A Detailed Survey on Machine Intelligence Based Frameworks for Software Defect Prediction

Mr..Raghvendra Omprkash Singh [a], Dr. Blessy Thankachan [b]
[a] Research ScholorDepartment of Computer and Systems SciencesJaipur National University,Jaipur Inida..
[b]Assistant Director and Guide,Department of Computer and Systems SciencesJaipur National University,Jaipur Inida..
Email id :[a] raghvendrasingh@live.in, [b]scss_jnu@jnujaipur.ac.in

Article History: Received: 10 January 2021; Revised: 12 February 2021; Accepted: 27 March 2021; Published online: 28 April 2021

Abstract: Software defect prediction has an important role to play in improving the quality of programming and helps to reduce the time and cost of programming testing. AI focuses on the advancement of computer programs that can be instructed to develop and change at a time when new information is presented. The capacity of a machine to improve its exposure depends on past results. Machine learning improves the productivity of human learning, finds new things or structures that are obscured to people, and discovers important data in the archive. For this reason, distinctive machine learning procedures are used to remove unnecessary, incorrect information from the data set. Software defect prediction is seen as an exceptionally significant capability when a product project is arranged and a much larger effort is expected to address this intricate issue using product measurement and deformity dataset. Metrics are the link between the mathematical value and are subsequently applied to the product for anticipation of deformity. The essential objective of this study paper is to comprehend the existing strategies for foreseeing programming deformity.

Keywords: defect prediction, machine learning, Software systems

1. Introduction

Software systems have gradually become confusing and flexible these days. It is critical to persistently distinguish and abandon the right programming plan. Subsequently, anticipating precisely whether a product element contains configuration plains can help to improve the nature of the product frameworks.

The software metric is part of some programming property. It has estimated the nature of the software during product promotion stages , for example, plan or encoding. In a programming item, a software deformity is a requirement not fulfilling the product requirement or then again ending customer. A defect is, therefore, an error of code or reasoning that causes a programme to fail or to achieve surprising results. The forecast of programming imperfections is the way to find defective programming modules. The eventual results should not have as much imperfections as reasonably expected to produce great programming In order to identify programme defects early, the cost of advancing can also decrease quickly, revised exercises and stronger software. The defect forecast is therefore essential to achieve software quality. The main function in building a measurable prediction model is deformity forecast measurements. Most measurements of error can be organised in two kinds: codes and measurements. At the beginning of software development advancements, the expectation models could then be used by the product associations to recognise defect modules. This sub-set of measurements can be used in the software development associations among the large, accessible range. These measurements can be used to develop model expectations of error. Many scientists have used various strategies to connect measurements of the static code with expectations of error. Such policies incorporate customary measurable strategies such as calculated relapse and machine learning policies, such as decision trees, Naive Bays, Support vector machines, artificial neural networks. These strategies include: The artificial neural network seeks a series of loads which can model data to limit the mean dividing line between the class expectations of the organisation and the actual type of information tuple. Backing Vector Machine modifies the unique information in a higher level, where a hyperplane for the division of the information can be discovered using fundamental tuples. The tree of choice plays the tree structure activity. It also uses a feature selection measure to select the attributes tested in the tree for each non-blue node.

2.Literature Review

This section will discuss the related work of the study. (Oladimeji et al. 2019), attempts to inspect existing insider threat detection techniques and also provides machine learning as an insider threat position route forward. The challenge from the insider is a big problem for any relationship. A repeated study challenge has been developed to resolve the issue through a strong relief method. In comparable queries, such as Anomaly Detect (AD) and Network Intrusion Detection (NID), machine learning Strategies have been proposed as the solution.

(Homoliak et al.2020)provide data systematization for in-depth threat analysis, using current dependent theory techniques for systematic written examination. This research will boost the efforts of specialists of insider threat, as it provides new real scientific auxiliary categories adding symmetrical characteristics and the degree to which protections are used against incidents, an analysis of publicly available databases which can be used to assess various works for new locational arrangements , Descriptions to the latest contextual inquiries and frameworks revealing the practice of insiders for sample and inclusion steps, and discussion of existing trends and supplementary headings for insider risk thinking.

