

A clustering based on optimization for object oriented quality prediction

Moksha Thakur^{1*}, Kailash Patidar², Sudeesh Chouhan³ and Rishi Kushwah³

M.Tech Student, Computer Science and Engineering, SSSIST, Sehore, Madhya Pradesh, India¹

Professor and HOD, Department of Computer Science, School of Engineering, Sri Satya Sai University of Technology & Medical Sciences, Sehore, Madhya Pradesh, India²

Assistant Professor, Department of Computer Science, School of Engineering, Sri Satya Sai University of Technology & Medical Sciences, Sehore, Madhya Pradesh, India³

©2018 ACCENTS

Abstract

In this paper software metric estimation has been established through object oriented parameters measures and quality estimation techniques. First the dataset has been prepared based on the object oriented programming. The four object oriented parameters have been considered for the experimentation are class, object, inheritance and dynamic memory allocation (DMA). The data is pre-processed based on these parameters and then cluster for the specified grouping. For the software strength analysis the data filtering has been done by the chi square testing method. Then software metrics F-Measure, Odd Ratio and Power have been used for the strength analysis. Finally by using binary particle swarm optimization (BPSO) quality estimation measures are optimized and retrieved the final accuracy based on the metrics input set. The result after BPSO shows better accuracy as comparison to the previous methods.

Keywords

DMA, Inheritance, Class, BPSO.

1.Introduction

In the software development the demand of object oriented modules has been increased because of the efficient software complexity handling. The demand is also increasing because the object-oriented software is more prominent than the procedural code [1]. It provides different mechanism to handle the complexity [2]. It supports the aims of development like maintainability and reusability [3].

In this outcome situated world, innovation is quickly upgrading which is setting the new worldview, because of which individuals are winding up more outcome driven as their way of life is winding up more reliable upon every new innovation in light of the fact that of their simple utilization [4]. This system is an accumulation of self-arranged modules called as objects that epitomizes information with usefulness to demonstrate the certifiable. It is programming which is produced by mix of articles that can be effectively demonstrated, reused, changed and supplanted by their prerequisite in the frameworks by the developer. It gives a view to the world as an arrangement of working together and agreeable objects [5-7].

In object oriented paradigm there are several concerns like security and exemption taking care of, must be executed in each question prompting repeated code also, loss of brought together security administration [8]. Moreover, the code that actualizes typical conduct and the code for its special case handlers are tangled together bringing about lost code understanding. Angle arranged programming, which bunches related usefulness that crosses particular limits, into concerns gives [8].

Furthermore, the code that implements normal behavior and the code for its exception handlers are tangled together resulting in a loss of code comprehension[8-12]. Aspect-oriented programming, which group's related functionality that crosses modular boundaries, into concerns provides a solution to these problems. Because of the large number of existing object-oriented systems, often it is more feasible to adapt an existing object-oriented system to some aspects of an aspect-oriented system rather than create a replacement fully aspect-oriented system.

In this paper an efficient framework has been presented for the object oriented quality prediction.

*Author for correspondence

2.Literature survey

In 2012, Muhammad et al. [13] suggested that there are several methods for the automatic modularization and architecture recovery of software systems. The common thing is that they support clustering. They suggested that the suitable algorithm and appropriate relationships are challenging issues. Despite the fact that scientists have utilized diverse calculations for modularizing object-situated programming frameworks, there has been generally little work to figure out which connections deliver better modularization comes about. The creators assess in this examination an extensive number of connections that may exist between elements in a protest situated framework, by partitioning the connections into various classifications. For modularization, tests are led utilizing different various leveled grouping calculations. The exploratory outcomes show the connections that enhance the nature of results for the calculations.

In 2012, Herraiz et al. [14] suggested that the statistical distributions in case of object-oriented systems, these distributions have been found to obey a power law. Their study is based on the Qualitas Corpus. For Java projects in the corpus the Chidamber and Kemerer metrics suite have been used. Our results demonstrate that the scope of high esteems for the diverse measurements takes after a power law circulation, though whatever is left of the range takes after a lognormal dispersion.

In 2012, Tagoug [15] suggested that the existing software maintenance can account for 60 percent of all exertion consumed by the assets utilized as a part of the system development life cycle. To enhance the circumstance, experts must outline framework that simple to keep up at the early phase of the improvement procedure. At the plan stage, a question arranged framework is decayed into subjects. Each subject is decayed into subsystems. A maintainable o-o framework is a framework where the change influences a less number of subsystems. In this work, viability metric is proposed. This metric measures the restriction of a change at the outline stage.

