

## An efficient logistic regression and ant colony optimization-based object-oriented quality prediction

Jitendra Kumar Saha<sup>1\*</sup>, Kailash Patidar<sup>2</sup>, Rishi Kushwah<sup>2</sup> and Gaurav Saxena<sup>2</sup>

M.Tech Scholar, Department of Computer Science, School of Engineering, Sri Satya Sai University of Technology & Medical Sciences, Sehore, Madhya Pradesh, India<sup>1</sup>

Assistant Professor, Department of Computer Science, School of Engineering, Sri Satya Sai University of Technology & Medical Sciences, Sehore, Madhya Pradesh, India<sup>2</sup>

©2020 Jitendra Kumar Saha et al. This is an open access article distributed under the Creative Commons Attribution (CC BY) License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

### Abstract

*In this paper an efficient logistic regression and ant colony optimization-based object-oriented quality prediction has been presented. The dataset has been considered based on the object-oriented codes. The code considered here are completely equipped with object-oriented features for the complete analysis. The major parameters considered for the experimentation are class, object, inheritance and dynamic behavior. Class and objects have been considered for the class labels and memory requirements checking with the proper stack call and constructor invocation. Inheritance has been considered for the code reusability testing. Dynamic behavior has been tested for the runtime allocation. Then clustering algorithm is used for the parameter preprocessing and grouping based on the parameters. For data filtration chi-square testing has been applied. Then logistic regression and ant colony optimization (LR-ACO) have been considered for the final classification. Then F-Measure, Odd Ratio and Power have been considered for the analysis of the classification based on LR-ACO. The result after LR-ACO shows better accuracy as comparison to the previous methods.*

### Keywords

*Class, Object, Inheritance, LR-ACO, F-Measure, Odd Ratio, Power.*

### 1. Introduction

In the current scenario the use of different computer-based applications is increasing. It has been increased drastically in the digital world [1]. So, the reliability and accurateness of the software and applications are very important as it affects the complete cycle [2, 3]. It also affects the development of configuration [4]. The major aim of analysis and prediction of the software faults are the project progress, inappropriate development tracking and the evaluation prediction [5].

The use of object-oriented metrics for the detection of fault and errors are important as based on these factors' prominence can be checked [6, 7]. It is also helpful in reusability checking and dynamic behavior assessment [8–11]. It is also important as most of the applications support object-oriented paradigm and the applicability of those paradigm is extremely important prerequisite for these types of framework.

The main objective of this paper is to analysis the hybrid classification of the object-oriented faults based on logistic regression and ant colony optimization (LR-ACO).

### 2. Literature survey

In 2015, Malhotra and Khanna [12] discussed about the defect correction and its evolution. They have suggested the use of different statistical techniques like machine learning and search-based techniques. They have assessed the performance of different machine learning algorithms and search-based techniques. Their results support inter-project validation.

In 2015, Kumar and Rath [13] discussed about fault prediction system. They have proposed a cost evaluation model for the analysis. This analysis has been presented for the fault's misclassification. They have considered Chidamber and Kemerer (CK) metrics. They have considered logistic regression and the combination of neural network and particle swarm optimization (PSO) combination for the

\*Author for correspondence

design. Their results show improved results in case of cost reduction.

In 2015, Harmanani et al. [14] presented different metrics. It is for the estimation for the software quality and measurable attributes. They have presented ant colony optimization (ACO). It has been used for the accuracy prediction. They have compared and analyzed the accuracy results. It is based on the intensify search.

In 2015, Suresh [15] discussed about the fault prone identification. They have evaluated meta-heuristic search techniques. It has been evaluated based on Apache integration framework. For the prediction model they have considered genetic algorithm and PSO. Their results show the improvement in terms of accurate fault classification.

In 2016, Suresh [16] discussed about the software product quality estimation. It has been discussed based on detection and prediction of faults. They have considered different classifiers. They have also considered Chid amber and Kemerer Metric suite. Their results show the effectiveness of logistic regression.