(Lu and Wong,2019)Our study will improve analysts' endeavors in the area of insider danger, since it gives: a) a novel auxiliary scientific categorization that adds to symmetrical arrangement of episodes and characterizing the extent of guard arrangements utilized against them, b) an outline on freely accessible datasets that can be utilized to test new identification arrangements against different works, link to present system and frameworks of insiders' activities showing surveys or extending their participation, and discussion of current trends and further exam headings which can be used for insider threat thinking.

The after- effect of exploring an exam indicates that using the operational and implementation layer information channel is more acceptable than that of the visual layer in the IoT environment. (Kim et al,2020)Speculation is undertaken on IoT parts of the insider threat and explores the studies focused on private as well public sources.

In order to provide a point by point interpretation of this topic (Atlam et al. ,2020)a thorough survey and examination of the best in class of risky admittance management model. 44 articles (1044 articles) have been selected for closer examination in terms of this search method. The obligations of the papers selected have been summarized from these papers. In addition , the risk factors for constructing the unsafe entry model have been separated and investigated. Furthermore, risk management techniques have been recognised in order to determine risks of access control activities.

The insider threat detection (Sheykhkanloo and Hall,2020) addresses an extraordinarily imbalanced dataset which uses a major-stakeholder adjustment technique called a spread-sub-sample. The findings show that despite the fact that modifying the dataset using this technique did not change the output measurements, the time needed to construct the model and the time taken to validate the model were increased. The developers have recognised that running the chosen classifiers with boundaries other than the norm effects both balancing and imbalanced conditions, but the impact is generally more significantly dependent on the imbalanced dataset.

(Dam et al. 2019)report on the ability to transmit a new model of deformity based on a deep learning tree.This model is based on the tree-built long term memory network that co-ordinates conveniently with the source code portrait of the Abstract Syntax Tree.

Filter feature ranking (FFR) and fourteen philtre feature subset selection (FSS)strategies have been analysed using more than five separate classifier data sets retrieved from the NSA archive, including 4 single classifiers. In(Balogun et al . 2019), four channel programmes have been analysed using four unique classifiers.The research has shown that the use of FS enhances the prestigious presentation of classifiers and the display of FS strategy can change across datasets and classifiers. In the FFR techniques, the best improvements in the presentation of forecast models were demonstrated by Knowledge Gain. In FSS strategies, collection of accuracy sub-sets based on the best first quest had the best effect on the anticipation models.

(Kaloudi and Li,2019)aims to analyse current machine learning-based digital attack inquiries and to prepare them for a planned method to understand new threats. The system contains a few pieces of machine learning 's dangerous usage during the digital loop of attacks and offers a premise for its detection so as to predict potential risks.

The late diagram for use by neural networks, including analysis and new policy suggestions (Drewek-Ossowicka, Mariusz Pietrołaj and Rumiński,2019), offers a detailed diagram of late writing in the interrupting locational region. In comparison, brief instructional activity representations of the designs of the neural network, interference positioning framework forms and data sets.Neural network structures are broadly utilized for making new models.

Contains k-nn KDD Cup 99 Another illustrate ML depiction strategy focused on group concentration and nearest neighbour approach. Lin et al. (2015) 214 Cluster position and nearest neighbour (CANN). Lin et al . (2015).

David and Netanyahu (2015) 61DBN Custom malware bases Ingre and Yadav (2015) 48NNN NSL-KDD Execution assessment using NN-related technique for double- and five-class clustering in the attack form for each NN malware signature recognition strategy.

(Taj et al.2020)Use the Anomaly Based Intrusion Detection System (ABIDS) to watch and interpret exercises which in dispersed computer conditions are not part of the authentic traffic organisation using VMs. The method suggested tracks the traffic of VMs, philtres and recognises the particular manner in which irregular parcels are handled and thus manages net traffic.

