elahi, E., Ayub, A. and Hussain, I., 2021, January. Two staged data preprocessing ensemble model for software fault prediction. In *2021 international Bhurban conference on applied sciences and technologies (IBCAST)* (pp. 506-511). IEEE.
40. Banga, M., Bansal, A. and Singh, A., 2019, April. Implementation of machine learning techniques in software reliability: A framework. In *2019 International Conference on Automation, Computational and Technology Management (ICACTM)* (pp. 241-245). IEEE.
41. Ali, H. and Khan, T.A., 2019, December. On fault localization using machine learning techniques. In *2019 International Conference on Frontiers of Information Technology (FIT)* (pp. 357-3575). IEEE.

42. Prabha, C.L. and Shivakumar, N., 2020, June. Software defect prediction using machine learning techniques. In *2020 4th International Conference on Trends in Electronics and Informatics (ICOEI)(48184)* (pp. 728-733). IEEE.
43. Bal, P.R. and Kumar, S., 2020. WR-ELM: Weighted regularization extreme learning machine for imbalance learning in software fault prediction. *IEEE Transactions on Reliability*, *69*(4), pp.1355-1375.
44. Garg, H., 2021, October. A Brief Analysis of Soft Computing Techniques in Software Fault Prediction. In *2021 5th International Conference on Information Systems and Computer Networks (ISCON)* (pp. 1-7). IEEE.
45. Rathore, S.S., Chouhan, S.S., Jain, D.K. and Vachhani, A.G., 2022. Generative Oversampling Methods for Handling Imbalanced Data in Software Fault Prediction. *IEEE Transactions on Reliability*, *71*(2), pp.747-762.
46. Gangolli, A., Mahmoud, Q.H. and Azim, A., 2022, October. A Machine Learning Based Approach to Detect Fault Injection Attacks in IoT Software Systems. In *2022 IEEE International Conference on Systems, Man, and Cybernetics (SMC)* (pp. 2900-2905). IEEE.
47. Islam, M.R., Akhtar, M.N. and Begum, M., 2022, February. Long short-term memory (LSTM) networks based software fault prediction using data transformation methods. In *2022 International Conference on Advancement in Electrical and Electronic Engineering (ICAEEE)* (pp. 1-6). IEEE.
48. Wang, H. and Khoshgoftaar, T.M., 2019, December. A study on software metric selection for software fault prediction. In *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)* (pp. 1045-1050). IEEE.
49. Uddin, M.N., Li, B., Mondol, M.N., Rahman, M.M., Mia, M.S. and Mondol, E.L., 2021, September. Sdp-ml: an automated approach of software defect prediction employing machine learning techniques. In *2021 International Conference on Electronics, Communications and Information Technology (ICECIT)* (pp. 1-4). IEEE.
50. Jiarpakdee, J., Tantithamthavorn, C.K., Dam, H.K. and Grundy, J., 2020. An empirical study of model-agnostic techniques for defect prediction models. *IEEE Transactions on Software Engineering*, *48*(1), pp.166-185.
51. Cui, C., Liu, B. and Li, G., 2019, January. A novel feature selection method for software fault prediction model. In *2019 Annual Reliability and Maintainability Symposium (RAMS)* (pp. 1-6). IEEE.
52. Hershkovich, E., Stern, R., Abreu, R. and Elmishali, A., 2021, April. Prioritized test generation guided by software fault prediction. In *2021 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW)* (pp. 218-225). IEEE.
53. Goseva-Popstojanova, K., Ahmad, M. and Alshehri, Y., 2019, July. Software fault proneness prediction with group lasso regression: On factors that affect classification performance. In *2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC)* (Vol. 2, pp. 336-343). IEEE.
54. Huda, S., Liu, K., Abdelrazek, M., Ibrahim, A., Alyahya, S., Al-Dossari, H. and Ahmad, S., 2018. An ensemble oversampling model for class imbalance problem in software defect prediction. *IEEE access*, *6*, pp.24184-24195.
55. Prakash, P. and Kumar, K., 2022, December. Artificial Neural Network Based Fault Prediction and Detection in Grid Computing. In *2022 IEEE 9th Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering (UPCON)* (pp. 1-5). IEEE.
56. Sharma, D. and Chandra, P., 2018, October. Efficient fault prediction using exploratory and causal techniques. In *2018 Second World Conference on Smart Trends in Systems, Security and Sustainability (WorldS4)* (pp. 193-197). IEEE.
57. Jagtap, M., Katragadda, P. and Satelkar, P., 2022, January. Software Reliability: Development of Software Defect Prediction Models Using Advanced Techniques. In *2022 Annual Reliability and Maintainability Symposium (RAMS)* (pp. 1-7). IEEE.
58. Phung, K., Ogunshile, E. and Aydin, M., 2021, October. A novel software fault prediction approach to predict error-type proneness in the Java programs using stream X-machine and machine learning. In *2021*

9th International Conference in Software Engineering Research and Innovation (CONISOFT) (pp. 168-179). IEEE.