In 2016, Jain and Chug [17] discussed about the object-oriented measuring parameters like cohesion, coupling, polymorphism and inheritance. The main objective of the software quality assurance is defect prediction, maintainability prediction, cost estimation and debugging. They have suggested regarding the dynamic measurement of the software. Their results shows the effectiveness of dynamic measure.

In 2016, Malhotra et al. [18] discussed about the software development. It has been discussed and developed based on the software system lead and their development. They have discussed the importance of advanced prediction of the software faults. They have considered and analyzed 17 machine learning algorithms. It has been considered for determining the prone classes. It has been discussed in terms of object-oriented software. It has been evaluated statistically. Their results show the predictive capability of machine learning algorithms.

In 2017, Nanda et al. [19] discussed about the software quality evaluation and process development. They have considered different object-oriented metrics. It has been considered based on feature amenability and changes.

In 2017, Singh and Malhotra [20] discussed about the software defect prediction. They have evaluated four different machine learning algorithms. Their results show that the multilayer perceptron algorithm found to be better.

In 2017, Boucher and Badri [21] discussed about the fault-proneness prediction. They have suggested that the literature supports the use of supervised learning algorithms for the same. They have adopted HySOM approach. It has been adopted for the code metrics calculation for the prediction of the fault-prone functions. It has been considered for the source code metrics threshold values. They have experimented on 12 datasets available publicly on the data repository. They have compared different supervised algorithms like naive Bayes, artificial neural network and random forest algorithms.

In 2017, Kartha et al. [22] discussed about the software reliability. They have discussed about the software error removal process. It has been discussed based on the development lifecycle. These cause the propagation of different errors and different dependent modules. Their study addresses the software defect aspect.

In 2018, Tripathi et al. [23] discussed about the service-oriented architecture. It has been discussed for the software quality parameter analysis. They have considered quality of service parameters for the analysis of different programming paradigm for the consideration of quality metrics. They have considered java files. They have considered 16 different aggregate measures. They have suggested that feature selection technique is not fit for prediction accuracy improvement.

In 2018, McCormac et al. [24] discussed about the truncated signed distance function reconstructions. It has been constructed based on the 3D foreground mask. They have considered RGB-D sequences for the validation of the object sequences.

In 2018, Aktaş and Buzluca [25] discussed about the complexity of advanced software systems. The main objective of their work is to eliminate design defects and testing codes reduction. They have suggested that the training of the models may affect the detection system accuracy. So, they have considered some of the important metrics. For this they have considered error rates. They have used correlation-based feature selection and principal component analysis for the extraction of the subset. Then they have applied

random forest classification for the determination of error-prone classes.

In 2019, Alakus et al. [26] discussed about the quality and measurements in case of software reliability. They have considered faults, bugs and errors. They have analyzed different metrics. The software survival is based on the errors based on the recommendation and different quality metrics.

In 2019, Merzah [27] discussed about the development process. They have suggested the development of high-quality systems and the code quality. They have suggested that the object-oriented metrics can be helpful in the case of code quality measurement. They have proposed to use the code test levels for the investigation and object-oriented metrics calculation. They have considered the design smells also. They have combined the object-oriented metrics with mining algorithms for the calculation of envy features.

In 2019, KS [28] discussed about the object-oriented software development. They have discussed about the structure of the code, reliability of the software along with the code determination. The object-oriented quality measurements are inheritance, effects of coupling, code complexity and size aspects. Their main aim is to correlate the relationship between the object-oriented software and its design.

In 2019, Samir et al. [29] discussed about the software defects. They have considered different defect predictions models for the same purpose. They have suggested that early prediction model relies on statistical approaches. They have considered deep neural network. It has been considered for software defect prediction.

