

A Survey of Software Systems Anomaly States Prediction Based on Artificial Techniques

¹Raghda Azad Hassan, ²Ibrahim Ahmed Saleh

¹Student, Department of Software, College of Computer & Math., University of Mosul, Iraq

²Professor, Department of Software, College of Computer & Math., University of Mosul, Iraq

Abstract - Predicting software anomaly is essential for improving software quality and saving costs and time associated with software testing and maintenance in advanced stages life of software development. Complex software systems can be made more reliable and field failures reduced by having the ability to anticipate failures before they occur, as the field of forecasting has witnessed. Software defect prediction (SDP) has seen recent developments, such as combining several classification algorithms to create an ensemble or hybrid approach. Many laterals have been conducted in the field of predicting, including predicting anomalies and disruptions in performance and predicting software errors. This paper presents a study and review of the literature on predicting anomalies in software systems, strategies and methods for detecting them, and a brief overview of predicting defects and future trends in forecasting, as the results of the reviews showed. Because ensemble predictors can, in some cases, enhance bug detection performance, many problems have been solved by machine learning techniques as a result of this recent success.

Keywords: software engineering, ensemble learning, software anomaly detection, Software defect prediction.

I. INTRODUCTION

With the development of technological and increasing need and reliability for complex software systems, that demand for software with excellence and high performance, low costs and high quality, as high performance is a critical factor for achieving successful software systems [1]. This matter has become gradually more important. The cost of defects and malfunctions in software reduces the quality and construction of the program, causing delays in product delivery and negatively affecting user dissatisfaction. Because system failures often occur due to software defects, they are a factor affecting software quality and serve as potential root causes of errors and failures in related systems [2]. Because software quality means an error-free product. Predicting errors in software plays a major role in improving software quality. Early detection helps rapid production and delivery of maintainable software.

Therefore, software defect prediction usually produces data sets, methods, and frameworks that allow software engineers to focus on development activities in terms of defect-prone code, thus improving software quality and making better use of it, and be able to produce predictable results and able to deliver within time and cost constraints. One of the requirements of our time is the increase in high-quality software that has the ability to compete in today's business world, and technological progress, hardware complexity and changing business requirements are necessary [2]. Therefore, anomaly detection techniques are vital techniques in the field of data analysis that include identifying errors, patterns or outliers, and performance anomalies that represent performance degradation problems[1]. The main contribution of anomaly detection is in saving time and effort for software systems.[3] In traditional methods of anomaly detection, traditional monitoring methods are often used through alerts and health check requests for systems. Detecting anomalies in modern and large applications and software that work well, but it is still possible that anomalies and unbearable performance behaviors may sometimes appear. These behaviors are unexpected and undesirable because of their impact on the costs of failure and others, and the process of distinguishing between abnormal behaviors and normal behaviors represents a difficult challenge [4]. These anomalies can be caused by various factors such as errors, malfunctions or fraudulent activities. In order to achieve stringent performance requirements and meet users' expectations, software engineers and system administrators must proactively, effectively and reliably detect undesired performance behaviors, as predicting failure before it occurs is an essential factor in reducing failures, identifying potential root causes and taking appropriate corrective measures [5]. Therefore, delays in detecting anomalies and undesirable behaviors cause systems failure, performance problems, service failures, and thus financial losses [1]. The current framework for predicting defects revolves around three implications: first, classifying the extent of vulnerability of software components into two categories (prone to failure or not prone to failure), second, identifying relationships and connections between defects, and third, estimating and estimating remaining defects in systems [6]. Therefore, they

proposed an approach to predict anomalies as soon as they occur, which helps with other systems in classifying fault-prone and non-fault-prone units and early detection using ensemble learning methods. Due to the size of large and complex software systems in which anomalies cannot be predicted manually, it is known that machine learning techniques help in predicting the location of defects [6].

Building an ensemble learning model achieves consistency in prediction with high accuracy by collecting numerous machine learning classifiers to improve the prediction process. The aim of using the collective approach is to overcome the restrictions that apply to individual works. Many researchers have indicated that ensemble learning methods provide higher classification accuracy than individual classifiers [7].

Anomalies can be defined as unintended situations that have a negative impact on the operation of the system in its normal state. These conditions may have a fatal impact on the system or remain in the system for a certain period without being noticed. In both cases, it is better to predict and detect them quickly [4]. The reset of paper in section 2.

