International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878 (Online), Volume-8 Issue-6, March 2020

A Probabilistic and Deterministic based Defect Prediction through Defect Association Learning in Software Development

Rohini B. Jadhav, Shashank D. Joshi , Umesh G. Thorat

Abstract*: *Software development is a multitasking activity by an individual or group of team. Every one activity engages diverse tasks and complication. To accomplish quality improvement, it is essential to make every activity task free of defects. But locating and correcting defects is more expensive and time-intense. In the past, many potential methods have been used to predict potential drawbacks in the program based on the theory of probability facts. Because the probability method applies a random variable and probability distributions to find a solution, the result is always in a possible range that can be true at some time or may also be wrong. Therefore, an additional calculation method coupled with the probability of making it more accurate and new in predicting the defect of the program. In this paper, we propose a Probabilistic and Deterministic based Defect Prediction (PD-DP) through Defect Association Learning (DAL). The PD-DP implements a Probability association method (PAM) and Deterministic association method (DAM) to predict the software defect accurately in software development. The experimental evaluation of the PP-DP in compare to existing prediction methods shows enhancement in prediction accuracy.

Keywords : Software Defect Prediction, Probabilistic, Deterministic, Association Learning

I. INTRODUCTION

This software program is an integral and vital part of any domain system. It is important to develop quality programs for reliable and secure systems. A good software system should improve its important functions and innovations. However, providing high quality, reliable software requires a lot of time and effort in verification, validation, and security. However, it is very difficult to measure the time and effort required for software engineering and fault prediction [1], [2].

Manuscript received on February 10, 2020.
Revised Manuscript received on February 20, 2020.
Manuscript published on March 30, 2020.
* Correspondence Author
Prof. R.B.Jadhav*, pursued Bachelor of Computer Engineering from Bharati Vidyapeeth Deemed University College of Engineering, Pune, India
Prof. Dr. S. D Joshi completed PhD in Computer Engineering from Bharati Vidyapeeth deemed to be University, Pune- 43 India.
Mr. Umesh Thorat pursued Bachelor of Mechanical Engineering from Sinhagad College of Engineering, Pune, India

Association rule-based mining [3] is a universal study of data mining techniques to identify important correlations [4] between records. The first attempt to mine groups of objects focuses on finding patterns that monitor the frequency in a common set of objects based on thresholds for supporting source data.

The construction of a fast and accurate classification for defective records is an essential task for data mining and knowledge discovery [5]. Most of the existing association rules based on classifiers have a higher classification accuracy but are often based on the object set's support and confidence rules. Therefore, some likely changes may have to be made in associative learning the recorded defect prediction to have a suitable classification for the efficient defect forecast.

It has been investigated that deterministic models [6], [7] are used very effectively in the real process to predict the unambiguous or exact match results. In real-time applications such as "Facebook", "Google Apps", and "Twitter", users can easily be assigned deterministically for the various analysis and predictions. In a deterministic model, the substantial concerns are utilized to predict a near-precise result, while in a "non-deterministic model", these reflections are applied to predict a likely result in a probability distribution.

In this task, a probabilistic based association learning method [8] for the efficient learning of defects should support the improvisation of the defect classifier accuracy. The frequent defect pattern is exploited against a "Bayesian probability" approximation method [9] to evaluate the reliability of the probability association between the defective classes. The defect classes and attributes are defined in a presentation tree to correlate the allocation gain with the predicted defect attributes.

The success of a defect-prediction model depends on whether the neglected element is not really important to investigate in the underlying phenomenon. It is also very difficult to ensure that a particular mathematical model is appropriate before some observations are tested. To infer the accuracy of the proposed models, it is essential to test the model results through actual observations under different conditions. So it is very important to understand the defect attributes of a system with a certain determinism decision to enhance in the real-time defect prediction.

In this proposal, we present a Probabilistic and Deterministic based Defect Prediction (PD-DP) through Defect Association Learning (DAL), where a determinism method will confirm the defect predicted of a probabilistic method to overcome the probable solution of the prediction. The following organized into five sections. In the section-2 the related works in relevance to the proposal was discussed, in section-3 the proposed PD-DP methodology and DAL process is presented, in section-4 it presents the experimental evaluation and the results analysis comparison, and in section-5 it presents the conclusion.