59. Rizwan, M., Nadeem, A. and Sindhu, M.A., 2020, January. Empirical evaluation of coupling metrics in software fault prediction. In *2020 17th International Bhurban Conference on Applied Sciences and Technology (IBCAST)* (pp. 434-440). IEEE.
60. Singh, P., 2019, December. Stacking based approach for prediction of faulty modules. In *2019 IEEE Conference on Information and Communication Technology* (pp. 1-6). IEEE.
61. ALSANGARI, B. and BİRCİK, G., 2023, June. Performance Evaluation of various ML techniques for Software Fault Prediction using NASA dataset. In *2023 5th International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)* (pp. 1-7). IEEE.
62. Niu, W., Cheng, J. and Song, X., 2021, March. Research on complex system fault prediction method for massive data. In *2021 4th International Conference on Advanced Electronic Materials, Computers and Software Engineering (AEMCSE)* (pp. 981-984). IEEE.
63. Rathore, S.S. and Kumar, S., 2018. An approach for the prediction of number of software faults based on the dynamic selection of learning techniques. *IEEE Transactions on Reliability*, *68*(1), pp.216-236.
64. Arshad, A., Riaz, S., Jiao, L. and Murthy, A., 2018. Semi-supervised deep fuzzy c-mean clustering for software fault prediction. *IEEE Access*, *6*, pp.25675-25685.
65. Selvi, R.T. and Patchaiammal, P., 2022, February. Fault Prediction for Large Scale Projects Using Deep Learning Techniques. In *2022 Second International Conference on Artificial Intelligence and Smart Energy (ICAIS)* (pp. 482-489). IEEE.
66. Dutta, A., 2022, April. Poster: EBFL-An Ensemble Classifier based Fault Localization. In *2022 IEEE Conference on Software Testing, Verification and Validation (ICST)* (pp. 473-476). IEEE.
67. Ayon, S.I., 2019, May. Neural network based software defect prediction using genetic algorithm and particle swarm optimization. In *2019 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT)* (pp. 1-4). IEEE.
68. Ha, T.M.P., Tran, D.H., Hanh, L.T.M. and Binh, N.T., 2019, October. Experimental study on software fault prediction using machine learning model. In *2019 11th International conference on knowledge and systems engineering (KSE)* (pp. 1-5). IEEE.
69. Bhandari, G.P. and Gupta, R., 2018, November. 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)* (pp. 1-6). IEEE.
70. Caulo, M., 2019, August. A taxonomy of metrics for software fault prediction. In *Proceedings of the 2019 27th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering* (pp. 1144-1147).
71. Arshad, A., Riaz, S., Jiao, L. and Murthy, A., 2018. Semi-supervised deep fuzzy c-mean clustering for software fault prediction. *IEEE Access*, *6*, pp.25675-25685.
72. Tsunoda, M., Monden, A., Toda, K., Tahir, A., Bennin, K.E., Nakasai, K., Nagura, M. and Matsumoto, K., 2022, May. Using bandit algorithms for selecting feature reduction techniques in software defect prediction. In *Proceedings of the 19th International Conference on Mining Software Repositories* (pp. 670-681).
73. Zhang, T., Du, Q., Xu, J., Li, J. and Li, X., 2020, December. Software defect prediction and localization with attention-based models and ensemble learning. In *2020 27th Asia-Pacific Software Engineering Conference (APSEC)* (pp. 81-90). IEEE.
74. Gupta, H., Kumar, L. and Neti, L.B.M., 2019, March. An empirical framework for code smell prediction using extreme learning machine. In *2019 9th Annual Information Technology, Electromechanical Engineering and Microelectronics Conference (IEMECON)* (pp. 189-195). IEEE.
75. Taheri, B. and Hosseini, S.A., 2020, December. Detection of high impedance fault in DC microgrid using impedance prediction technique. In *2020 15th International Conference on Protection and Automation of Power Systems (IPAPS)* (pp. 68-73). IEEE.

76. Li, Y., 2022, December. Software Defect Prediction Technology Based on Fuzzy Support Vector Machine. In *2022 IEEE 2nd International Conference on Mobile Networks and Wireless Communications (ICMNWC)* (pp. 1-4). IEEE.
77. Park, S., Li, S. and Mahlke, S., 2018, June. Low cost transient fault protection using loop output prediction. In *2018 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W)* (pp. 109-113). IEEE.
78. Grigoriou, M. and Kontogiannis, K., 2022, October. Project Features That Make Machine-Learning Based Fault Proneness Analysis Successful. In *2022 IEEE 29th Annual Software Technology Conference (STC)* (pp. 59-68). IEEE.
79. Ali, A. and Gravino, C., 2022, August. The Impact of Parameters Optimization in Software Prediction Models. In *2022 48th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)* (pp. 217-224). IEEE.
80. Pandey, S.K. and Tripathi, A.K., 2021, July. Class imbalance issue in software defect prediction models by various machine learning techniques: An empirical study. In *2021 8th International Conference on Smart Computing and Communications (ICSCC)* (pp. 58-63). IEEE.
81. Thongkum, P. and Mekruksavanich, S., 2020, March. Design Flaws Prediction for Impact on Software Maintainability using Extreme Learning Machine. In *2020 Joint International Conference on Digital Arts, Media and Technology with ECTI Northern Section Conference on Electrical, Electronics, Computer and Telecommunications Engineering (ECTI DAMT & NCON)* (pp. 79-82). IEEE.
82. Anjali, C., Dhas, J.P.M. and Singh, J.A.P., 2022, September. A Study on Predicting Software Defects with Machine Learning Algorithms. In *2022 International Conference on Intelligent Innovations in Engineering and Technology (ICIIET)* (pp. 195-198). IEEE.
83. Yu, H., Sun, X., Zhou, Z. and Fan, G., 2021, July. A novel software defect prediction method based on hierarchical neural network. In *2021 IEEE 45th Annual Computers, Software, and Applications Conference (COMPSAC)* (pp. 366-375). IEEE.
84. Kabir, M.A., Keung, J.W., Bennin, K.E. and Zhang, M., 2020, July. A drift propensity detection technique to improve the performance for cross-version software defect prediction. In *2020 IEEE 44th Annual Computers, Software, and Applications Conference (COMPSAC)* (pp. 882-891). IEEE.
85. Sun, Yuanyuan, Lele Xu, Ye Li, Lili Guo, Zhongsong Ma, and Yongming Wang. "Utilizing deep architecture networks of VAE in software fault prediction." In *2018 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Ubiquitous Computing & Communications, Big Data & Cloud Computing, Social Computing & Networking, Sustainable Computing & Communications (ISPA/IUCC/BDCloud/SocialCom/SustainCom)*, pp. 870-877. IEEE, 2018.
86. Samantaray, R. and Das, H., 2023, March. Performance Analysis of Machine Learning Algorithms Using Bagging Ensemble Technique for Software Fault Prediction. In *2023 6th International Conference on Information Systems and Computer Networks (ISCON)* (pp. 1-7). IEEE.
87. Rizwan, M., Habib, Z., Sarwar, S., Iqbal, M., Safyan, M. and Almakhles, D., 2022, December. An Exploratory Framework for Intelligent Labelling of Fault Datasets. In *2022 Human-Centered Cognitive Systems (HCCS)* (pp. 1-11). IEEE.
88. Parande, P.V. and Banga, M.K., 2019, December. Evolutionary Computing Assisted Heterogenous Ensemble Model for Web-of-Service Software Reusability Prediction. In *2019 4th International Conference on Electrical, Electronics, Communication, Computer Technologies and Optimization Techniques (ICEECCOT)* (pp. 158-163). IEEE.
89. AlShaikh, F. and Elmedany, W., 2021. Estimate the performance of applying machine learning algorithms to predict defects in software using weka.
90. Afshinpour, B., Groz, R. and Amini, M.R., 2022, October. Correlating test events with monitoring logs for test log reduction and anomaly prediction. In *2022 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW)* (pp. 274-280). IEEE.