Submit an organised appraisal protocol, which covers defining techniques, safeguarded arrangements, execution devices, test requirements and IDS execution (Rakas, Stovajanic and Markovic Petrovic 2020). The assessment of usage growth is based on an exceptional consideration, as is the importance of any overviewed scheme in the future Internet environment.The final assessment will be undertaken in conjunction with the previous review, including strengths, limitations, growth and flexibility for FIN environment. The consequences of our investigation in relation to similar work indicate critical advances in the creation and development of new intrusion detection techniques (in addition to machine learning , the use of open source equipment and the development of modern test beds.

(Legg et al. 2020)propose an insider threat system that transcends traditional mechanical perceptions and takes a closer view of insider risks, daily history, and individual actions and behaviour.The model consolidates a thought system that can alert an inspector about estimates based on true expectations, which helps the specialist to detail hypotheses which examine the potential about insider hazards. In order to preserve the adaptability for the modules to be fused at a later stage, the technical model has been established with regard to elements of common sense for application, meaning that the model is well verified for the future.

(Allen et al. ,2015))The mechanism for the distinction between insider dangers in an enterprise has been identified. The system provides the organisation with knowledge that can arrange action and concentrates observable operations from information specific to a task to distinguish the insider threat. The discernible actions are joined together to achieve rational and reflective outcomes.

(Myers, Grimaila and Mills,2014) discuss our underlying analysis activities at the position of deceptive insiders that harass authoritative internet staff. The purpose of the study is to use activities that the executives will learn in network inspection spaces, to examine various methodologies to classify insider threat activities using common equipment and a traditional articulation framework of functions.

(Glasser and Lindauer,2013)to provide an enhanced overview of our overall approach to integrate a section of the problems and exercises found on the use and misuse of information in engineering, in particular with regard to credibility and job importance in the information generated, and to make use of open doors for further inquiry.

(Nostro et al., 2014)defines the inspirations and the aims of an insider, explores the similarities and gravity of potential infringements, and eventually determines effective counter-measures. The strategy also involves a support period in which the evaluation can be refreshed to represent improvements in the framework.

(Hu et al . ,2019) suggest a consumer confirmation methodology based on mice's bioconductual qualities and profound performance that can correctly and productively execute continuous validation of character on current PC clients in order to counter risks to the insiders in these fields..The test results demonstrated that our proposed strategy could recognise the client's personalities in short order and has a false accept rate (FAR)and false reject rate (FRR).

(Representative et al. ,2013)reports to define, organise and evaluate new directions in which fragiles typical signals for insiders dangers on data frames in collaboration, are established and established by the community comprising the SAIC and four schools, on strategies and implications of the applied exploration undertaking. The platform consolidates basic and semantic data from an authentic organisational knowledge base on the computers of the customs so that noxious insider exercises can be detected instantly by a Red Party.

The test relied on 30-redundancy holdout authorisation and 10 * 10 cross-approval and suggested (Dish et al. ,2019)a improved CNN model for inside-control imperfection (WPDP), and compared our findings with the present results of the CNNs and an empirical analysis. Exploratory findings found that our revamped CNN model is comparable to the existing CNN model and basically graded WPDP against the cutting edge machine learning models.

A Novel SDP model, known as "Siamese equivalent fully connected networks" (SPFCNN) is proposed for this problem in (Zhao et al . , 2019) which joins upsides of Siamese organisations, and deep learning in a combined technique. In addition, AdamW measurement is used for preparing this model to find the right loads. A single recette's basis approximation is the planning of the SPFCNN process. This process. Basically, SPFCNN and the best class SDP approaches are contrasted with six readily available NASA archive data sets.

(Ifthikar et al . ,2018) reviewed several exploration papers that used SC methods mainly at risk to managers and to promote programming, programming consistency, software dependance and the board's mission.It has been observed that fuzzy logic, false neural structure and genetic algorithms are generally used.

Greitzer et al. ,2014)examine cases of UIT derived from social adventure creation. Reports are analysed to compile and dissect information from instances of UIT social design in order to recognise conceivable activities and advanced examples and to shed light on future novel UIT relaxation approaches.

(Rich et al. ,2005)report on the work to distinguish apparently sensible hierarchical activities that may unintentionally lead to increased risk exposure. Two intertwined work items are introduced: a contextual analysis that presents a specific type of internal hazard – long-haul extortion – and a recreation model that underlies the case, a fundamental unique hypothesis, and an evaluation of the strategy alternatives case and the model join forces to deliver a persuasive and valuable exercise that addresses the issues of insider digital threats.