1.1 Performance abnormalities and their types

Performance anomalies often appear on a graph as a point or group of data points that serve as discrete measurements of a performance metric, shown in Figure 1. This set of points usually lies outside the normal area. Researchers Olumuyiwa et al have identified three main types of faults that cause service failures performance. Because point anomalies, ensemble anomalies, and contextual anomalies, when plotted, performance measures often show distinct shapes when the resource is in a saturated state. Therefore, researchers introduced another type of anomaly, typical anomalies [5].

- a) The anomalous point is the point that lies outside the expected area. A deviation that occurs randomly without being related to the pattern.
- b) A group (or collective) anomaly is a group of data points that fall outside the expected area.
- c) Contextual anomalies: The contexts or environments of performance anomalies can be identified through levels (moderates, high, burst).
- d) The anomaly (pattern) can be considered a form of collective anomaly because it consists of a group of points. When performance metrics are plotted to identify abnormal behavior, certain patterns emerge.

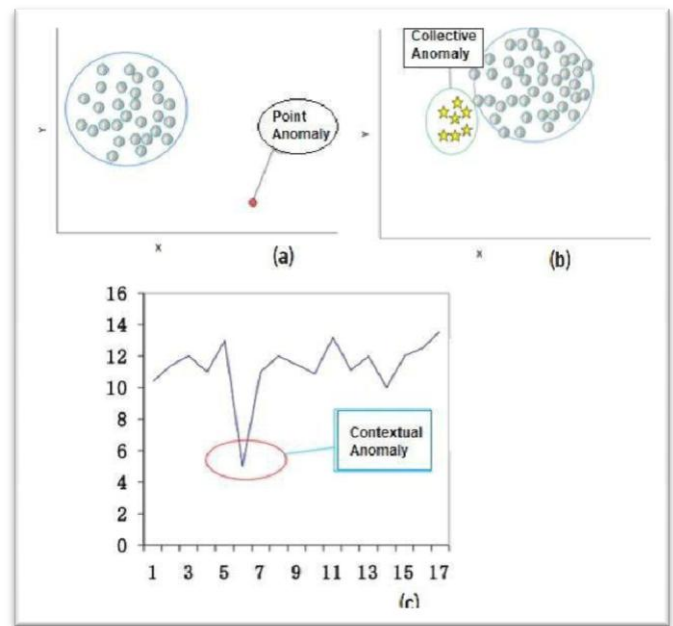


Figure 1: examples of types of anomalies

1.2 The process of predicting anomalies

The process of developing software projects requires a comprehensive and expensive testing phase. With the increase in the size and complexity of software, manual prediction becomes difficult. Therefore, the goal of the process of predicting software anomalies is to predict software units that are vulnerable to error or failure. Early detection of anomalies helps reduce the efforts that are applied. In the later stages of software development [5] [8].

1.2.1 Mechanisms for detecting and preventing anomalies:

Prediction models are usually trained using some basic characteristics of the project. The process of predicting software errors can be explained by saying that if a project is developed in a specific environment and produces faulty results, the use of any module developed in a similar environment with similar characteristics can lead to producing a defective project. One error can lead to several errors or product failure.

To avoid software failure, error detection mechanisms are usually used based on their importance for each stage of the software product development life cycle (i) detection using automated static analysis and (ii) detection using mining. (iii) Diagram (detection using classifiers) (iv) Detection using pattern mining. The process of detecting anomalies is the process of improving the quality of the software product and reducing costs. The majority of software development teams focus on detecting them.

1.2.2 Fault Classification and Analysis:

To determine the type and group of faults, the Orthogonal Defect Classification (ODC) technique can be used to classify faults. When a fault first arises and is corrected, it is classified using the ODC technique. The purpose of Root Cause Analysis (RCA) is to find the source of errors and defects and take appropriate action after eliminating them. The errors are analyzed one by one to achieve this [2].

II. SOLUTION STRATEGIES AND METHODS

In general, detecting performance issues and errors in software systems involves an approach that requires continuous estimation (prediction) of the normal behavior of the system at specific points. Some methods such as learning methods and statistical methods are used to achieve the goals of detection and prediction. In performance studies, machine learning algorithms are used to identify patterns of interests and relationships that are not obvious. As shown in figure (2), Machine learning algorithms are classified into four categories according to inputs and outputs. Expected (1) Supervised learning includes regression and classification because it is a method that involves the use of trained classification data, and it is the most common machine learning method. Among its

common examples are (a) neural network (b) support vector machine (c) linear regression (d), Bayesian learning (2) Unsupervised learning, which has another name, learning from observation, as this method works without the need for labeled or trained data, and it groups data into categories based on their statistical properties.