### 3.Methods

In this paper an efficient logistic regression and ant colony optimization-based object-oriented quality prediction has been presented. The complete process has been divided into following phases:

1. **Data analysis:** Java modules have been considered for the experimentation. The dataset has been considered based on the object-oriented codes. The code considered here are completely equipped with object-oriented features for the complete analysis. The major parameters considered for the experimentation are class, object, inheritance and dynamic behavior. Class and objects have been considered for the class labels and memory requirements checking with the

proper stack call and constructor invocation. Inheritance has been considered for the code reusability testing. Dynamic behavior has been tested for the runtime allocation.

2. **Data Clustering and testing based on threshold:** Then clustering algorithm is used for the parameter preprocessing and grouping based on the parameters. For data filtration chi-square testing has been applied. It has been applied for parametric consideration and applicability based on the threshold parameter. In our case it is 0.5.
3. **Data Classification:** Then logistic regression and ant colony optimization (LR-ACO) have been considered for the final classification.

#### Logistic Regression:

The steps considered in the logistic regression are as follows:

- Linear weights have been considered as the input values. Let input values as x. Based on it the output(y) has been considered. The range considered between 0 to 1.

$$y = e^{\frac{(b_0+b_1 \times x)}{1+e^{(b_0+b_1 \times x)}}}$$

Where x is the input, y is the output and (b0, b1) are the coefficient values.

- Then the probability of modelling the same input can be represented as follows:

$$P(x) = P(y=1|x)$$

- It can be transformed in terms of function for the representation of the linear combination following the below equation:

$$P(x) = e^{\frac{(b_0+b_1 \times x)}{1+e^{(b_0+b_1 \times x)}}}$$

- Then the complete combination can be found on the linear plane.

- The ratios can be calculated as follows:

$$\ln(\text{odds}) = b_0 + b_1 \times x$$

- Final exponents retrieved for the input are as follows:

$$F(\text{odds}) = e^{(b_0+b_1 \times x)}$$

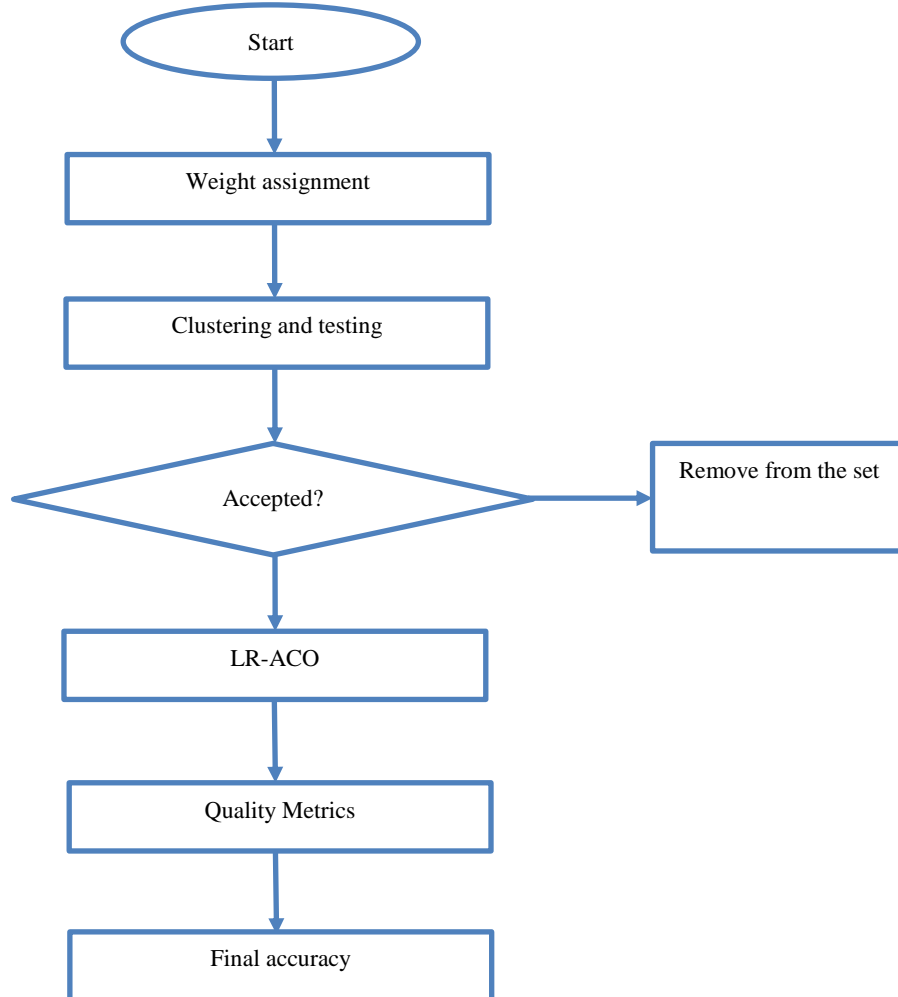
#### ACO Algorithm:

The steps involved in the ACO algorithm is shown below:

- Initialize the population.
- Initialize the pheromone values based on the faults.
- Repeat the second step for each ant and move for the next weight based on the parameters.
- Solution constructed based on the pheromone trails.
- Update the complete trails until the stopping iterations.

4. Then F-Measure, Odd Ratio and Power have been considered for the analysis of the classification based on LR-ACO. The result after LR-ACO shows better accuracy as comparison to the previous methods.

The complete process can be better understood from the flowchart as shown in *Figure 1*.



**Figure 1** Flowchart of the LR-ACO based fault detection

#### 4.Results

The results considered here are based on 5 different sets. The sets have been considered from Java based code modules. Total ten iterations have been

considered. These are from I1 to I10. Set denotes the modules. The results (*Figure 2 to Figure 6*) clearly depict that the average and individual accuracy of the fault detection are found to be satisfactory.

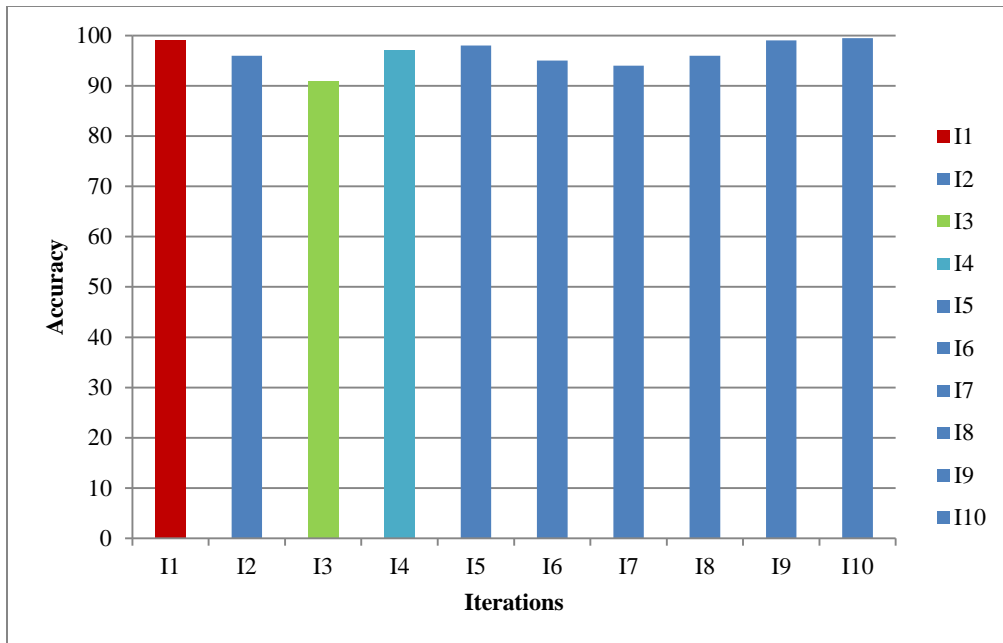


Figure 2 Fault detection accuracy in case of LR-ACO (Set 1)