91. Bandarupalli, P. and Yalla, P., 2022, May. Coupling and Cohesion Metrics-based Fault Predictions using Machine learning Algorithm. In *2022 International Conference on Applied Artificial Intelligence and Computing (ICAAIC)* (pp. 412-418). IEEE.
92. McIntosh, S. and Kamei, Y., 2018, May. Are fix-inducing changes a moving target? a longitudinal case study of just-in-time defect prediction. In *Proceedings of the 40th International Conference on Software Engineering* (pp. 560-560).
93. Lin, J.S. and Huang, C.Y., 2022. Queueing-Based Simulation for Software Reliability Analysis. *IEEE Access*, *10*, pp.107729-107747.
94. Tan, X. and Liu, J., 2021, December. ACLM: Software Aging Prediction of Virtual Machine Monitor Based on Attention Mechanism of CNN-LSTM Model. In *2021 IEEE 21st International Conference on Software Quality, Reliability and Security (QRS)* (pp. 759-767). IEEE.
95. Malhotra, R. and Jain, J., 2020, January. Handling imbalanced data using ensemble learning in software defect prediction. In *2020 10th International Conference on Cloud Computing, Data Science & Engineering (Confluence)* (pp. 300-304). IEEE.
96. Malhotra, R., Gupta, S. and Singh, T., 2020, July. A systematic review on application of deep learning techniques for software quality predictive modeling. In *2020 International Conference on Computational Performance Evaluation (ComPE)* (pp. 332-337). IEEE.
97. Sharma, P. and Sangal, A.L., 2021, May. Extensive Software Fault Prediction: An Ensemble based comparison. In *2021 2nd International Conference on Secure Cyber Computing and Communications (ICSCCC)* (pp. 432-436). IEEE.
98. Chiang, M.C., Huang, C.Y., Wu, C.Y. and Tsai, C.Y., 2020. Analysis of a fault-tolerant framework for reliability prediction of service-oriented architecture systems. *IEEE Transactions on Reliability*, *70*(1), pp.13-48.
99. Chakravarty, K. and Singh, J., 2022, December. Optimizing Defect Removal Efficiency by Defect Prediction using Machine Learning. In *2022 OITS International Conference on Information Technology (OCIT)* (pp. 205-210). IEEE.
100. Saha, D. and Chatterjee, S., 2022, November. Optimized Decision Tree-based Early Phase Software Dependability Analysis in Uncertain Environment. In *2022 International Interdisciplinary Conference on Mathematics, Engineering and Science (MESIICON)* (pp. 1-6). IEEE.
101. Pethe, Y.S. and Das, H., 2023, May. Software fault prediction using a differential evolution-based wrapper approach for feature selection. In *2023 International Conference on Communication, Circuits, and Systems (IC3S)* (pp. 1-6). IEEE.
102. Young, S., Abdou, T. and Bener, A., 2018, May. A replication study: just-in-time defect prediction with ensemble learning. In *Proceedings of the 6th International Workshop on Realizing Artificial Intelligence Synergies in Software Engineering* (pp. 42-47).
103. Gupta, D., 2021, September. Prediction of sensor faults and outliers in IoT devices. In *2021 9th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions)(ICRITO)* (pp. 1-5). IEEE.
104. Campos, J.R. and Costa, E., 2020, October. Fault injection to generate failure data for failure prediction: A case study. In *2020 IEEE 31st International Symposium on Software Reliability Engineering (ISSRE)* (pp. 115-126). IEEE.
105. Yakymets, N. and Adedjouma, M., 2020, October. Model-based quantitative fault tree analysis based on fides reliability prediction. In *2020 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW)* (pp. 161-162). IEEE.
106. Sabnis, P.S., Joshi, S. and Naveenkumar, J., 2022, September. A Study on Machine Learning Techniques based Software Reliability Assessment. In *2022 4th International Conference on Inventive Research in Computing Applications (ICIRCA)* (pp. 687-692). IEEE.
107. Raveendran, R., Devika, K.B. and Subramanian, S.C., 2020. Brake fault identification and fault-tolerant directional stability control of heavy road vehicles. *IEEE Access*, *8*, pp.169229-169246.