A common example of it is (a) sequential ensemble mining (b) and (c) association rule (3) To make the most of supervised and unsupervised learning, a hybrid approach is semisupervised learning, which is a new strategy that achieves performance Higher by taking advantage of the presence of confidential data. According to the semi-supervised algorithm, a small part of the dataset is classified as normal, and the remaining unlabeled examples in the dataset are considered anomalous. Because they use labeled data to find an innate pattern in the data, they often outperform their supervised and unsupervised counterparts [5]. The development of different learning algorithms that can be used in a variety of situations has led to the rapid expansion of machine learning research. Moreover, the ultimate usefulness of machine learning algorithms is often determined by their ability to deal with real-world problems, and thus the reproducibility and application of these algorithms new tasks are essential to advance in this field [6].

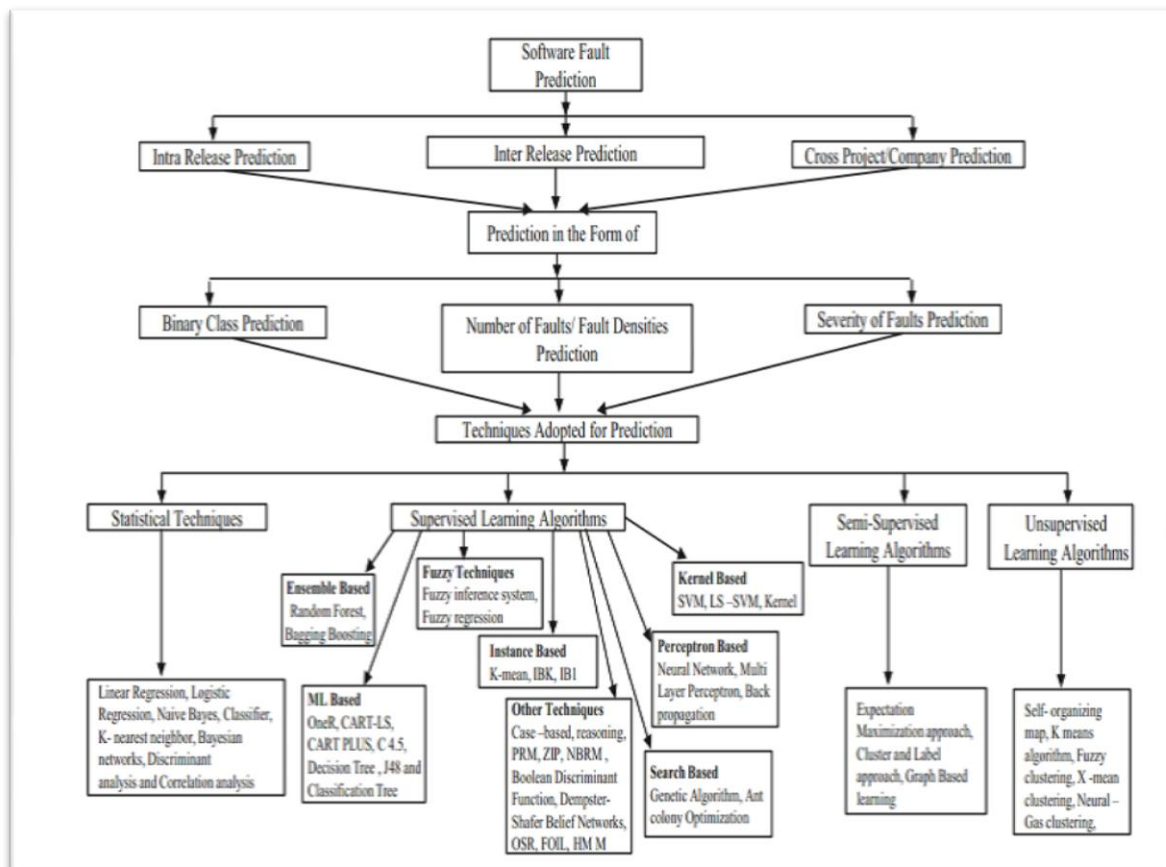


Figure 2: Classification of software fault prediction strategies and techniques

III. RELATED WORK

There are many researchers were interesting about anomaly detection, it is a proactive process for programs that are in their normal state before they fall into anomalous situations. Some of the them in below.