In 2012, Hussain et al. [16] suggested that the procurement of project charter is the staring task of the software developer's. It is important in the software requirement specification (SRS). The SRS of the association is a content record consolidating the necessities of the association. The product advancement of any data framework depends on the SRS of the customer association. This paper

endeavors to abstracts plan parts (Object class name, Object strategies, and its traits, Actors and interfaces of performers) from programming necessity particular. The goal of this paper is to build up a solitary semi mechanized technique for the deliberation of various useable parts from SRS, with the goal that they can be changed as model components. To give a semiotic domain to the plan of model components to the change of useable segments.

In 2014, Singh [17] suggested that the testing is as good as its test cases. It implies testing is profoundly needy upon shortcomings location capacity of experiments. Experiment age assumes an imperative part in testing process and is principle territory of research in the field of programming testing. The test cases diminish the odds of disappointment of the framework and guarantee the nature of the framework. They show the study of the different experiment age systems/strategies for question situated frameworks in more complete way.

In 2014, Khanna [18] investigated testability considering the metrics used in an object-oriented system. The thought is to give a review of question situated plan measurements with the prioritization of same keeping testability as the general objective. They have used analytic hierarchy process (AHP) method. They suggested that this metric is used for testability.

In 2015, Hu and Ding [19] mainly focus on regression testing suitable for class testing in object-oriented systems. They mine static call graphs and dynamic call trees to speak to the static highlights and dynamic trial of the program. They have analysis he method by graph analysis. These strategies enhance testing effectiveness for class testing from the accompanying angles: robotization; multi-edge assessments of experiments; change and administration of experiments; giving diverse prioritization criteria and streamlining criteria for relapse testing to meet diverse testing prerequisites and so on.

In 2016, Desai and Parmar [20] suggested that the object-oriented software frameworks are to be worked to last finish the time yet they will debase as much as any heritage programming framework. Programming reengineering essentially centers on re-executing more established frameworks to enhance or make it more viable. Refactoring is on sort of re-designing with-in an object oriented system.

In 2017, Jha and Ratha [21] suggested that the question arranged measurements plays the crucial role. Object masterminded estimations is an estimation device for the inquiry arranged perspective to help speedier headway. In this the different dissent arranged estimations are investigated taken after by the proposed systems from different works. The accurate characteristics are figured and choose the correct quantitative examination for C++ and Python.

In 2017, Mourad et al. [22] aims for finding the impact of clone which is used in refactoring on the test code size, in terms of number of operations, in object-oriented software. They have investigated the impact of clone refactoring. They have considering three attributes name coupling, complexity and size. Then clone impact refactoring on the test code size has been investigated and final the variations after clone refactoring have been investigated. They have used linear regression and k-nearest neighbors, naïve Bayes and random forest for the development of predictive and explanatory models. They have used an open source Java software system (ANT) for refactoring. The investigations show that there is a solid and positive connection between clone refactoring and the decrease of the test code measure. In 2017, Yadav et al. [4] suggested that the object oriented software system selection is a multiple-criteria decision analysis (MCDM) problem. It is analyzed based on the preference on different parameters according to the authors. Usefulness characteristic is one such parameter chose from Software quality model which comprises of a few different parameters by which nature of any programming can be broke down. This examination helps specialists in determination of programming frameworks whose worry is more about the utilitarian task of framework. A MCDM show comprises of AHP-FTOPSIS has been proposed for positioning of choices in view of the quantitative what's more, subjective parameters by the authors.

In 2017, Wang et al. [23] suggested that the classes' identification is the most important in a software system. They have proposed an approach utilizing different complex organize measurements to consequently distinguish key classes from worldwide and nearby angles. From the worldwide angle, the area of a class and its capacity to control the data stream of programming are basically considered. From neighborhood perspectives, we center on the cooperation's of classes with their neighbors, and also the many-sided quality of the class itself.

Analyses are performed on two java open-source ventures. Results demonstrate that this approach can precisely distinguish key classes contrasted and existing writing.