108. Thaher, T. and Arman, N., 2020, April. 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)* (pp. 249-254). IEEE.
109. Alharthi, Z.S., Alsaeedi, A. and Yafooz, W.M., 2021, December. Software Defect Prediction Approaches: A Review. In *2021 4th International Conference on Bio-Engineering for Smart Technologies (BioSMART)* (pp. 1-6). IEEE.
110. Eghbali, S., Kudva, V., Rothermel, G. and Tahvildari, L., 2019, May. Supervised tie breaking in test case prioritization. In *2019 IEEE/ACM 41st International Conference on Software Engineering: Companion Proceedings (ICSE-Companion)* (pp. 242-243). IEEE.
111. Li, L., Lu, M. and Gu, T., 2018, October. A systematic modeling approach for failure indicators of complex software-intensive systems. In *2018 12th International Conference on Reliability, Maintainability, and Safety (ICRMS)* (pp. 43-51). IEEE.
112. Feng, Y., Jones, J., Chen, Z. and Fang, C., 2018, April. An empirical study on software failure classification with multi-label and problem-transformation techniques. In *2018 IEEE 11th International Conference on Software Testing, Verification and Validation (ICST)* (pp. 320-330). IEEE.
113. Gradišnik, M., Beranič, T., Karakatič, S. and Mausaš, G., 2019, May. Adapting God Class thresholds for software defect prediction: A case study. In *2019 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)* (pp. 1537-1542). IEEE.
114. Kulamala, V.K., Teja, A.S.C., Maru, A., Singla, Y. and Mohapatra, D.P., 2018, December. Predicting software reliability using computational intelligence techniques: a review. In *2018 international conference on information technology (ICIT)* (pp. 114-119). IEEE.
115. Ali, A. and Gravino, C., 2020, December. Bio-inspired algorithms in software fault prediction: A systematic literature review. In *2020 14th International Conference on Open Source Systems and Technologies (ICOSST)* (pp. 1-8). IEEE.
116. Ali, A. and Gravino, C., 2020, December. Bio-inspired algorithms in software fault prediction: A systematic literature review. In *2020 14th International Conference on Open Source Systems and Technologies (ICOSST)* (pp. 1-8). IEEE.
117. Chen, L., Jing, Y., Zheng, Q. and Xiao, F., 2019, December. Fault diagnosis device for nuclear radiation monitoring system. In *2019 IEEE 4th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)* (Vol. 1, pp. 2012-2018). IEEE.
118. Long, B., Cao, T., Shen, D., Guerrero, J.M., Rodriguez, J. and Chong, K.T., 2022. Fault-Tolerant Sequential MPC for Vertical Switch Open-Circuit Fault and ZSCC Suppression for Parallel T-Type Converters. *IEEE Transactions on Power Electronics*, *37*(10), pp.11787-11802.
119. Gillani, S.F.U.H., Nadeem, A. and Rizwan, M., 2022, August. An Effective Undersampling Approach to Deal with Class Imbalance Problem in Software Defect Prediction. In *2022 19th International Bhurban Conference on Applied Sciences and Technology (IBCAST)* (pp. 225-230). IEEE.
120. Ahmed, M.M., Kiran, B.S., Sai, P.H. and Bisi, M., 2021, July. Software fault-prone module classification using learning automata based deep neural network model. In *2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT)* (pp. 1-6). IEEE.
121. Afshinpour, B., Groz, R. and Amini, M.R., 2022, December. Telemetry-based Software Failure Prediction by Concept-space Model Creation. In *2022 IEEE 22nd International Conference on Software Quality, Reliability and Security (QRS)* (pp. 199-208). IEEE.
122. Amar, A. and Rigby, P.C., 2019, May. Mining historical test logs to predict bugs and localize faults in the test logs. In *2019 IEEE/ACM 41st International Conference on Software Engineering (ICSE)* (pp. 140-151). IEEE.
123. Bahaweres, R.B., Agustian, F., Hermadi, I., Suroso, A.I. and Arkeman, Y., 2020, October. Software defect prediction using neural network based smote. In *2020 7th International Conference on Electrical Engineering, Computer Sciences and Informatics (EECSI)* (pp. 71-76). IEEE.

124. Gaol, F.L., Trisetyarso, A. and Budiharto, W., 2022, November. Software Testing Model by Measuring the Level of Accuracy Fault Output Using Neural Network Algorithm. In *2022 IEEE 7th International Conference on Information Technology and Digital Applications (ICITDA)* (pp. 1-6). IEEE.
125. Gaol, F.L., Trisetyarso, A. and Budiharto, W., 2022, November. Software Testing Model by Measuring the Level of Accuracy Fault Output Using Neural Network Algorithm. In *2022 IEEE 7th International Conference on Information Technology and Digital Applications (ICITDA)* (pp. 1-6). IEEE.
126. Molawade, M.H., Joshi, S.D. and Jadhav, R., 2021, November. Software reliability prediction using data abstraction and Random forest Algorithm. In *2021 IEEE 8th Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering (UPCON)* (pp. 1-5). IEEE.
127. Copic, M., Leupers, R. and Ascheid, G., 2020, October. Modelling machine learning components for mapping and scheduling of Autosar runnables. In *2020 IEEE 31st International Symposium on Software Reliability Engineering (ISSRE)* (pp. 127-137). IEEE.
128. Shaikh, S., Changan, L., Rasheed, M. and Rizwan, S., 2019, January. Wide research on software defect model with overgeneralization problems. In *2019 2nd International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)* (pp. 1-6). IEEE.
129. Yu, X., Bennin, K.E., Liu, J., Keung, J.W., Yin, X. and Xu, Z., 2019, February. An empirical study of learning to rank techniques for effort-aware defect prediction. In *2019 IEEE 26th International Conference on Software Analysis, Evolution and Reengineering (SANER)* (pp. 298-309). IEEE.
130. Golagha, M., Pretschner, A. and Briand, L.C., 2020, October. Can we predict the quality of spectrum-based fault localization?. In *2020 IEEE 13th International Conference on Software Testing, Validation and Verification (ICST)* (pp. 4-15). IEEE.
131. Wen, W., Zhang, B., Gu, X. and Ju, X., 2019, February. An empirical study on combining source selection and transfer learning for cross-project defect prediction. In *2019 IEEE 1st International Workshop on Intelligent Bug Fixing (IBF)* (pp. 29-38). IEEE.
132. Massoudi, M., Jain, N.K. and Bansal, P., 2021, February. Software defect prediction using dimensionality reduction and deep learning. In *2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV)* (pp. 884-893). IEEE.
133. Dai Vu, D., Vu, X.T. and Kim, Y., 2021, October. Deep Learning-based fault prediction in cloud system. In *2021 International Conference on Information and Communication Technology Convergence (ICTC)* (pp. 1826-1829). IEEE.
134. Tran, M.N., Vu, X.T. and Kim, Y., 2022. Proactive Stateful Fault-Tolerant System for Kubernetes Containerized Services. *IEEE Access*, *10*, pp.102181-102194.
135. Mahdi, M.N., MH, M.Z., Yusof, A., Cheng, L.K., Azmi, M.S.M. and Ahmad, A.R., 2020, August. Design and Development of Machine Learning Technique for Software Project Risk Assessment-A Review. In *2020 8th International Conference on Information Technology and Multimedia (ICIMU)* (pp. 354-362). IEEE.
136. Monni, C. and Pezzè, M., 2019, May. Energy-based anomaly detection a new perspective for predicting software failures. In *2019 IEEE/ACM 41st International Conference on Software Engineering: New Ideas and Emerging Results (ICSE-NIER)* (pp. 69-72). IEEE.
137. Rani, S. and Ahmad, N., 2020, October. Software Reliability Growth Modeling with Burr Type XII using Fuzzy Logic. In *2020 5th International Conference on Computing, Communication and Security (ICCCS)* (pp. 1-5). IEEE.
138. Nehi, M.M., Fakhrpoor, Z. and Moosavi, M.R., 2018, May. Defects in The Next Release; Software Defect Prediction Based on Source Code Versions. In *Electrical Engineering (ICEE), Iranian Conference on* (pp. 1589-1594). IEEE.
139. Ghosh, S., Rana, A. and Kansal, V., 2018, June. A Novel Model Based on Nonlinear Manifold Detection for Software Defect Prediction. In *2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS)* (pp. 140-145). IEEE.
140. Podgurski, A. and Küçük, Y., 2020, September. Counterfault: Value-based fault localization by modeling and predicting counterfactual outcomes. In *2020 IEEE International Conference on Software Maintenance and Evolution (ICSME)* (pp. 382-393). IEEE.