II. RELATED WORK

In the past, many software technologies [10], [11] have been developed to support "log-based defect analysis" and the integration of modern capture techniques for the processing and modeling of historical data, such as "MEADEP" [12], " Analyze NOW" [13] and "SEC" [14]. However, "log-based analysis" is not sustained by completely automated practices, so most analyst protocol processing campaigns rely on often limited system knowledge. For example, in [15] authors have identified a complex algorithm to restart the operating system from the record to select it from the sequential analysis of log information. In addition, because a bug in activating multiple messages in the log causes a large load, the entries lead to the same results that are incorporated into the same defect representation. Pre-treatment tasks are necessary to analyze fine defects [16]. While in many case studies [17], [18], [19] the defect forecast when applying for the industry, only a few estimated studies were reported to reduce test effort or software quality through early detection of software defects.
T. Mende et. [20] suggested that the effort should be measured to assess the accuracy of the prediction of defects. While traditional metrics such as "recall", "accuracy", and "ROC curves" ignore the cost of quality assurance, it is assumed that auditing or verifying a unit is almost proportional to size. C. F. Kemerer et al. [21] examined the effect of the verification rate on software quality, while the control unit considered a comprehensive set of factors that could affect the analysis. The data arrives from the "personal software process (PSP)", which is carried out through inspections, which are the activities of the various development group. In particular, the design and review rates of the PSP code are the same as the setup cycles for inspections.
The "Association Learning methods" [22] in the process of execution makes to learn the terms of the value of attributes that often occur in a record. Standard assignment rules are a certain type of assignment rules, which are designed for a group of records that are illustrated by a group of attributes, the order assignment rules determine the order of relationships between the attributes of the record that apply to an assured percentage of the records. In real-world datasets, attributes exist with other domains and associations amid them as ordinal numbers. In such cases, association rules are not strong sufficient to explain the regularity of data. As a result, rules for relational associations [23] were introduced to capture different types of relationships between log and captured data attributes.

Defect Prediction Model Illustration

The possible defect prediction models are usually categorized into two types of probabilities: "non-deterministic or probabilistic models" and "deterministic models".

A. Probabilistic Models

Probabilistic methods or models [24], [25] rely on the fact that probability theory or randomization plays an important role in predicting upcoming events. The reverse is determinism, a random reflection - suggest that to some extent if it predicted precisely, exclusive of the additional complexity of unpredictability.

It has a "random variables" and "probability distributions" are integrated into the event or phenomenon model. While the peremptory model provides one probable conclusion to an event, whereas the probabilistic model provides the probability distribution as a clarification. These models acquire the detail that they seldom recognize the whole thing concerning the situation. There is always a random element to be observed. For example, system operations depend on the functions we know for certain that perform some functions, but if a defect occurs it will fail, but we do not know when it can be or in any case. These models can be partial, partially random or completely random. Even software defects are also uncertain event has a significant impact on the system. In such a case, the probability models may be combined to determine the potential impact more useful.
The probabilistic method presented by Paul Erdös for the first time [26] presents a means proving the survival of a formation with definite characteristics. The thought is to construct a random space and to prove that the selection of a random element has the positive properties required for any random element in space. This method is extensively utilized in a diversity of fields which including "statistical physics", "quantum mechanics", and "theoretical computer science".
In a non-deterministic model, the circumstances in which the experiment is observed will determine only the probabilistic behavior of the observed result. For example, we would like to determine the amount of rainfall due to a particular storm system passing through a specific location. Tools that record precipitation is available. Meteorology might provide important information about the nearby storm system, atmospheric pressure at different points, changes in pressure, origin and direction of the storm, and so on. But this information does not make it possible to accurately determine such rainfall. This phenomenon does not lead to an inevitable approach, but rather to a probability model that describes the phenomenon more precisely. Thus, while deterministic models use physical considerations to predict almost accurate results, non-deterministic models use these considerations to predict more probability distributions.
Let's assume that the software consists of three modules: A, B, and C. One of these modules has been found to be faulty because the system is not functioning abnormally. If the system cannot correct the correct module, time and money can be lost. In this case, all modules must rest to anticipate actual defects.