(Elmrabit, Yang and Yang, 2015)provide a diagram of the different fundamental attributes of insider threat. Also, a specialised methodology without anyone else may not be the best method for preventing and distinguishing harmful insider threats.

(Koroglu et al . ,2013) present our methodology as a report on our experience.aIn particular, information from seven more seasoned variants of the product venture is collected and used extra features to anticipate flow version imperfections. The results show that the test ef-posts re focused by directing the test group to only 8 per cent of the product where 53 per cent of the actual deformities can be found. The model is 90 percent accurate.

(Li et al. ,2017)e propose a novel Ensemble Multiple Kernel Correlation Alignment (EMKCA) based on a way to deal with HDP, which looks at the two attributes of imperfect expectations information. In particular, first the source is guided and target threat data to high-dimensional portion space through various bit tilts, where faulty and undamaged modules can be better isolated.

(Ifthikar et al. ,2018)surveyed a few exploratory articles in which researchers used SC strategies for the most part in executive threats, maintenance programming, quality programming, reliability programming and the board of directors. It has been shown that fuzzy logic , artificial neural network and genetic algorithm are generally used. Bayesian Networks, Rough Set Hypotheses and Ant Colony Optimization are used by not many scientists who have prospects for future exploration.

(Pattnaik and Pattanayak, 2016)conduct a broad overview of different AI methods, such as fuzzy logic , neural network and Bayesian model, and so on, used for quality forecast programming alongside diagnostic protection for each of the proposed solutions. The noteworthy quality expectation programming during the improvement cycle has provided the inspiration for this review.

In(Zhao et al. ,2016), a novel SDP model is proposed for this issue, called Siamese Equal Completely Associated Networks (SPFCNN), which consolidates Siamese networks and deep learning into a combined technique. What's more, the preparation of this model is controlled by the AdamW algorithm to find the best loads. The basic estimation of a specific equation is the objective of the preparation of the SPFCNN model.

(Pandey et al. ,2017)analyze how machine learning methods could be used to perform this assignment. Distinctive order algorithms are applied , specifically credulous Bayes, straight discriminant examination, k-closest neighbours, support vector machine (SVM) with different parts, choice tree and irregular forest independently to characterise the reports of three open-source ventures. Their presentation is assessed to the extent of F-measuring, normal precision and weighted normal F-measuring. Thetests show that irregular woods perform best, while SVM with specific bits also performs superior.

(Okutan and Yıldız,2012)use Bayesian networks to decide on the probabilistic, compelling linkages between programme measurement and imperfection inclination. Notwithstanding the measurements used in the Promise Information Store, two additional measurements are characterised , e.g. the Design Quantity and LOCQ for the quality of the source code. Towards the end of our display, we learn about the marginal deformity inclination probability of the entire programming framework, the arrangement of the best measurements, and the compelling connexions between measurements and imperfection.

(Xu and Zhang,2018) propose a two-stage CVDP system that combines Hybrid Active Learning and Kernel PCA (HALKP) to address these two issues. In the main stage, HALKP uses a hybrid dynamic learning technique to choose some illuminating and delegating non-labelled modules from the current adaptation to question their names, at which point they are consolidated into the marked modules of the earlier form to shape and upgrade the preset.

(Li et al., 2017)comprehensively investigating, dissecting and talking about the best forecast in the imperfection class. The developers review around 70 delegate imperfection expectations papers as of late (January 2014–April 2017), most of which are distributed in the noticeable programming of diaries and high-level meetings. Selected imperfection expectations papers are summarised in four perspectives:Machine-learning -based forecast algorithms, information control, mindful expectation exercise, and experimental investigations. The exploration network is currently facing a number of challenges in the development of strategies and a number of review openings exist. Distinguished difficulties may give rise to some reasonable rules for both the programming expectations of design scientists and experts in future programming imperfections.