---

141. Immaculate, S.D., Begam, M.F. and Floramary, M., 2019, March. Software bug prediction using supervised machine learning algorithms. In *2019 International conference on data science and communication (IconDSC)* (pp. 1-7). IEEE.
142. Soomro, A.M., Naeem, A.B., Senapati, B., Bashir, K., Pradhan, S., Maaliw, R.R. and Sakr, H.A., 2023, January. Constructor Development: Predicting Object Communication Errors. In *2023 IEEE International Conference on Emerging Trends in Engineering, Sciences and Technology (ICES&T)* (pp. 1-7). IEEE.
143. Sourashtriya, A. and Tomar, M., 2021, May. A comparative study for fault location detection in bipolar HVDC transmission systems using wavelet transform and artificial neural networks. In *2021 3rd International Conference on Signal Processing and Communication (ICPSC)* (pp. 161-165). IEEE.
144. Sajan, S., Chacko, S.J., Pai, V. and Pai, B.K., 2020, January. Performance evaluation of various algorithms that affect fault detection in wireless sensor network. In *2020 Fourth International conference on inventive systems and control (ICISC)* (pp. 540-545). IEEE.
145. Xia, X. and Lo, D., 2023. Information Retrieval-Based Techniques for Software Fault Localization. *Handbook of Software Fault Localization: Foundations and Advances*, pp.365-391.
146. Nayrolles, M. and Hamou-Lhadj, A., 2018, May. Clever: Combining code metrics with clone detection for just-in-time fault prevention and resolution in large industrial projects. In *Proceedings of the 15th international conference on mining software repositories* (pp. 153-164).
147. Sohn, J. and Yoo, S., 2019. Empirical evaluation of fault localisation using code and change metrics. *IEEE Transactions on Software Engineering*, *47*(8), pp.1605-1625.
148. Lin, J., Ye, R., Su, X. and An, D., 2021, October. A Novel Nonlinear Mapping based Compatibility Method for Zero-Shot Classification in Intelligent Fault Diagnosis. In *2021 China Automation Congress (CAC)* (pp. 5719-5724). IEEE.
149. Hoang, T., Dam, H.K., Kamei, Y., Lo, D. and Ubayashi, N., 2019, May. DeepJIT: an end-to-end deep learning framework for just-in-time defect prediction. In *2019 IEEE/ACM 16th International Conference on Mining Software Repositories (MSR)* (pp. 34-45). IEEE.
150. Wu, X., Zheng, W., Chen, J., Bai, H., Hu, D. and Mu, D., 2018, December. A GMM and SVM Combined Approach for Automatically Software Fault Localization. In *2018 IEEE International Conference on Progress in Informatics and Computing (PIC)* (pp. 357-363). IEEE.
151. Kaitovic, I. and Malek, M., 2018. Impact of failure prediction on availability: Modeling and comparative analysis of predictive and reactive methods. *IEEE Transactions on Dependable and Secure Computing*, *17*(3), pp.493-505.
152. Campos, J.R., Vieira, M. and Costa, E., 2018, September. Exploratory study of machine learning techniques for supporting failure prediction. In *2018 14th European Dependable Computing Conference (EDCC)* (pp. 9-16). IEEE.
153. Domingos, J., Barbosa, R. and Madeira, H., 2021, September. Why is it so hard to predict computer systems failures?. In *2021 17th European Dependable Computing Conference (EDCC)* (pp. 41-44). IEEE.
154. Mohamed, F.A., Salama, C.R., Yousef, A.H. and Salem, A.M., 2020, October. A Universal Model for Defective Classes Prediction Using Different Object-Oriented Metrics Suites. In *2020 2nd Novel Intelligent and Leading Emerging Sciences Conference (NILES)* (pp. 65-70). IEEE.
155. Jacob, R.J., Kamat, R.J., Sahithya, N.M., John, S.S. and Shankar, S.P., 2021, October. Voting based ensemble classification for software defect prediction. In *2021 IEEE Mysore Sub Section International Conference (MysuruCon)* (pp. 358-365). IEEE.
156. Thaher, T., Mafarja, M., Abdalhaq, B. and Chantar, H., 2019, October. Wrapper-based feature selection for imbalanced data using binary queuing search algorithm. In *2019 2nd international conference on new trends in computing sciences (ICTCS)* (pp. 1-6). IEEE.
157. Eken, B., Atar, R., Sertalp, S. and Tosun, A., 2019, November. Predicting defects with latent and semantic features from commit logs in an industrial setting. In *2019 34th IEEE/ACM International Conference on Automated Software Engineering Workshop (ASEW)* (pp. 98-105). IEEE.