If we think that a failure prediction model has knowledge of defects similar to those of the past, then the likelihood of a prediction may be true or false, and to overcome such a combination of probabilistic methodologies, It is necessary to make predictions. The probability depends on the knowledge of the system rather than the "real" possibility.
Probabilistic models typically identify defect gaps based on algorithms and assume project-to-project flaws. Most results show up to 90% of the correct prediction rate. However, the key to achieving accurate probability matching is to correlate defects or defects using defect profiles that contain highly specific and relevant information.

Project module versions are accessed using different types of hardware interface devices, but all have the same IP address. In such cases, it is difficult to identify and match the defects in a probable manner. Therefore, additional information knowledge can be an additional consideration in all project modules to predict the exact defect.

B. Deterministic Models

In the deterministic model [6], [7], it is assumed that the actual result (value or other value) is determined by the conditions under which the experiment is performed. For example, running a program in a simple module becomes a predictive model that describes the observable flow of the program. The model predicts the value of the variable as soon as the input variable is provided. If you perform the above experiment multiple times, you can expect similar output values using variable inputs. It can be very small if the deviations that can occur are not likely to occur in the system. It is important to note that probability and determinism are not mutually exclusive. The system can be quite specific, but only the probabilities within peremptory systems can be assessed. In other words, the probabilistic theory does not imply the inevitability of the use of defects / non-specificity in the development of software systems. The probability is not "currently" other than what is actually in the system by default. The defect is actually 100% in the module program, but the probability based on the hypothesis can overlook the existence of a defect.
F. Chang et al. [27] present a static analysis of the three main aspects of an advanced industrial software system analyzed by Nortel Networks determines the extent to w
hich it able to support the high-quality artifacts in economic production, investigate defects and lose customer reports through static analysis. Data show that automated static analysis is the appropriate way to detect software defects. The identification and investigation of defects through automated static analysis by means of "Orthogonal defect classification" systems which allows subsequent program creation steps to focus on more complex, functional and logarithmic defects.
T. Khoshgovar et al. [28] proposed a model for testing software quality features based on a list of future and defect density units. Typical defective inputs raise software complexity metrics in terms of LOC, various unique operators, and process complexity. Perform the following gradual regression to find weights for each factor. It uses the object-oriented scales to predict categories that may include defects and predict the defect-prone groups using PCA with logistic regression. S. Morasca et al. [29] predict defective units are expected to use coarse group theory and logistic regression in business programs.
P. L. Li et al. [30] in the "ABB Inc." presented the experience of predicting application defects. The experiment is a practical question concerning how to choose the correct modeling method and how to estimate the correctness of the forecasts for the different versions of the development.

III. PROBABILISTIC AND DETERMINISTIC BASED DEFECT PREDICTION

Defect prediction mostly performed based on past defects observation in different software development. As the development of software does not follow a common guideline or measures for specific domains, due to which the variation of defect kinds generates. So, it is essential to learn these defects kinds precisely to minimize these defects in development. We propose a Probabilistic and Deterministic based Defect Prediction (PD-DP) through the Defect Association Learning (DAL) based on attributes association measures as shown in Fig. 1. The process of DAL generates a knowledge of defect associated attributes which can be utilized for defect prediction. The integration of Probability and Deterministic association method makes a probable prediction to a definite prediction. In the following sections, we describe the functionality of the methods in details.

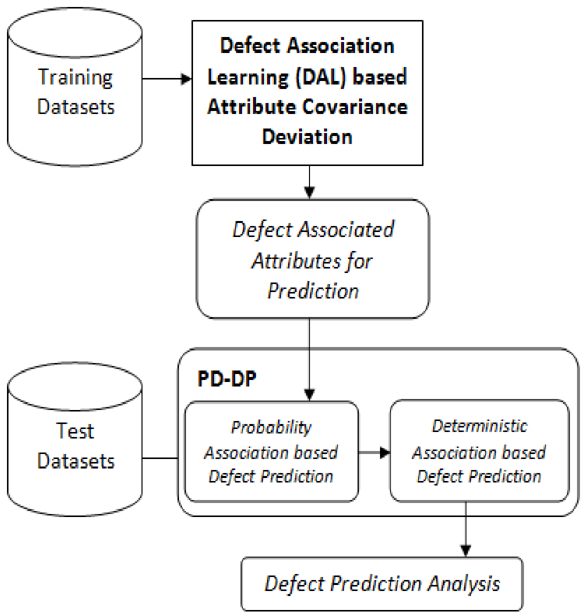

Fig.1: System Architecture for PD-DP mechanism

A. Defect Association Learning (DAL)

The information knowledge discovery is broadly used multiple data mining approaches in integration for data analysis, such as Associative Classification approach [31]. An associative Classification is an effective approach for building classifier [32]. It is a novel and strong approach based on association rule mining. It mostly utilized small and qualified association rules for prediction, decision, and classification.