In (2012) Donghun Lee et al [9] they developed a framework for managing performance anomalies where they used statistical process control (SPC) charts to detect performance anomalies. Tasks including differential profiling were automated to identify and find the root causes of anomalies, reducing tedious manual work and reducing the cost and time of detection. Analysis, where they were able to get rid of them by about 90 percent, relying on the agile development methodology in the development process to maintain the source code so that it is ready to work. They created 20 data sets randomly, each as a group consisting of 60 randomly, that is, they used ready data, and to address implementation problems, CUSUM charts were used, which It is considered one of the SPC schemes.

In the (2017) Santosh S. Rathore et al [10] presented a study on predication of software anomaly regarding: data quality concerns, software metrics, bug prediction strategies, and performance evaluation metrics. This study identified and analyzed several activities and their impacts on prediction performance in order to focus on the many aspects of software bug prediction. The purpose of this study is to examine various problems related to software bug prediction and help in understanding the different components of the bug prediction process. He summarized the current research and review and presented its benefits and drawbacks in a tabular form. The analysis showed that process metrics with data and object-oriented(OO) metrics accounted for the majority of the work. The majority of publications have used precision, precision, and recall to evaluate the performance of general and statistical procedures.

In (2018), researchers Ashraf Sayed and Naji Ramadan [11] explained that predicting software defects in the advanced stages of the software life cycle, which depends on collective learning, produces successful software. Through their study, the researchers confirm that there is not enough evidence to say that any particular type of machine learning techniques is more effective or accurate in predicting software defects. As a more precise alternative, they suggest group learning techniques. They present a resampling technique that uses eight baseline learners evaluated on seven different types of benchmark datasets available in the PROMISE repository. The three types of learners used are boosting, bagging, and rotation forest. The results showed that the combined methods enhanced accuracy more than the individual methods,

especially when combined with Rotation Forest and the resampling technique in the majority of algorithms used in the experimental results.

In (2019), Faseeha Matloob et al [12] presented a framework for predicting software defects using ensemble learning consisting of four stages. The first stage is data collection, where 6 data sets were used from the NASA repository and a set of pre-processing operations were applied to them. The second stage includes normalization, balancing, feature selection, and selection. Features They use Wrapper approach and in the later stage six search methods are implemented including Best First (BF), Greedy Stepwise (GS), Genetic Algorithm Search (GA), Particle Swarm Optimization (PSO), Rank Search (RS) and Linear Forward Selection (LFS). The performance is evaluated using precision MCC and ROC, where the results showed that the performance of the proposed method outperforms all individual classifiers. In (2020), YUE TAN et al [13] used a deep learning method called long-short memory (LSTM) to predict anomalous positions in nonlinear dynamic systems using the local averaging with adaptive parameters (LAAP) algorithm on the (wall shear stress) dataset. They used a sliding window system to collect training samples, and experimental results showed the superiority of this approach. In terms of high performance and accuracy compared to traditional methods, the proposed method achieved multi-step predictions with low root mean square error (RMSE).

In (2020) GUOLIANG ZHAO and other[1] researchers proposed an approach that can predict performance anomalies in software systems and raise anomalous warnings in advance. The experimental results were that approach achieves high performance in predicting anomalies in general, using a neural network with long shortterm memory (LSTM) to capture system behaviors. To evaluate the performance of the proposed approach, data collected from two software systems (HadoopMapReduce and Elaticsearch) were adopted, as their method excels in Performance on baselines and predicts various performance anomalies with 97-100% accuracy and - 80% recall. 100% .

In (2021), FASEEHA MATLOB and others researchers [7] explained, through their study and survey of the literatures that they adopted on the use of ensemble learning approaches to prediction since 2012, that they confirmed that the ensemble approach to predicting software defects enhances the prediction performance by removing the limitations imposed on a single classifier and enhancing the performance of ensemble classifiers through the process of combining classification techniques. In addition, the ensemble approach includes preprocessing steps. According to the study, stacking, voting, and additive trees are used less by academics than

random forests, boosting, and bagging, which are the most widely used clustering methods. In the same year, Elena N. Akimova and others [14] presented a study on a collection of literature on defect prediction. She confirmed through the study that one of the difficulties facing researchers lies in the lack of data and the lack of data sets for predicting defects. Pre-trained contextual embedding elements can be used to reduce this problem. Used to pre-train the language model on a huge set of source code, where two tables was provided for the dataset: unlabeled code datasets and labeled code dataset.