158. Arya, A. and Malik, S.K., 2023, January. Design an Improved Model of Software Defect Prediction Model for Web Applications. In *2023 International Conference on Artificial Intelligence and Smart Communication (AISC)* (pp. 119-123). IEEE.
159. Prashant, P., Tickoo, A., Sharma, S. and Jamil, J., 2019, January. Optimization of cost to calculate the release time in software reliability using python. In *2019 9th International Conference On Cloud Computing, Data Science & Engineering (Confluence)* (pp. 471-474). IEEE.
160. Wang, F., Ai, J. and Zou, Z., 2019, July. A cluster-based hybrid feature selection method for defect prediction. In *2019 IEEE 19th International Conference on Software Quality, Reliability and Security (QRS)* (pp. 1-9). IEEE.
161. Wei, Xinlei, Yujuan Tan, Duo Liu, Daitao Wu, Yu Wu, Xianzhang Chen, and Jian Li. "Weak Network Oriented Mobile Distributed Storage: A Hybrid Fault-Tolerance Scheme Based on Potential Replicas." In *2022 IEEE 24th Int Conf on High Performance Computing & Communications; 8th Int Conf on Data Science & Systems; 20th Int Conf on Smart City; 8th Int Conf on Dependability in Sensor, Cloud & Big Data Systems & Application (HPCC/DSS/SmartCity/DependSys)*, pp. 372-379. IEEE, 2022.
162. Behera, S., Wan, L., Mueller, F., Wolf, M. and Klasky, S., 2022, May. P-ckpt: Coordinated Prioritized Checkpointing. In *2022 IEEE International Parallel and Distributed Processing Symposium (IPDPS)* (pp. 436-446). IEEE.
163. Efendioglu, M., Sen, A. and Koroglu, Y., 2018. Bug prediction of systemc models using machine learning. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, *38*(3), pp.419-429.
164. Parul, P., Kontogiannis, K. and Brealey, C., 2023, June. Prediction of Bug Inducing Commits Using Metrics Trend Analysis. In *2023 IEEE 47th Annual Computers, Software, and Applications Conference (COMPSAC)* (pp. 830-839). IEEE.
165. Andrade, E., Pietrantuono, R., Machida, F. and Cotroneo, D., 2021. A comparative analysis of software aging in image classifiers on cloud and edge. *IEEE Transactions on Dependable and Secure Computing*.
166. Omer, A., Rathore, S.S. and Kumar, S., 2023. ME-SFP: A Mixture-of-Experts-Based Approach for Software Fault Prediction. *IEEE Transactions on Reliability*.
167. Widyasari, R., Prana, G.A.A., Haryono, S.A., Tian, Y., Zachiary, H.N. and Lo, D., 2022, May. XAI4FL: enhancing spectrum-based fault localization with explainable artificial intelligence. In *Proceedings of the 30th IEEE/ACM International Conference on Program Comprehension* (pp. 499-510).
168. Campos, J.R., Costa, E. and Vieira, M., 2022, September. On the Applicability of Machine Learning-based Online Failure Prediction for Modern Complex Systems. In *2022 18th European Dependable Computing Conference (EDCC)* (pp. 49-56). IEEE.
169. Canito, A., Fernandes, M., Mourinho, J., Tosun, S., Kaya, K., Turupcu, A., Lagares, A., Karabulut, H. and Marreiros, G., 2021, October. Flexible architecture for data-driven predictive maintenance with support for offline and online machine learning techniques. In *IECON 2021–47th Annual Conference of the IEEE Industrial Electronics Society* (pp. 1-7). IEEE.
170. Campos, J.R., Costa, E. and Vieira, M., 2019. Improving failure prediction by ensembling the decisions of machine learning models: A case study. *IEEE Access*, *7*, pp.177661-177674.
171. Lodhi, E., Lina, W., Pu, Y., Javed, M.Y., Lodhi, Z., Zhijie, J. and Javed, U., 2020, February. Performance evaluation of faults in a photovoltaic array based on VI and VP characteristic curve. In *2020 12th International Conference on Measuring Technology and Mechatronics Automation (ICMTMA)* (pp. 85-90). IEEE.
172. Paterson, D., Campos, J., Abreu, R., Kapfhammer, G.M., Fraser, G. and McMinn, P., 2019, April. An empirical study on the use of defect prediction for test case prioritization. In *2019 12th IEEE Conference on Software Testing, Validation and Verification (ICST)* (pp. 346-357). IEEE.
173. Wheeler, K.A. and Faried, S.O., 2020, November. An Analysis and Protection Scheme to Prevent Loss of Coordination due to Microgrid Contributions: Part II–Optimization and Mitigation. In *2020 IEEE Electric Power and Energy Conference (EPEC)* (pp. 1-6). IEEE.

174. Nezzari, Y. and Bridges, C.P., 2019. ACEDR: automatic Compiler error detection and recovery for COTS CPU and caches. *IEEE Transactions on Reliability*, *68*(3), pp.859-871.
175. Huang, P.W. and Chung, K.J., 2019, October. The prediction of positioning shift for a robot arm using machine learning techniques. In *2019 14th International Microsystems, Packaging, Assembly and Circuits Technology Conference (IMPACT)* (pp. 58-61). IEEE.
176. Malik, A. and de Fréin, R., 2020, July. A proactive-restoration technique for SDNs. In *2020 IEEE Symposium on Computers and Communications (ISCC)* (pp. 1-6). IEEE.
177. Jazebi, S., De Leon, F. and Nelson, A., 2019. Review of wildfire management techniques—Part I: Causes, prevention, detection, suppression, and data analytics. *IEEE Transactions on Power Delivery*, *35*(1), pp.430-439.
178. Nair, V., Yu, Z., Menzies, T., Siegmund, N. and Apel, S., 2018. Finding faster configurations using flash. *IEEE Transactions on Software Engineering*, *46*(7), pp.794-811.
179. Tarfulea, N., Simo, A., Vatau, D., Frigura-Iliasa, F.M., Musuroi, S. and Andea, P., 2020, January. Fuzzy Logic Based Diagnosis of SF6 Switching Devices. In *2020 IEEE 18th World Symposium on Applied Machine Intelligence and Informatics (SAMI)* (pp. 31-34). IEEE.
180. Yugapriya, M., Judeson, A.K.J. and Jayanthy, S., 2022, March. Predictive Maintenance of Hydraulic System using Machine Learning Algorithms. In *2022 International Conference on Electronics and Renewable Systems (ICEARS)* (pp. 1208-1214). IEEE.
181. Khan, F.N., Fan, Q., Lu, C. and Lau, A.P.T., 2018, July. Machine learning-assisted optical performance monitoring in fiber-optic networks. In *2018 IEEE Photonics Society Summer Topical Meeting Series (SUM)* (pp. 53-54). IEEE.
182. Yang, N. and Wang, Y., 2019, December. Predicting the silent data corruption vulnerability of instructions in programs. In *2019 IEEE 25th International Conference on Parallel and Distributed Systems (ICPADS)* (pp. 862-869). IEEE.
183. Arévalo, F., Oestanto, E. and Schwung, A., 2018, July. Development of a Mobile App for Fault Detection Assessment based on Information Fusion. In *2018 IEEE 16th International Conference on Industrial Informatics (INDIN)* (pp. 635-640). IEEE.
184. Gunasekaran, J.R. and Mishra, C.S., 2021, October. MLPP: Exploring Transfer Learning and Model Distillation for Predicting Application Performance. In *2021 IEEE International Conference on Networking, Architecture and Storage (NAS)* (pp. 1-4). IEEE.
185. Pinto, V.H.S.C., Souza, S.R. and Souza, P.S., 2019, May. A preliminary fault taxonomy for multi-tenant SaaS systems. In *2019 19th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID)* (pp. 178-187). IEEE.
186. Rahman, A. and Williams, L., 2018, April. Characterizing defective configuration scripts used for continuous deployment. In *2018 IEEE 11th International conference on software testing, verification and validation (ICST)* (pp. 34-45). IEEE.
187. Samir, R., Elhakim, Y., Adel, S., Samy, I. and Ismail, T., 2022, October. Stability Analysis and Fault Detection of Telecommunication Towers Using Decision Tree Algorithm Under Wind Speed Condition. In *2022 4th Novel Intelligent and Leading Emerging Sciences Conference (NILES)* (pp. 45-49). IEEE.
188. Gerlin, N., Kaja, E., Vargas, F., Lu, L., Breitenreiter, A., Chen, J., Ulbricht, M., Gomez, M., Tahiraga, A., Prebeck, S. and Jentzsch, E., 2023, May. Bits, Flips and RISCs. In *2023 26th International Symposium on Design and Diagnostics of Electronic Circuits and Systems (DDECS)* (pp. 140-149). IEEE.
189. Ruospo, A., Gavarini, G., Bragaglia, I., Traiola, M., Bosio, A. and Sanchez, E., 2022, April. Selective hardening of critical neurons in deep neural networks. In *2022 25th International Symposium on Design and Diagnostics of Electronic Circuits and Systems (DDECS)* (pp. 136-141). IEEE.
190. Li, Z., Tian, S., Liu, Y. and Wu, Y., 2023, July. Machine Learning Driven Identification of Coincidental Correct Test Cases. In *2023 3rd International Symposium on Computer Technology and Information Science (ISCTIS)* (pp. 284-288). IEEE.