To understand the base association among the defects it necessary to build associated rules using multiple attributes patterns for accurate prediction through a Defect Association Learning (DAL). The DAL method combines the highly defect associated patterns obtained from multiple domain defects sets of different datasets to form a combined rules patterns which will be highly efficient for defect prediction and useful for decision making. It implements two mechanisms to build effective prediction rules. It initially finds the highly associated attributes through attribute reduction mechanism and the reduced attributes patterns are generated, which are combined to form the efficient defect prediction rules.

The attribute reduction is an effective mechanism to find highly interesting attributes required for the prediction. Some attributes might have abundant redundancy of data due to the inconsistency of resource generated. The process of DAL measures each attributes association using a "Covariance Deviation (*CD*)" measure [33] to find the highly impacting defect attributes. In such case, for a given two attributes we compute the strong association implies using a probability and statically CD between two or more attributes.
Let's assume, X and Y are the two attributes having a unique set of k values as $\{(x_1,y_1), . . . , (x_k ,y_k)\}$, and the entropy, E of these values of X as $E(X)$ and Y as $E(Y)$ is computed using the Eq. (1) and Eq. (2) as,

$$E(X) = \bar{X} = \frac{\sum_{v=1}^{k} x_v}{k} \tag{1}$$

$$E(Y) = \bar{Y} = \frac{\sum_{v=1}^{k} y_v}{k} \tag{2}$$

Based on the computed E of the attribute we compute each attributes CD variance using Eq. (3) as,

$$CD_A = \sum_{v=i+1}^{k} H(X) - H(Y_v) \tag{3}$$

Utilizing the attributes *CD* values it creates sets of attributes as, F which are >=1. The attributes which CD value is < 1 are considered as low variance and less impact on prediction and the attributes which are >=1 are considered as high variance and have impacts on prediction. Now, using the generated F from the reduced and associated attributes, it will build the required defect associated attributes for prediction of each defect. This will minimize the computational overhead and provide efficient prediction rules.

B. Probabilistic and Deterministic based Defect Prediction

The existing prediction classifiers [34] are mainly based on the association rule mining patterns. These rules are mostly learned from a set of data instances which are trained and labeled, and this learning is applied for the new data instance for the classification. In similar to this, the proposed Probabilistic and Deterministic based Defect Prediction (PD-DP) perform the prediction utilizing the trained knowledge data obtained from DAL in the form of the reduced attributes set as F, and its unique item sets to generate each attribute pattern to perform PD-DP. The process of PD-DP implements two methods to enhance the potential of defect prediction as, *(1) a Probabilistic association method (PAM)*, and *(2) a Deterministic association method (DAM)* as discussed below.

Probabilistic Association Method
Let's assume a set a defect datasets as, D_n having an F reduced attributes sets which build a P pattern for the defect prediction using attributes values. To generate the individual patterns for each attributes it utilize a DAL method using Eq. (4) as,

$$P_n = R(F_k) \tag{4}$$

where "P_n" → is the extracted pattern of each attribute and n=*1, ... , N*, "*R*" → is the data processing method used for itemset extraction and "F_k" → is the defect attributes value from $k=1, ... , K$.

The obtain patterns as P_n of each attribute as F_k will merge to generate a combined pattern as P_k for each dataset of D_n.

$$P_k = C_F(P_n) \tag{5}$$

Using, the equation (4) and (5), a new defect prediction rule will be formed for all dataset, D_n in the combine as,

$$P_n = R\,(F_1 \wedge \ldots \ldots \wedge F_k) \longrightarrow A_k \tag{6}$$

$$P := C\,(P_1 \wedge \ldots \ldots \wedge P_n) \longrightarrow Q \tag{7}$$

where," A_k " → is referred to as associated patterns, and " Q " → is referred to as qualified patterns for the defect prediction rules. If the attributes are associated with the test data itemsets then the data record is considered a defect, i.e., considered as "True Positive", and it is further qualified for the decision making through a deterministic method.