In 2018, Bai et al. [24] suggested that the development of the Internet has become the standard for integrating and exchanging data. They have studied the methodology of modeling fuzzy spatiotemporal data and transforming fuzzy spatiotemporal data from object-oriented databases to XML for this they have developed a fuzzy spatiotemporal data model.

3. Proposed work

This dissertation provides a way in the direction of modules quality estimation based on the object oriented measures. The modules conceded are object oriented programs in our case java based programming are used. The parameters used for pre-processing are class, object, inheritance and dynamic memory allocation (DMA). The classes are categorized based on class keyword. Object and dynamic behavior are categorized with the new keywords. Inheritance are categorized based on extends keyword. The pre-processing is performed based on these properties are input to the clustering mechanism to divide it in different modules. Then for any module the data is tested for the next phase. If it is qualified then it qualifies for the next input set. For this chi-square test has been applied. The selected values from the testing are then input for the module quality estimation. In our case F- measure (FM), Power (PO) and Odd Ratio (OR) have been considered. The values obtained from these metrics are normalized in the range of 0 and 1 for the next phase input. Then binary particle swarm optimization (BPSO) has been applied for the final data accuracy. *Figure 1* shows the flowchart of the working mechanism.

The parameters used for pre-processing are class, object, inheritance and dynamic memory allocation (DMA). The classes are categorized based on class keyword. Object and dynamic behaviour are categorized with the new keywords. Inheritance are categorized based on extends keyword. The pre-processing is performed based on these properties are input to the clustering mechanism to divide it in different modules. Then for any module the data is tested for the next phase. If it is qualified then it qualifies for the next input set. This process is shown in *Table 1*. Then qualifying values achieved are

shown in *Table 2*. FM, PO and OR values are shown in *Table 3, 4 and 5*.

The formula for FM, OR and PO are shown below. PR shows the precision value and RC shows the recall value.

$$F = \frac{2 * PR * RC}{PR + RC}$$

$$OR = \frac{2 * RC * (1-PR)}{(1-PR) * RC}$$

$$PO = \frac{((1-PR)^k - (1-RC)^k)}{k}$$

4.Result evaluation

Figure 2, 3 and 4 shows the overall accuracy in case of three different categorization retrieved from our approach after applying BPSO. We have considered the highest and lowest accuracy both. It depicts the performance which shows the good accuracy obtained after optimization. If we compare our result with the previous methodology then our method outperforms from the previous method. The comparison from previous methodology is shown in *Figure 5*.