191. de Araujo Neto, J.P., Pianto, D.M. and Ralha, C.G., 2018, September. A Fault-Tolerant Agent-Based Architecture for Transient Servers in Fog Computing. In *2018 30th International Symposium on Computer Architecture and High Performance Computing (SBAC-PAD)* (pp. 282-289). IEEE.
192. Seyezhai, R., 2022, December. Reliability Prediction of Bridgeless AC-DC SEPIC with V-Fill for LED Applications. In *2022 International Conference on Power, Energy, Control and Transmission Systems (ICPECTS)* (pp. 1-6). IEEE.
193. Mao, D., Chen, L. and Zhang, L., 2019, April. An extensive study on cross-project predictive mutation testing. In *2019 12th IEEE Conference on Software Testing, Validation and Verification (ICST)* (pp. 160-171). IEEE.
194. Rossini, R., Prato, G., Conzon, D., Pastrone, C., Pereira, E., Reis, J., Gonçalves, G., Henriques, D., Santiago, A.R. and Ferreira, A., 2021, September. AI environment for predictive maintenance in a manufacturing scenario. In *2021 26th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)* (pp. 1-8). IEEE.
195. Chandrasekaran, K.S., Mahalakshmi, V. and Padmanaban, M.A., 2019, December. Measuring Performance Reliability of a system using DataMining. In *2019 11th International Conference on Advanced Computing (ICoAC)* (pp. 164-166). IEEE.
196. Tang, X., Liu, Y., Zeng, Z. and Veeravalli, B., 2021. Service Cost Effective and Reliability Aware Job Scheduling Algorithm on Cloud Computing Systems. *IEEE Transactions on Cloud Computing*.
197. Wang, H., Xu, J., Xu, C., Ma, X. and Lu, J., 2020, June. Dissector: Input validation for deep learning applications by crossing-layer dissection. In *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering* (pp. 727-738).
198. Cui, A., Zhang, Y., Zhang, P., Dong, W. and Wang, C., 2020, December. Intelligent health management of fixed-wing UAVs: A deep-learning-based approach. In *2020 16th International Conference on Control, Automation, Robotics and Vision (ICARCV)* (pp. 1055-1060). IEEE.
199. Chaudhary, A., Agarwal, A.P., Rana, A. and Kumar, V., 2019, February. Crow search optimization based approach for parameter estimation of SRGMs. In *2019 Amity International Conference on Artificial Intelligence (AICAI)* (pp. 583-587). IEEE.
200. De La Vega, A., Sánchez, P. and Kolovos, D., 2018, September. Pinset: A DSL for extracting datasets from models for data mining-based quality analysis. In *2018 11th International Conference on the Quality of Information and Communications Technology (QUATIC)* (pp. 83-91). IEEE.
201. Kaur, I. and Kaur, A., 2021. A novel four-way approach designed with ensemble feature selection for code smell detection. *IEEE Access*, *9*, pp.8695-8707.
202. Zheng, W., Shen, Y. and Xiao, T., 2020, September. Asocial: Adaptive Task Re-Allocation in Distributed Computing Systems with Node Failures. In *2020 21st Asia-Pacific Network Operations and Management Symposium (APNOMS)* (pp. 179-184). IEEE.
203. Faisal, M.I., Bashir, S., Khan, Z.S. and Khan, F.H., 2018, December. An evaluation of machine learning classifiers and ensembles for early stage prediction of lung cancer. In *2018 3rd international conference on emerging trends in engineering, sciences and technology (ICEEST)* (pp. 1-4). IEEE.
204. Dall'Ora, N., Centomo, S. and Fummi, F., 2019, June. Industrial-iot data analysis exploiting electronic design automation techniques. In *2019 IEEE 8th International Workshop on Advances in Sensors and Interfaces (IWASI)* (pp. 103-109). IEEE.
205. Pimentel, J., McEwan, A.A. and Yu, H.Q., 2022, August. Towards a Real-Time Smart Prognostics and Health Management (PHM) of Safety Critical Embedded Systems. In *2022 25th Euromicro Conference on Digital System Design (DSD)* (pp. 696-703). IEEE.
206. Yang, C.C.Y., Li, G., Liu, X., Wu, Z. and Zhang, K., 2019, October. The Future of Broadband Access Network Architecture and Intelligent Operations. In *2019 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC)* (pp. 308-316). IEEE.
207. Yin, W., Gumabay, M.V.N., Lin, H., Tu, C. and Ao, C., 2022, April. Overhead Transmission Lines Early Warning and Decision Support System with Predictive Analytics. In *2022 5th International Conference on Advanced Electronic Materials, Computers and Software Engineering (AEMCSE)* (pp. 310-314). IEEE.