(Manjula and Florence,2018) present a hybrid methodology by consolidating genetic algorithms (GAs) to include enhancement of deep neural network (DNNs) for grouping. The improved form of GA is merged, which incorporates a different strategy for chromosome planning and wellness fitness function . DNN strategy is also ad libbed using a versatile auto-encoder that better portrays selected programming features.

(Prasad, Florence and Arya, 2015)o assist engineers in distinguishing absconds dependent on existing programming measurements using information mining strategies and, along these lines, improve the quality of the product. In this paper , different ordering methods are returned which are used for programming imperfection predictions using programming measurements in reading.

(Liang et al. ,2019)proposes Seml, a novel system that combines word implanting and deep learning strategies for predicting deformity. The LSTM model can naturally get acquainted with semantic project data and perform imperfect expectations. The results of the assessment of eight open source ventures show that Seml outperforms three best in class deformity forecast approaches for most datasets for both in-company imperfection expectations and cross-venture imperfection projections.

(Tong et al . ,2018) propose a novel approach to the SDP, SDAESTSE, which takes into account the points of interest of the SDAEs and the gathering of learning , specifically the proposed two-stage troupe (TSE). Technique The strategy mainly consists of two stages: the deep learning stage and the two-stage ensemble (TSE) stage.First, SDAEs is used to separate the DPs from the usual programming measurements, and then a novel learning gathering approach, TSE, is proposed to address the issue of class-awareness. Results Experiments are performed on 12 NASA datasets to demonstrate the adequacy of DPs, proposed TSEs, and SDAEsTSEs separately.

3.Comparative Analysis

This section will comprise the comparative analysis of the different techniques in a tabular form. The comparison can be made with respect to the objectives, the approach, techniques being used, limitations and research gaps

Reference no	**Objectives**	**Approach**	**Techniques**	**Limitations**	**Research gaps**
1	To develop a model using deep neural network and decision forest for exploring powerful image features	Neural forest (NF)	A model using deep neural network and decision forest	It can not handle the functionality classification tasks in software engineering such as defect prediction, software code review, discovery of software vulnerability, malware detection.	Future works need to handle functionality classification tasks in software engineering such as defect prediction, software code review, discovery of software vulnerability, malware detection.
2	To develop a novel Deep Learning based method predicting Software Maintainability Metrics.	Deep Learning based method predicting Software Maintainability Metrics	Deep Learning for Software Maintainability Metrics prediction.	The correct measurement of accuracy is lacking.	Future work includes usage of a fluffy deep neural system, and investigating whether the accuracy of the results achieved using this method is comparable to the results achieved through Other methods.-Other methods.

3	To find approximate solutions for search and optimization problems.	Artificial Neural Network based software fault prediction technique is used.	Artificial Neural Network based software fault prediction technique is used.	Since the link weights are initialised randomly, the neural network results in different data sets	Improvement is needed when the link weights are randomly initialised, the neural network gives different dataset results.
4	To develop an Adaptive dimensional biography Related optimization of the RBFNN classifier model	An Adaptive dimensional biography Related optimization	An Adaptive dimensional biography Related optimization of the RBF classifier model	The effects of false positive and false negative costs is not discussed much.	Future work needs to discuss the effects of false positive and false negative costs.
5	To develop the newDeep Learning Tree Defect Prediction Model	To develop the newDeep Learning Tree Defect Prediction Model	Deep Learning Tree Defect Prediction Model	It does not predict such defect forms including security weakness and code security-critical hazards.	In future defect forms including security weakness and code security-critical hazards need to be addressed.

4.Challenges In Software Defect Prediction

Since it was first proposed, the field of software defect prediction has been well studied. As the software defect prediction framework realised that accessibility and receptivity of information is a critical element for its wealth, many deformity forecasts started to exchange information and also to study the scripts. The software defect prediction area has advanced.As the software defect prediction network realised that the accessibility and receptiveness of information were a key factor for its prospects, numerous software defect prediction studies began to share their details, but even their scripts for examinations have also been examined.The thought behind utilizing measure measurements in imperfection forecast is that the cycle used to build up the code may prompt imperfections, subsequently the cycle measurements might be a decent marker of imperfections