In (2022), Yu Tang and others [15] presented an algorithm to predict software defects using ensemble learning based on the adaptive variable sparrow search algorithm. They explained that the main reason that affects the model’s performance is the unbalanced data distribution, so they proposed using a new model to predict defects. For pre-processing, they used the non-stationary cut point’s algorithm in the packing model for the group samples. To improve the learning machine, the adaptive Sparrow search algorithm was used. To show the results, they used the voting method. The proposed method outperformed the other methods according to the evaluation index of the experimental results that were implemented on 15 publicly available data sets.

In (2023), Tao Shi and others [16] demonstrated through their study that the process of anomaly detection in software systems has a significant impact on the quality and reliability of software, and that current anomaly detection devices and methods do not take into account the change in data in the time dimension of software anomalies. Therefore, they proposed a new method for detecting anomalies in software systems by relying on multiple time windows that help in comprehensively understanding the anomalies. To overcome the challenges hindering the detection of cumulative anomalies, a voting mechanism using multiple time windows was used to achieve progressively better performance for detecting cumulative anomalies. . In order to improve performance metrics that include each of the following (precision, recall, area under the curve, and F1 score), they presented two hybrid methods based on precision and sensitivity. Evaluation of the method using eight datasets

divided between real data and real-time data (SKAB, Ad-cpc, Occu, speed, twitter, and cardio). They achieved high performance demonstrated by The minimal decrease in precision. In the same year, RuyueXin and others[17] emphasized something very important, which is the detection of anomalies in runtime performance, and the current research conducted on it, and that methods for detecting anomalies must meet three main requirements (accuracy, robustness, and capacity). They pointed out that current research works on different features in data, which have an impact when implemented on different types of data, and their performance is inconsistent and different. Current research is only improving the accuracy in detecting anomalies. Therefore, they proposed a framework that addresses and integrates current research challenges of detection methods to improve detection methods. This framework is based on ensemble learning. To evaluate the performance of the proposed framework, three datasets (Decentralized Application Monitoring Data (DApp) and also SMD (Server Machine Dataset)) that are publicly available and also available to everyone Vishalana are used. Experimental results showed that the proposed framework significantly improved the accuracy and robustness of performance anomaly detection, especially since the clustering method achieved a high ARP_score of 5.1821.

In (2024), Jasmeet Kaur [2] presented a summary of recent research and methodologies for identifying software bugs, predicting software defects, and developing preventive measures. There are three different types of repositories that generally make up a bug dataset: 1. Commercial and proprietary software; 2. Partially free software; and 3. Public software that anyone can access, such as a NASA data repository. He explained that Bagging classifiers and Naïve Bayes are the most widely used and popular methods for defect detection. The results of the evaluations indicated that future research could attempt to develop bug prediction models that can predict across projects. For companies that don't have a significant history of bug projects, this can be useful. Table (1) illustrates the summary of these projects and Weaknesses according to researcher’s opinions.

Table 1: Summary of the literature used in related work

References	Study name	Year	Techniques and methods	Weaknesses
Donghun Lee et al[10]	A Performance Anomaly Detection and Analysis Framework for DBMS Development	2012	Statistical Process Control.	Requiring continuous monitoring of a certain set of performance metrics.
Santosh S. Rathore et al [10]	A study on software fault prediction techniques	2017	Search for digital libraries and locate all relevant studies since	The topic of new software metrics sets and