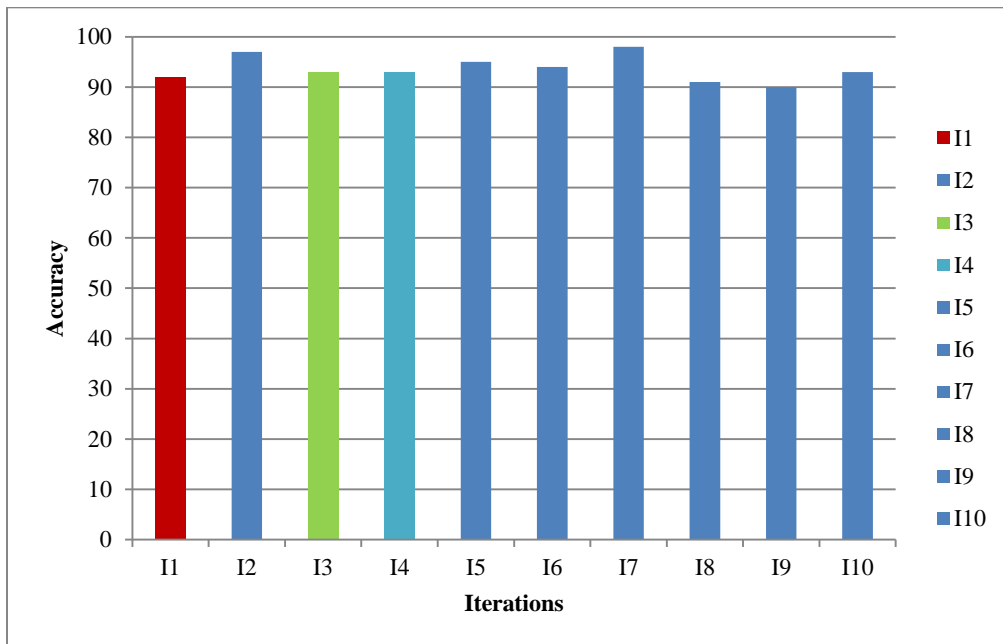


Figure 3 Fault detection accuracy in case of LR-ACO (Set 2)

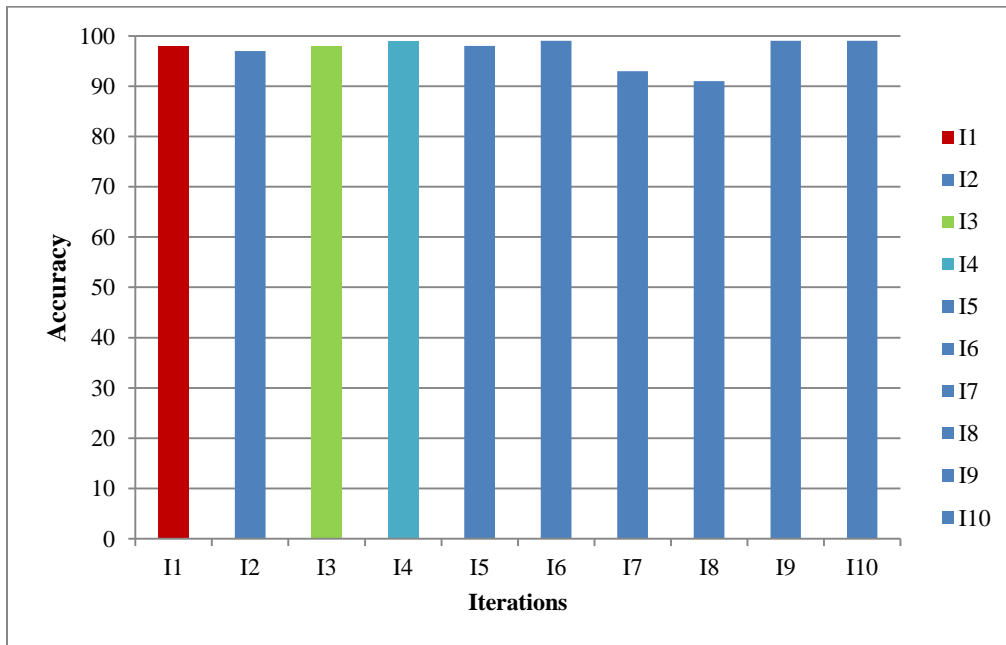


Figure 4 Fault detection accuracy in case of LR-ACO (Set 3)