5.Conclusion

This paper provides an overview of various machine learning procedures for the software defect predication. The survey indicates that software defects are definitely a big problem in the design of programming. Imperfect programming module projection using distinct machine learning procedures is designed to enhance the nature of programming development. The software manager distributes assets through this process. We analysed the preferences and constraints of Artificial Vector Machine, decision tree, Association rule and Clustering AI methods in the prevision of deserts. We addressed the preferences and restriction

References

1. Karim, S., Warnars, H. L. H. S., Gaol, F. L., Abdurachman, E., & Soewito, B. (2017, November). Software metrics for fault prediction using machine learning approaches: A literature review with PROMISE repository dataset. In 2017 IEEE International Conference on Cybernetics and Computational Intelligence (CyberneticsCom) (pp. 19-23). IEEE.
2. Qiu, Y., Liu, Y., Liu, A., Zhu, J., & Xu, J. (2019). Automatic Feature Exploration and an Application in Defect Prediction. IEEE Access, 7, 112097-112112.
3. Jha, S., Kumar, R., Abdel-Basset, M., Priyadarshini, I., Sharma, R., & Long, H. V. (2019). Deep learning approach for software maintainability metrics prediction. Ieee Access, 7, 61840-61855.
4. Mundada, D., Murade, A., Vaidya, O., & Swathi, J. N. (2016). Software fault prediction using artificial neural network and Resilient Back Propagation. International Journal of Computer Science Engineering, 5(03).
5. Kumudha, P., & Venkatesan, R. (2016). Cost-sensitive radial basis function neural network classifier for software defect prediction. The Scientific World Journal, 2016.
6. Laradji, I. H., Alshayeb, M., & Ghouti, L. (2015). Software defect prediction using ensemble learning on selected features. Information and Software Technology, 58, 388-402.
7. Dam, H. K., Pham, T., Ng, S. W., Tran, T., Grundy, J., Ghose, A., ... & Kim, C. J. (2019, May). Lessons learned from using a deep tree-based model for software defect prediction in practice. In 2019 IEEE/ACM 16th International Conference on Mining Software Repositories (MSR) (pp. 46-57). IEEE.
8. Balogun, A. O., Basri, S., Abdulkadir, S. J., & Hashim, A. S. (2019). Performance analysis of feature selection methods in software defect prediction: a search method approach. Applied Sciences, 9(13), 2764.
9. Yohannese, C. W., & Li, T. (2017). A combined-learning based framework for improved software fault prediction. International Journal of Computational Intelligence Systems, 10(1), 647-662.
10. Shepperd, M., Bowes, D., & Hall, T. (2014). Researcher bias: The use of machine learning in software defect prediction. IEEE Transactions on Software Engineering, 40(6), 603-616.
11. Li, J., He, P., Zhu, J., & Lyu, M. R. (2017, July). Software defect prediction via convolutional neural network. In 2017 IEEE International Conference on Software Quality, Reliability and Security (QRS) (pp. 318-328). IEEE.
12. Li, M., Zhang, H., Wu, R., & Zhou, Z. H. (2012). Sample-based software defect prediction with active and semi-supervised learning. Automated Software Engineering, 19(2), 201-230.
13. Malhotra, R. (2015). A systematic review of machine learning techniques for software fault prediction. Applied Soft Computing, 27, 504-518.
14. Lu, H., Kocaguneli, E., & Cukic, B. (2014, November). Defect prediction between software versions with active learning and dimensionality reduction. In 2014 IEEE 25th International Symposium on Software Reliability Engineering (pp. 312-322). IEEE.
15. Shao, Y., Liu, B., Wang, S., & Li, G. (2018). A novel software defect prediction based on atomic class-association rule mining. Expert Systems with Applications, 114, 237-254.
16. Rhmann, W., Pandey, B., Ansari, G., & Pandey, D. K. (2020). Software fault prediction based on change metrics using hybrid algorithms: An empirical study. Journal of King Saud University-Computer and Information Sciences, 32(4), 419-424.
17. Wang, Z., & Srinivasan, R. S. (2017). A review of artificial intelligence based building energy use prediction: Contrasting the capabilities of single and ensemble prediction models. Renewable and Sustainable Energy Reviews, 75, 796-808.
18. Kakkar, M., & Jain, S. (2016, January). Feature selection in software defect prediction: A comparative study. In 2016 6th International Conference-Cloud System and Big Data Engineering (Confluence) (pp. 658-663). IEEE.
19. Niu, Y., Tian, Z., Zhang, M., Cai, X., & Li, J. (2018). Adaptive two-SVM multi-objective cuckoo search algorithm for software defect prediction. International Journal of Computing Science and Mathematics, 9(6), 547-554.
20. Pan, C., Lu, M., Xu, B., & Gao, H. (2019). An Improved CNN Model for Within-Project Software Defect Prediction. Applied Sciences, 9(10), 2138.
21. Anbu, M., & Mala, G. A. (2019). Feature selection using firefly algorithm in software defect prediction. Cluster Computing, 22(5), 10925-10934.
22. Ghaffarian, S. M., & Shahriari, H. R. (2017). Software vulnerability analysis and discovery using machine-learning and data-mining techniques: A survey. ACM Computing Surveys (CSUR), 50(4), 1-36.
23. Koroglu, Y., Sen, A., Kutluay, D., Bayraktar, A., Tosun, Y., Cinar, M., & Kaya, H. (2016, May). Defect prediction on a legacy industrial software: A case study on software with few defects. In 2016