Deterministic Association Method
A deterministic association method (DAM) perform the defective data prediction utilizing the PAM generated associated defected records. Let's assume a set of test data predicted as defected through PAM as Z_k , which having F_x attributes. As the initial attributes pattern extraction using Eq. (4) generates P_n patterns which will be combined using Eq. (5) to get the combined pattern of the input data records is presented in Eq. (8) and Eq. (9).

$$P_n = R\,(F_1, \ldots \ldots, F_x) \tag{8}$$

$$P_k = C_F(P_n) \tag{9}$$

Now, to have a definite prediction using deterministic association method we compute the correlation of the patterns of each predicted records as E obtain through the PAM, by measuring the "Support", "Confidence", and "Lift" using Eq. (10), (11), and (12).

Retrieval Number: F8778038620/2020©BEIESP
DOI: 10.35940/ijrte.F8778.038620
Journal Website: www.ijrte.org

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

$$Support = Prob(E \wedge Q) \quad (10)$$

$$Confidence = Prob(E \wedge Q)/(Prob(E)) \quad (11)$$

$$Lift = Prob(E \wedge Q)/(Prob(E) * Prob(Q)) \quad (12)$$

The measure of Lift using Eq. (12) present the deterministic of the prediction. If the lift value is less than one with the DAL generated *Q* rules patterns, then the prediction is considered is negative, and if the value is greater or equal to one then it is positively correlated with the prediction rules and qualifies as accurate defect prediction.

In the following section evaluate the proposed methods over a "NASA PROMISE repository" [35].

IV. EXPERIMENTAL EVALUATION

A. Datasets

The dataset is collected from the "NASA PROMISE repository" [35] of the dataset as "PC1, PC2, CM1, KC3, and JM1" are used to evaluate the proposed operation. All data sets have different attributes, modules, and defect ratios. Descriptions for each data set are shown in Table 1.

Table-1: Dataset description

Modules	Number of Instances	% of defects
PC1	759	8.1 %
PC2	1585	1.0 %
CM1	327	12.8 %
KC3	200	18.0 %
JM1	9371	18.5 %

The data records are presented with the class value (defects) as "false" or "true". The "false" states that the module may have or not one or more defect, whereas "true" states that it is reported defects. These datasets are evaluated in comparison with few existing probabilistic based classifiers as, "BayesNet", "NaiveBayes", "JRip", and "OneR". To perform the evaluation analysis we implement the DAL method using java and performance evaluation is measured using Weka-3.6 Tool.

B. Result Evaluation

Based on the attributes patterns selected through DAL we implement the PD-DP mechanism to evaluate the defect prediction accuracy in comparison to the existing classifiers. To outcome measure of the classifier performance are presented in Table-2 below.

The comparison of defect prediction accuracy among the classifiers is shown in Fig.2. It shows that PD-DP approach achieves an average of 6% higher accuracy with a lower defect rate. The improvisation is due to the accurate attribute selection through DAL and accurately predicting the defected records through the discriminated method with a 2% support, confidence and lift measures.

The comparison of relative absolute defect comparison is shown in Fig.3. The proposed PD-DP shows a lower defect rate in comparison. The BayesNet, JRip, and OneR are averagely showing similar accuracy and error rate, whereas NaiveBayes showing better prediction accuracy but an average of 3% low in compare to PD-DP. As NaiveBayes and PD-DP both apply probability approach initially but the integration deterministic method enhances the prediction accuracy in comparison.