			1993 across three categories: software metrics, error prediction techniques, and data quality issues 2	their validation has not been addressed
Ashraf Sayed and Nagy Ramadan [11]	Early Prediction of Software Defect using Ensemble Learning: A Comparative Study	2018	Three types of learners were used in the proposed technique (boosting, bagging, and rotation forest) and evaluated through seven types of PROMISE datasets.	Other algorithms of the group were not compared with the basic classifiers to enhance the results, and no pre-processing techniques were identified to choose the best ones.
Faseeha Matloob [12]	A Framework for Software Defect Prediction Using Feature Selection and Ensemble Learning Techniques	2019	Used Six search methods (BF), (GS), (GA), (PSO), (RS) and (LFS) on NASA dataset	The proposed framework has not been applied to other real-world datasets
YUE TAN. et al.[13]	LSTM-Based Anomaly Detection for Non-Linear Dynamical System	2020	LSTM and a Local Average with Adaptive Parameters (LAAP) algorithm. a wall shear stress dataset are collected by an adaptive sliding window.	-
GUOLIANG ZHAO et al.[1]	Predicting Performance Anomalies in Software Systems at Run-time	2020	LSTM neural network to predict performance anomalies. HadoopMapReduce sample applications which are JAVA open-source applications and Elatsticsearch which is a JAVA open-source	The performance of this approach has not been compared with more existing methods
FASEEHA MATLOOB et al.[7]	Software Defect Prediction Using Ensemble Learning: A Systematic Literature Review	2021	Random forest, boosting, and bagging. Data source ACM, IEEE, Springer Link, Science Direct.	The study of the diversity of classifiers that optimize ensemble and the impact of feature selection techniques on ensemble learning is not discussed.
Elena N. Akimova et al.[14]	A Survey on Software Defect Prediction Using Deep Learning	2021	Deep Belief Networks (DBN), Convolutional Neural Networks (CNN), Long Short Term Memory (LSTM), and Transformer architecture.	Researchers have not reached the optimal solution or identified the best models for predicting defects, and thus the problem of predicting anomalies remains open.
Yu Tang et al [15]	Software defect prediction ensemble learning algorithm based on adaptive variable sparrow search algorithm	2022	Ensemble learning(voting, Bagging)and adaptive variable sparrow search algorithm	The method addresses prediction accuracy, but takes a long time to research and select parameters.
Tao Shi et al [16]	Software Operation Anomalies Diagnosis Method Based on a Multiple Time Windows Mixed Model	2023	Voting mechanism and using 11 algorithms in the PYOD library of the Python language	Final results are unknown due to the influence of multiple time windows, the choice of which may be unnecessary, and

				models integrating precision and sensitivity provide conflicting views.
RuyueXin et al [17]	Robust and accurate performance anomaly detection and prediction for cloud applications: a novel ensemble learning-based framework.	2023	Four basic detection methods were applied (IForest, KNN, LOF, OCSVM).	-
Jasmeet Kaur[2]	A Study on Software Fault Detection Prediction in Software Engineering Activities.	2024	Naïve Bayes and bagging. Private/commercial, Partially public/freeware, Public(NASA and PROMISE data repositories).	The topic of new software metrics sets and their validation has not been addressed.

IV. CONCLUSION

The study examined literature related to a group of topics related to predicting performance anomalies, software error prediction activities, prediction techniques, and software metrics. Through this study, it can be concluded that there is a need for more studies and data collection from realistic software systems and more research on concurrent performance errors and word performance errors. Keywords and proposing new software metrics sets Subsequent research endeavors may aim to create bug prediction models for cross-project prediction that may be useful for organizations that lack a proper history of bug projects. The evaluations conducted on the results of this work showed that the effectiveness of forecasting performance varies. The literature on machine learning also suggests that using an ensemble of predictors can be an additional way to improve prediction performance. Ensemble techniques are used in most scientific fields to improve the performance of a prediction system. But the application of ensemble prediction techniques to detect software defects has only been partially investigated up to this point.