IEEE/ACM 4th International Workshop on Conducting Empirical Studies in Industry (CESI) (pp. 14-20). IEEE.
24. Li, Z., Jing, X. Y., Zhu, X., & Zhang, H. (2017, September). Heterogeneous defect prediction through multiple kernel learning and ensemble learning. In 2017 IEEE International Conference on Software Maintenance and Evolution (ICSME) (pp. 91-102). IEEE.
25. Iftikhar, A., Musa, S., Alam, M., Su'ud, M. M., & Ali, S. M. (2018, May). A survey of soft computing applications in global software development. In 2018 IEEE International Conference on Innovative Research and Development (ICIRD) (pp. 1-4). IEEE.
26. Zhao, L., Shang, Z., Zhao, L., Zhang, T., & Tang, Y. Y. (2019). Software defect prediction via cost-sensitive Siamese parallel fully-connected neural networks. Neurocomputing, 352, 64-74.
27. Pattnaik, S., & Pattanayak, B. K. (2016). A survey on machine learning techniques used for software quality prediction. International Journal of Reasoning-based Intelligent Systems, 8(1-2), 3-14.
28. Pandey, N., Sanyal, D. K., Hudait, A., & Sen, A. (2017). Automated classification of software issue reports using machine learning techniques: an empirical study. Innovations in Systems and Software Engineering, 13(4), 279-297.
29. Okutan, A., & Yıldız, O. T. (2014). Software defect prediction using Bayesian networks. Empirical Software Engineering, 19(1), 154-181.
30. Xu, Z., Liu, J., Luo, X., & Zhang, T. (2018, March). Cross-version defect prediction via hybrid active learning with kernel principal component analysis. In 2018 IEEE 25th International Conference on Software Analysis, Evolution and Reengineering (SANER) (pp. 209-220). IEEE.
31. Li, Z., Jing, X. Y., & Zhu, X. (2018). Progress on approaches to software defect prediction. IET Software, 12(3), 161-175.
32. Manjula, C., & Florence, L. (2019). Deep neural network based hybrid approach for software defect prediction using software metrics. Cluster Computing, 22(4), 9847-9863.
33. Prasad, M. C., Florence, L., & Arya, A. (2015). A study on software metrics based software defect prediction using data mining and machine learning techniques. International Journal of Database Theory and Application, 8(3), 179-190.
34. Liang, H., Yu, Y., Jiang, L., & Xie, Z. (2019). Seml: A semantic LSTM model for software defect prediction. IEEE Access, 7, 83812-83824.
35. Tong, H., Liu, B., & Wang, S. (2018). Software defect prediction using stacked denoising autoencoders and two-stage ensemble learning. Information and Software Technology, 96, 94-111.