Table-2: Defect Prediction Accuracy and Relative Abs. Error Comparison

Classifiers	Modules	Correctly Classified	Incorrectly Classified	Prediction Accuracy	Relative Absolute Error
BayesNet	PC1	701	58	92.358	7.642
	PC2	1351	234	85.237	14.763
	CM1	298	29	91.131	8.869
	KC3	185	15	92.500	7.500
	JM1	7681	1690	81.966	18.034
NaiveBayes	PC1	721	38	94.993	5.007
	PC2	1453	132	91.672	8.328
	CM1	304	23	92.966	7.034
	KC3	186	14	93.000	7.000
	JM1	8457	914	90.247	9.753
JRip	PC1	685	74	90.250	9.750
	PC2	1247	338	78.675	21.325
	CM1	285	42	87.156	12.844
	KC3	181	19	90.500	9.500
	JM1	7541	1830	80.472	19.528
OneR	PC1	698	61	91.963	8.037
	PC2	1326	259	83.659	16.341
	CM1	288	39	88.073	11.927
	KC3	176	24	88.000	12.000
	JM1	7210	2161	76.939	23.061
PD-DP	PC1	748	11	98.551	1.449
	PC2	1504	81	94.890	5.110
	CM1	315	12	96.330	3.670
	KC3	198	2	99.000	1.000
	JM1	9018	353	96.233	3.767

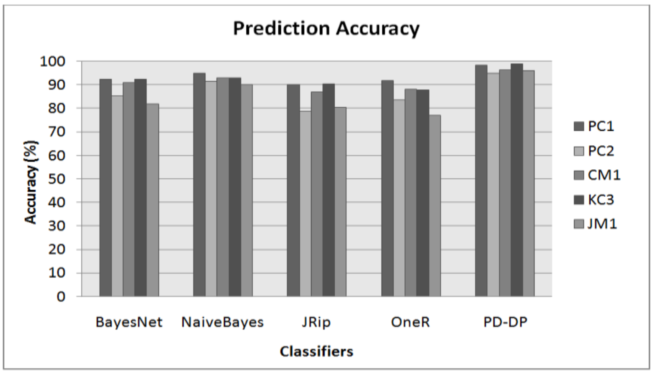

Fig.2: Defect Prediction Accuracy

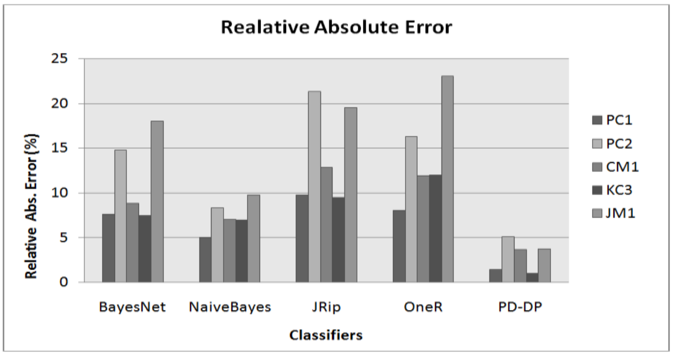

Fig.3 Relative Abs. Error Rate Comparison

V. CONCLUSION

This paper proposes a Probabilistic and Deterministic based Defect Prediction (PD-DP) through Defect Association Learning (DAL). The DAL process associates the defect attributes through measuring the Covariance Deviation between the attributes to find the highly impacting attributes for the defect prediction. The learn knowledge pattern of each defect is utilized for defect prediction. The PD-DP implements two methods to do accurate defect prediction. First, a Probability association method is implemented to associate the test data probability to a defect pattern through DAL and classify it as defective or non-defective according to the attribute association to the DAL patterns. Later, it implements a Deterministic association method to predict the software defect accurately in software development. It computes the deterministic of the probable defect predicted data through support, confidence and lift measures. The experimental evaluation shows an improvisation inaccuracy and low error rate in comparison with existing classifier approaches. The improvisation in the defect prediction will be effective in software development in the enterprise-level organization projects for critical defect analysis and decision making.