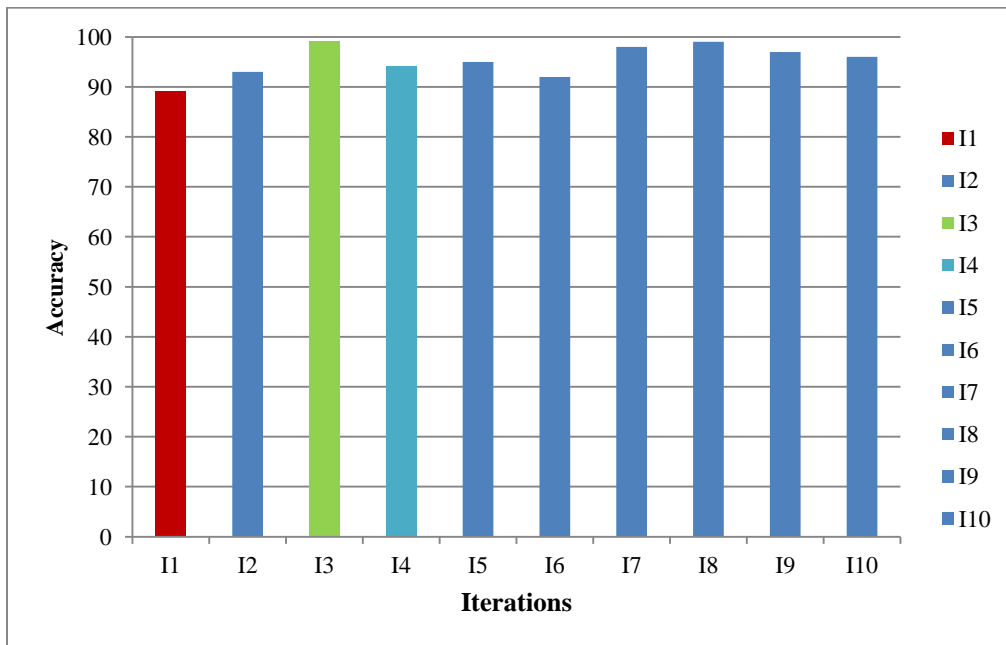
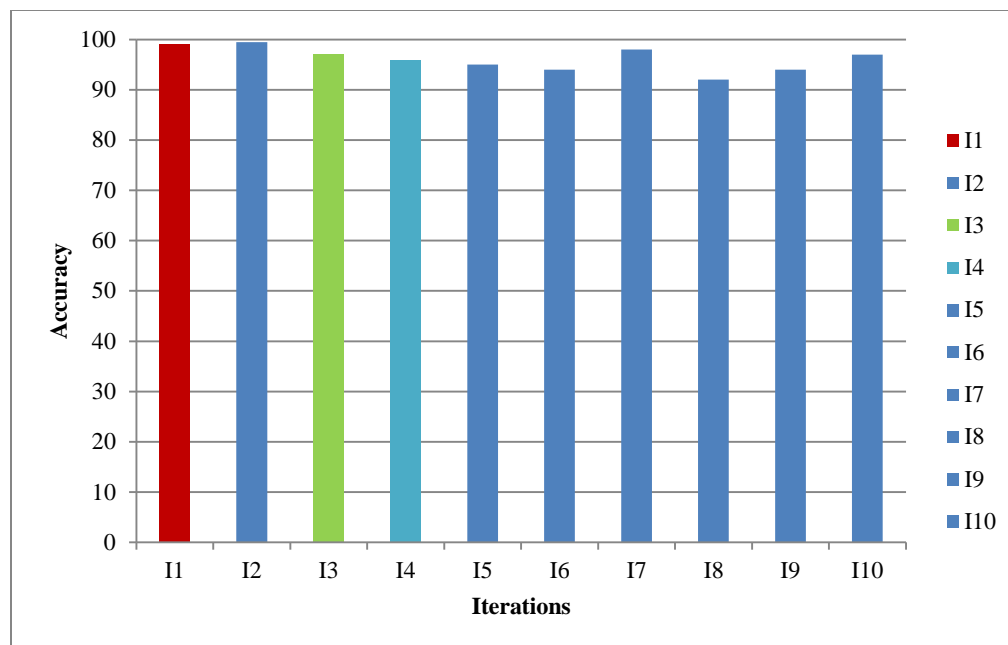