208. Catelani, M., Ciani, L. and Venzi, M., 2018. RBD model-based approach for reliability assessment in complex systems. *IEEE Systems Journal*, *13*(3), pp.2089-2097.
209. Abdul Manan, M.S., Abang Jawawi, D.N. and Ahmad, J., 2021, September. A systematic literature review on test case prioritization in combinatorial testing. In *Proceedings of the 5th International Conference on Algorithms, Computing and Systems* (pp. 55-61).
210. Sun, H., Sun, J., Wang, L., Liu, X., Chen, Y. and Song, J., 2022, October. Research on Feature Extraction of CSP-based Channel Wave Signal. In *2022 IEEE 6th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)* (pp. 778-783). IEEE.
211. Mach, J., Kohútka, L. and Čičák, P., 2023, June. A New RISC-V CPU for Safety-Critical Systems. In *2023 12th Mediterranean Conference on Embedded Computing (MECO)* (pp. 1-4). IEEE.
212. Nguyen, H., Kamel, M., Alexis, K. and Siegwart, R., 2021, June. Model predictive control for micro aerial vehicles: A survey. In *2021 European Control Conference (ECC)* (pp. 1556-1563). IEEE.
213. Henze, D., Gorishti, K., Bruegge, B. and Simen, J.P., 2019, December. Audioforesight: A process model for audio predictive maintenance in industrial environments. In *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)* (pp. 352-357). IEEE.
214. Climente-Alarcon, V., Arkkio, A. and Antonino-Daviu, J., 2018. 2-D magnetomechanical transient study of a motor suffering a bar breakage. *IEEE Transactions on Industry Applications*, *54*(3), pp.2097-2104.
215. Guan, D., Wei, H., Yuan, W., Han, G., Tian, Y., Al-Dhelaan, M. and Al-Dhelaan, A., 2018. Improving label noise filtering by exploiting unlabeled data. *IEEE Access*, *6*, pp.11154-11165.
216. Su, F., Goteti, P. and Zhang, M., 2020, November. Unleashing the Power of Anomaly Data for Soft Failure Predictive Analytics. In *2020 IEEE International Test Conference (ITC)* (pp. 1-10). IEEE.
217. Shuvro, R.A., Das, P., Hayat, M.M. and Talukder, M., 2019, October. Predicting cascading failures in power grids using machine learning algorithms. In *2019 North American Power Symposium (NAPS)* (pp. 1-6). IEEE.
218. Sang, W., Yuan, S., Yong, X., Jiao, X. and Wang, S., 2020. DCNNs-based denoising with a novel data generation for multidimensional geological structures learning. *IEEE Geoscience and Remote Sensing Letters*, *18*(10), pp.1861-1865.
219. Domingos, J., Cerveira, F., Barbosa, R. and Madeira, H., 2023, June. Predicting Cloud Applications Failures from Infrastructure Level Data. In *2023 53rd Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W)* (pp. 9-16). IEEE.
220. Choudhury, A. and Sikdar, B.K., 2019, November. Soft error resilience in chip multiprocessor cache using a markov model based re-usability predictor. In *2019 IEEE 37th International Conference on Computer Design (ICCD)* (pp. 468-476). IEEE.
221. Farias, H.E.O., Rangel, C.A.S., Canha, L.N., Stringini, L.W., Santana, T.A.S. and Nadal, Z.L.I., 2021, August. Battery energy storage systems management in a day-ahead market scenario with transactive energy and private aggregators. In *2021 56th International Universities Power Engineering Conference (UPEC)* (pp. 1-6). IEEE.
222. Vijayan, A., Kiamehr, S., Ebrahimi, M., Chakrabarty, K. and Tahoori, M.B., 2017. Online soft-error vulnerability estimation for memory arrays and logic cores. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, *37*(2), pp.499-511.
223. Abella, J., Perez, J., Englund, C., Zonooz, B., Giordana, G., Donzella, C., Cazorla, F.J., Mezzetti, E., Serra, I., Brando, A. and Agirre, I., 2023, April. SAFEXPLAIN: Safe and Explainable Critical Embedded Systems Based on AI. In *2023 Design, Automation & Test in Europe Conference & Exhibition (DATE)* (pp. 1-6). IEEE.
224. Gava, J., Hanneman, A., Abich, G., Garibotti, R., Cuenca-Asensi, S., Bastos, R.P., Reis, R. and Ost, L., 2023. A Lightweight Mitigation Technique for Resource-constrained Devices Executing DNN Inference Models under Neutron Radiation. *IEEE Transactions on Nuclear Science*.
225. Apiletti, Daniele, Claudia Barberis, Tania Cerquitelli, Alberto Macii, Enrico Macii, Massimo Poncino, and Francesco Ventura. "istep, an integrated self-tuning engine for predictive maintenance in industry 4.0." In *2018 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Ubiquitous Computing &*

Communications, Big Data & Cloud Computing, Social Computing & Networking, Sustainable Computing & Communications (ISPA/IUCC/BDCloud/SocialCom/SustainCom), pp. 924-931. IEEE, 2018.

226. Li, C., Zigerelli, A., Yang, J., Zhang, Y., Ma, S. and Guo, Y., 2018. A dynamic and proactive GPU preemption mechanism using checkpointing. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, *39*(1), pp.75-87.
227. Li, T., Choi, M., Guo, Y. and Lin, L., 2018, December. Opinion mining at scale: A case study of the first self-driving car fatality. In *2018 IEEE International Conference on Big Data (Big Data)* (pp. 5378-5380). IEEE.
228. Wu, X.X., Hung, Y.W., Chen, Y.C. and Chang, S.C., 2020, March. Accuracy tolerant neural networks under aggressive power optimization. In *2020 Design, Automation & Test in Europe Conference & Exhibition (DATE)* (pp. 774-779). IEEE.