REFERENCES

1. H. Liang, Y. Yu, L. Jiang, Z. Xie, "Seml: A Semantic LSTM Model for Software Defect Prediction", IEEE Access, Volume: 7, 2019.
2. Y. Kamei, E. Shihab, B. Adams, A. E. Hassan, A. Mockus, A. Sinha, and N. Ubayashi, "A Large-Scale Empirical Study of Just-in-Time Quality Assurance", IEEE Transactions On Software Engineering, Vol. 39, No. 6, June 2013.
3. X. Xue, C. Yao, W. Yan-en, "Study on Mining Theories of Association Rules and Its Application" International Conference on Innovative Computing and communication, 978-0-7695-3942-3/10, 2010.
4. Q. Song, M. Shepperd, M. Cartwright, C. Mair, "Software Defect Association Mining and Defect Correction Effort Prediction", IEEE Transactions on software engineering, Vol. 32, no. 2, 2016.
5. J. -h. REN, F. LIU, "Predicting Software Defects Using Self-Organizing Data Mining", IEEE Access, 2019.
6. W. Charlotte Werndl, "On the observational equivalence of continuous-time deterministic and indeterministic descriptions", European journal for philosophy of science, Vol. 1(2), pp. 193-225, 2011.
7. C. Woodruff, L. Vu, K. A. Morgansen, D. Tomlin, "Deterministic Modeling and Evaluation of Decision-Making Dynamics in Sequential Two-Alternative Forced Choice Tasks", IEEE Proceedings, Vol.100 (3), 2012.
8. J. Ge, J. Liu, W. Liu, "Comparative Study on Defect Prediction Algorithms of Supervised Learning Software Based on Imbalanced Classification Data Sets", 19th IEEE/ACIS International Conf. on SNPD, pp. 399 - 406, 2018.
9. O. O. T. Yildiz, "Software defect prediction using Bayesian networks", Empirical Software Engineering, pp. 1-28, 2012.
10. J. Chen, Y. Yang, K. Hu, Q. Xuan, Y. Liu, C. Yang, "Multiview Transfer Learning for Software Defect Prediction", IEEE Access, Vol. 7, 2019.
11. C. Tantithamthavorn, S. McIntosh, A. E. Hassan, K. Matsumoto, "An Empirical Comparison of Model Validation Techniques for Defect Prediction Models", IEEE Transactions on Software Engineering, Volume 43, No. 1, pp. 1 - 18, 2017.
12. D. Tang, M. Hecht, J. Miller, and J. Handal, "Meadep: A Dependability Evaluation Tool for Engineers", IEEE Transaction Reliability, Vol. 47(4), pp. 443-450, 1998.
13. A. Thakur and R.K. Iyer, "Analyze-Now—An Environment for Collection and Analysis of Failures in a Networked of Workstations", IEEE Transaction Reliability, Vol. 45(4), pp. 561-570, 1996.
14. R. Vaarandi, "SEC—A Lightweight Event Correlation Tool", Proc. Workshop IP Operations and Management, 2002.
15. C. Simache and M. Kaaniche, "Availability Assessment of SunOS/ Solaris Unix Systems Based on Syslogd and Wtmpx Log Files: A Case Study", EE Pacific Rim International Symposium on Dependable Computing (PRDC'2005), pp. 49-56, 2005.
16. D. -L. Miholca, "An Improved Approach to Software Defect Prediction using a Hybrid Machine Learning Model", IEEE 20th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC), 2018.
17. E. A. Felix, S. P. Lee, "Integrated Approach to Software Defect Prediction", IEEE Access, Volume 5, pp. 21524 - 21547, 2017.
18. X. Yu, J. Liu, Z. Yang, X. Jia, Q. Ling, S. Ye, "Learning from Imbalanced Data for Predicting the Number of Software Defects", IEEE 28th International Symposium on Software Reliability Engineering (ISSRE), pp. 78 - 89, 2017.
19. M. A. Kabir, J. W. Keung, K. E. Benniny, M. Zhang, "Assessing the Significant Impact of Concept Drift in Software Defect Prediction", IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC), Vol. 1, 2019.
20. T. Mende and R. Koschke, "Effort-Aware Defect Prediction Models", Proc. European Conf. Software Maintenance and Reeng., pp. 109-118, 2010.
21. C. F. Kemerer and Mark C. Paulk, "The Impact of Design and Code Reviews on Software Quality: An Empirical Study Based on PSP Data", IEEE Transactions on Software Engineering, Vol. 35(4), 2009.
22. K. Yang, H. Yu, G. Fan, X. Yang, S. Zheng, C. Leng, "Software Defect Prediction Based on Fourier Learning", IEEE International Conference on Progress in Informatics and Computing (PIC), 2018.
23. S. Herbold, A. Trautsch, J. Grabowski, "Correction of - A Comparative Study to Benchmark Cross-Project Defect Prediction Approaches", IEEE Transactions on Software Engineering, Vol. 45 (6), 2019.
24. Q. Yu, J. Qian, S. Jiang, Z. Wu, G. Zhang, "An Empirical Study on the Effectiveness of Feature Selection for Cross-Project Defect Prediction", IEEE Access, Vol. 7, 2019.
25. F. Zhang, A. Mockus, I. Keivanloo and Y. Zou, "Towards building a universal defect prediction model", In Proceedings of the 11 th ACM Working Conference on Mining Software Repositories, p. 182-91, 2014.
26. F. R. K. Chung, "Open Problems of Paul Erdos in Graph Theory", Journal of Graph Theory, Vol. 25(1), pp. 3-36, 1997.
27. F. Zhang, Q. Zheng, Y. Zou, A. E. Hassan, "Cross-Project Defect Prediction Using a connectivity-based Unsupervised Classifier", IEEE/ACM 38th International Conference on Software Engineering (ICSE), pp. 309 - 320, 2016.
28. T. Khoshgoftaar and E. Allen, "Predicting the Order of FaultProne Modules in Legacy Software", Proc. Int'l Symp. Software Reliability Eng., pp. 344-353, 1998.
29. S. Morasca and G. Ruhe, "A Hybrid Approach to Analyze Empirical Software Engineering Data and Its Application to Predict Module Fault-Proneness in Maintenance", J. Systems Software, Vol. 53(3), pp. 225-237, 2000.
30. P. L. Li, J. Herbsleb, M. Shaw, and B. Robinson, "Experiences and Results from Initiating Field Defect Prediction and Product Test Prioritization Efforts at ABB Inc.", Proc. Int'l Conf. Software Eng., pp. 413-422, 2006.