Figure 5 Fault detection accuracy in case of LR-ACO (Set 4)



**Figure 6** Fault detection accuracy in case of LR-ACO (Set 5)

#### 4. Conclusion

In this paper an efficient fault detection mechanism has been presented and analyzed. LR-ACO method has been used for this purpose. The complete process is divided into four phases. In the first phase data preprocessing has been performed. Second phase is of data arrangement based on clustering and data acceptance based on chi-square threshold. Then LR-ACO has been used for the further data classification. Finally, software performance metrics has been used for the complete analysis. The results support the approach as it is found to be prominent in the fault detection with less error rate.

#### Acknowledgment

None.

#### Conflicts of interest

The authors have no conflicts of interest to declare.

#### References

- [1] Abaei G, Selamat A. A survey on software fault detection based on different prediction approaches. *Vietnam Journal of Computer Science*. 2014; 1(2):79-95.
- [2] Sarkar MK, Chatterjee T, Mukherjee D. Reverse engineering: An analysis of static behaviors of object oriented programs by extracting UML class diagram. *International Journal of Advanced Computer Research*. 2013; 3(3):135.
- [3] Catal C, Diri B. A systematic review of software fault prediction studies. *Expert Systems with Applications*. 2009; 36(4):7346-54.
- [4] Catal C. Software fault prediction: a literature review and current trends. *Expert Systems with Applications*. 2011; 38(4):4626-36.
- [5] Thakur M, Patidar K, Chouhan S, Kushwah R. A review for the quality estimation of object oriented programming. *International Journal of Advanced Technology and Engineering Exploration*. 2018; 5(40):50-4.
- [6] Thakur M, Patidar K, Chouhan S, Kushwah R. A clustering based on optimization for object oriented quality prediction. *International Journal of Advanced Technology and Engineering Exploration*. 2018; 5(41):62-9.
- [7] Zheng J. Predicting software reliability with neural network ensembles. *Expert systems with applications*. 2009; 36(2):2116-22.
- [8] Alsmadi I, Najadat H. Evaluating the change of software fault behavior with dataset attributes based on categorical correlation. *Advances in Engineering Software*. 2011; 42(8):535-46.
- [9] Haneef F, Kushwaha GR, Dubey AK. Analysis with data mining and ant colony algorithm for implementing of object pool optimization. In *international conference on communication systems and network technologies 2011* (pp. 313-7). IEEE.
- [10] Singh A, Bhatia R, Singhrova A. Taxonomy of machine learning algorithms in software fault prediction using object oriented metrics. *Procedia computer science*. 2018; 132:993-1001.
- [11] Chiacchio F, Aizpurua JI, Compagno L, D'Urso D. SHyFTOO, an object-oriented Monte Carlo simulation library for the modeling of Stochastic Hybrid Fault