REFERENCES

- [1] GUOLIANG. Z., SAFWAT. H., YING. Z., DEREK .T and TOBY. C, “Predicting Performance Anomalies in Software Systems at Run-time”, ACM Transactions on Software Engineering and Methodology 30(3), <https://www.researchgate.net/publication/350099594>, DOI: 10.1145/3440757, January 2020.
- [2] Jasmeet. K., “A Study on Software Fault Detection Prediction in Software Engineering Activities”, ISSN-2349-5162, JETIR January 2024, Volume 11, Issue paradigm for sensor networks,” in Proc. ACM SIGMOBILE Int. Conf. Mobile Computer Network (MOBICOM), pp. 56–67, 2000.
- [3] Jens. E, André .v. H, Jan .W and Wilhelm. H, “Self-Adaptive Software System Monitoring for Performance Anomaly Localization”, DOI:10.1145/1998582.1998628, ICAC '11: Proceedings of the 8th ACM international conference on Autonomic computing June 2011 Pages 197–200.
- [4] Fabian. H, Mojdeh .G, Ana. P, and Alexander .K, “Machine learning-based run-time anomaly detection in software systems: An industrial evaluation”, DOI:10.1109/MALTESQUE.2018.8368453, Conference: 2018 IEEE Workshop.
- [5] Olumuyiwa. I, “Performance Anomaly Detection and Bottleneck Identification”, <http://dx.doi.org/10.1145/2791120>, ACM Computing Surveys 2015 Volume 48 Issue 1 Article No.: 4pp 1–35.
- [6] Mohammad .A and Wan .H, “Predicting Software Defects using Machine Learning Techniques”, <https://doi.org/10.30534/ijtcse/2020/352942020>, volume 9 No 4 August 2020.
- [7] FASEEHA. M, TAHER M. G, NASSER .T, SHABIB. A, MUNIR .A, MUHAMMAD .A. K, SAGHEER .A and AND .T, “Software Defect Prediction Using Ensemble Learning: A Systematic Literature Review”, Digital Object Identifier 10.1109/ACCESS.2021.3095559, VOLUME 9, 2021.
- [8] Fatih. Y, Akin .O, Emin .B and Deniz .K, “Multiple-classifiers in software quality engineering: Combining predictors to improve software fault prediction ability”, <https://doi.org/10.1016/j.jestch.2019.10.005> ,Volume 23, Issue 4, August 2020, Pages 938-950.
- [9] Donghun. L, Sang. k, Member, IEEE, and Arthur “A Performance Anomaly Detection and Analysis Framework for DBMS Development”, IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 24, NO. 8, AUGUST 2012.

- [10] Santosh S. R, Sandeep K, “A study on software fault prediction techniques”, DOI 10.1007/s10462-017-9563-5, Article Published: 30 May 2017 Volume 51, pages 255–327, (2019).
- [11] Ashraf A and Nagy R, “Early Prediction of Software Defect using Ensemble Learning: A Comparative Study”, *International Journal of Computer Applications* (0975 – 8887) Volume 179 – No.46, June 2018.
- [12] Faseeha Matloob, Shabib Aftab and Ahmed Iqbal, “A Framework for Software Defect Prediction Using Feature Selection and Ensemble Learning Techniques”, DOI: 10.5815/ijmecs.2019.12.02, Published Online December 2019 in MECS (<http://www.mecspress.org/>).
- [13] YUE. T, CHUNJING. H, KUAN. Z, ETHAN A, and JAE .S. P, “LSTM-Based Anomaly Detection for Non-Linear Dynamical System”, Digital Object Identifier 10.1109/ACCESS.2020.2999065, VOLUME 8, 2020.
- [14] Elena. N. A, Alexander Yu. B, Artem A. D, Konstantin. S. K, Anton .V. K, Ilya. P. M and Vladimir .E. M, “A Survey on Software Defect Prediction Using Deep Learning”, <https://doi.org/10.3390/math9111180>, 2021, MSC: 68N30; 68T07.
- [15] Yu Tang, • Qi Dai, Tony Du and • Lifang Chen, “Software defect prediction ensemble learning algorithm based on adaptive variable sparrow search algorithm” , *International Journal of Machine Learning and Cybernetics* <https://doi.org/10.1007/s13042-022-01740-2>.
- [16] Tao S., Zhuoliang Z and Jun A, “Software Operation Anomalies Diagnosis Method Based on a Multiple Time Windows Mixed Model”, *Appl. Sci.* 2023, 13, 11349. <https://doi.org/10.3390/app132011349>.
- [17] Ruyue X, Hongyun L, Peng C, and Zhiming Z, “Robust and accurate performance anomaly detection and prediction for cloud applications: a novel ensemble learning-based framework”, <https://doi.org/10.1186/s13677-022-00383-62023>.

Citation of this Article:

Raghda Azad Hassan, Ibrahim Ahmed Saleh, “A Survey of Software Systems Anomaly States Prediction Based on Artificial Techniques”, Published in *International Research Journal of Innovations in Engineering and Technology - IRJIET*, Volume 8, Issue 3, pp 173-180, March 2024. Article DOI <https://doi.org/10.47001/IRJIET/2024.803023>