31. K. Hewawasam, K. Premaratne, and M.L. Shyu, "Rule mining and classification in a situation assessment application: A belief-theoretic approach for handling data imperfections," IEEE Transactions On Syst., vol. 37, no. 6, pp. 1446-1459, Dec. 2007.
32. Y. Sun, Y.Wang, and A.Wong, "Boosting an associative classifier," IEEE Transactions On Knowledge and Data Engineering, vol. 18, no. 7, pp. 988-992, Jul. 2006.
33. C. Tantithamthavorn, S. McIntosh, A. E. Hassan, K. Matsumoto, "The Impact of Automated Parameter Optimization on Defect Prediction Models", IEEE Transactions on Software Engineering, Vol. 45 (7), 2019.
34. E. Baralis, L. Cagliero, and P. Garza, "EnBay: A Novel Pattern-Based Bayesian Classifier", IEEE Transactions On Knowledge And Data Engineering, Vol. 25(12), 2013.
35. PROMISE Software Engineering Repository, Available at: http://promise.site.uottawa.ca/ SERepository, accessed June 2014.

AUTHORS PROFILE

Prof. R.B.Jadhav pursued Bachelor of Computer Engineering from Bharati Vidyapeeth Deemed University College of Engineering, Pune, India in year 2012 and Master of Computer Engineering from Bharati Vidyapeeth deemed to be University, Pune-43 India in year 2014. She is currently working as Assistant Professor in Department of Cpmputer Engineering, Bharati Vidyapeeth Deemed to be University, Pune-43 India since 2016. Se has published research papers in reputed international journals including Scopus Journal and conferences which are available online. Her main research work focuses on Software Engineering, Machine Learning . She has 6 years of teaching experience.

Prof. Dr. S. D Joshi completed PhD in Computer Engineering from Bharati Vidyapeeth deemed to be University, Pune- 43 India. He is currently working as Professor from last 27years in Department of Computer Engineering, Bharati Vidyapeeth Deemed to be University, Pune-43 India. He has published 215 research papers in reputed international journals including Scopus Journal and international conferences which are available online. His main research work focuses on Software Engineering.

Mr. Umesh Thorat pursued Bachelor of Mechanical Engineering from Sinhagad College of Engineering, Pune, India in year 2008 and Master of Business Administration from Balaji Institute of Modern Management, Pune India in year 2014. He is currently working as Solution Architect in Tata Consultancy Services, Pune.He has published research papers in reputed international journals including Scopus Journal and conferences which are available online. His main research work focuses on Software Engineering, Machine Learning and artificial Intelligence He has 8+ years of experience.