- Tree Automaton. *Expert Systems with Applications*. 2020; 146:113139.
- [12] Malhotra R, Khanna M. Mining the impact of object oriented metrics for change prediction using machine learning and search-based techniques. In international conference on advances in computing, communications and informatics (ICACCI) 2015 (pp. 228-34). IEEE.
- [13] Kumar L, Rath SK. A model to assess the effectiveness of fault prediction techniques for quality assurance. In annual IEEE india conference (INDICON) 2015 (pp. 1-6). IEEE.
- [14] Harmanani H, Azar D, Zgheib G, Kozhaya D. An Ant colony optimization heuristic to optimize prediction of stability of object-oriented components. In international conference on information reuse and integration 2015 (pp. 225-8). IEEE.
- [15] Suresh Y. Software quality assurance for object-oriented systems using meta-heuristic search techniques. In international conference on applied and theoretical computing and communication technology (iCATccT) 2015 (pp. 441-8). IEEE.
- [16] Suresh Y. Software quality assessment for open source software using logistic & naive bayes classifier. In international conference on computation system and information technology for sustainable solutions (CSITSS) 2016 (pp. 267-72). IEEE.
- [17] Jain A, Chug A. Stepping towards dynamic measurement for object oriented software. In india international conference on information processing (IICIP) 2016 (pp. 1-6). IEEE.
- [18] Malhotra R, Shukla S, Sawhney G. Assessment of defect prediction models using machine learning techniques for object-oriented systems. In 5th international conference on reliability, infocom technologies and optimization (Trends and Future Directions)(ICRITO) 2016 (pp. 577-83). IEEE.
- [19] Nanda S, Bala A, Saxena S. Evaluation of feature selection techniques for software maintenance prediction. In 2<sup>nd</sup> international conference on computational systems and information technology for sustainable solution (CSITSS) 2017 (pp. 1-5). IEEE.
- [20] Singh P, Malhotra R. Assessment of machine learning algorithms for determining defective classes in an object-oriented software. In 6th international conference on reliability, infocom technologies and optimization (Trends and Future Directions) (ICRITO) 2017 (pp. 204-9). IEEE.
- [21] Boucher A, Badri M. Predicting fault-prone classes in object-oriented software: an adaptation of an unsupervised hybrid SOM algorithm. In international conference on software quality, reliability and security (QRS) 2017 (pp. 306-17). IEEE.
- [22] Kartha GP, Anjali C, Nair RV, Venkateswari S. Prediction of defect susceptibility in object oriented software. In international conference on networks & advances in computational technologies (NetACT) 2017 (pp. 467-72). IEEE.
- [23] Tripathi MK, Chaubisa D, Kumar L, Neti LB. Prediction of quality of service parameters using aggregate software metrics and machine learning techniques. In 15th india council international conference (INDICON) 2018 (pp. 1-6). IEEE.
- [24] McCormac J, Clark R, Bloesch M, Davison A, Leutenegger S. Fusion++: volumetric object-level slam. In international conference on 3D vision (3DV) 2018 (pp. 32-41). IEEE.
- [25] Aktaş F, Buzluca F. A learning-based bug prediction method for object-oriented systems. In IEEE/ACIS 17th international conference on computer and information science (ICIS) 2018 (pp. 217-23). IEEE.
- [26] Alakus TB, Das R, Turkoglu I. An overview of quality metrics used in estimating software faults. In international artificial intelligence and data processing symposium (IDAP) 2019 (pp. 1-6). IEEE.
- [27] Merzah BM. Software quality prediction using data mining techniques. In international conference on information and communications technology (ICOIACT) 2019 (pp. 394-7). IEEE.
- [28] KS VK. A method for predicting software reliability using object oriented design metrics. In international conference on intelligent computing and control systems (ICCS) 2019 (pp. 679-82). IEEE.
- [29] Samir M, El-Ramly M, Kamel A. Investigating the use of deep neural networks for software defect prediction. In ACS 16th international conference on computer systems and applications (AICCSA) 2019 (pp. 1-6). IEEE.