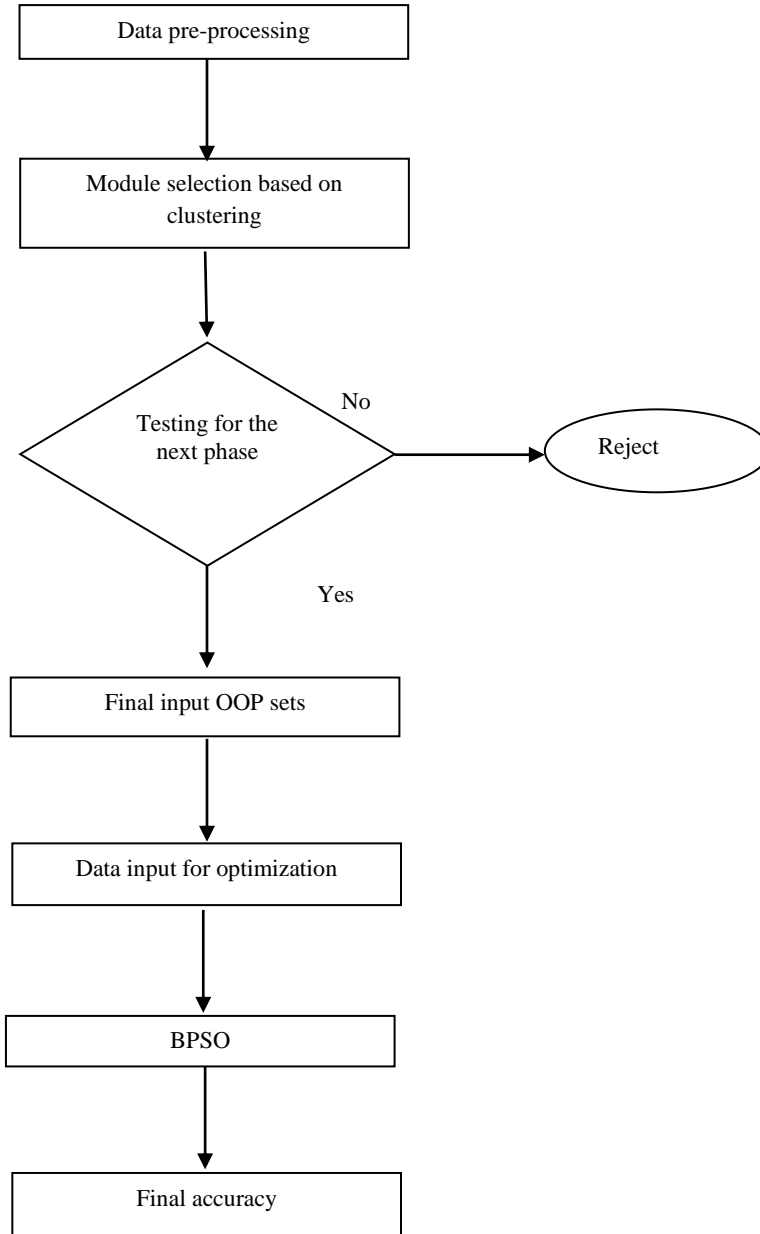


Figure 1 Data pre-processing flowchart

Table 1 Object oriented pre-processing

File Name	Size (Kb)	Class	Object	Inheritance	Dma
fifthi.java	11	1	45	1	45
P1.java	7	28	0	21	0
P10.java	8	14	14	14	14
P11.java	8	7	14	0	14
P12.java	8	6	12	0	12
P13.java	8	104	156	52	156
P14.txt	7	3	6	0	6
P15.txt	7	16	32	0	32
P16.java	8	4	0	3	0
P17.java	8	9	6	2	6
P2.java	7	29	58	0	58
P3.java	7	33	297	0	297
P4.java	7	1	9	0	9
P5.java	7	63	42	14	42
P6.java	7	396	198	297	198
P7.java	7	108	108	0	108
P8.java	7	76	0	0	0
P9.java	7	16	20	0	20
thirdi.java	11	1	45	1	45

Table 2 Qualification values

File Name	Class	Object	Inheritance	DMA
fifthi.java	3.2	320	3.2	320
P11.java	0.8	16.2	5	16.2
P12.java	0.2	9.8	5	9.8
P14.txt	0.8	0.2	5	0.2
P16.java	0.2	5	0.8	5
P17.java	3.2	0.2	1.8	0.2
P4.java	3.2	3.2	5	3.2
thirdi.java	3.2	320	3.2	320

Table 3 Result of FM ratio

S. NO	File Name	Class	Object	Inheritance	DMA
1	fifthi.java	0.43	42.67	0.43	42.67
2	P11.java	0.11	2.16	0.67	2.16
3	P12.java	0.03	1.31	0.67	1.31
4	P14.txt	0.11	0.03	0.67	0.03
5	P16.java	0.03	0.67	0.11	0.67
6	P17.java	0.43	0.03	0.24	0.03
7	P4.java	0.43	0.43	0.67	0.43
8	thirdi.java	0.43	42.67	0.43	42.67

Table 4 Result of OR ratio

S. NO	File Name	Class	Object	Inheritance	DMA
1	fifthi.java	0.14	0.98	0.14	0.98
2	P11.java	0.07	0.85	0	0.85
3	P12.java	0.02	1.02	0	1.02
4	P14.txt	0.07	0.02	0	0.02
5	P16.java	0.02	0	0.07	0
6	P17.java	0.14	0.02	0.12	0.02
7	P4.java	0.14	0.14	0	0.14
8	thirdi.java	0.14	0.98	0.14	0.98

Table 5 Result of PO ratio

S. NO	File Name	Class	Object	Inheritance	DMA
1	fifthi.java	-0.33	3008	-0.33	3008
2	P11.java	-0.14	4.63	-0.25	4.63
3	P12.java	-0.04	0.92	-0.25	0.92
4	P14.txt	-0.14	-0.04	-0.25	-0.04
5	P16.java	-0.04	-0.25	-0.14	-0.25
6	P17.java	-0.33	-0.04	-0.26	-0.04
7	P4.java	-0.33	-0.33	-0.25	-0.33
8	thirdi.java	-0.33	3008	-0.33	3008

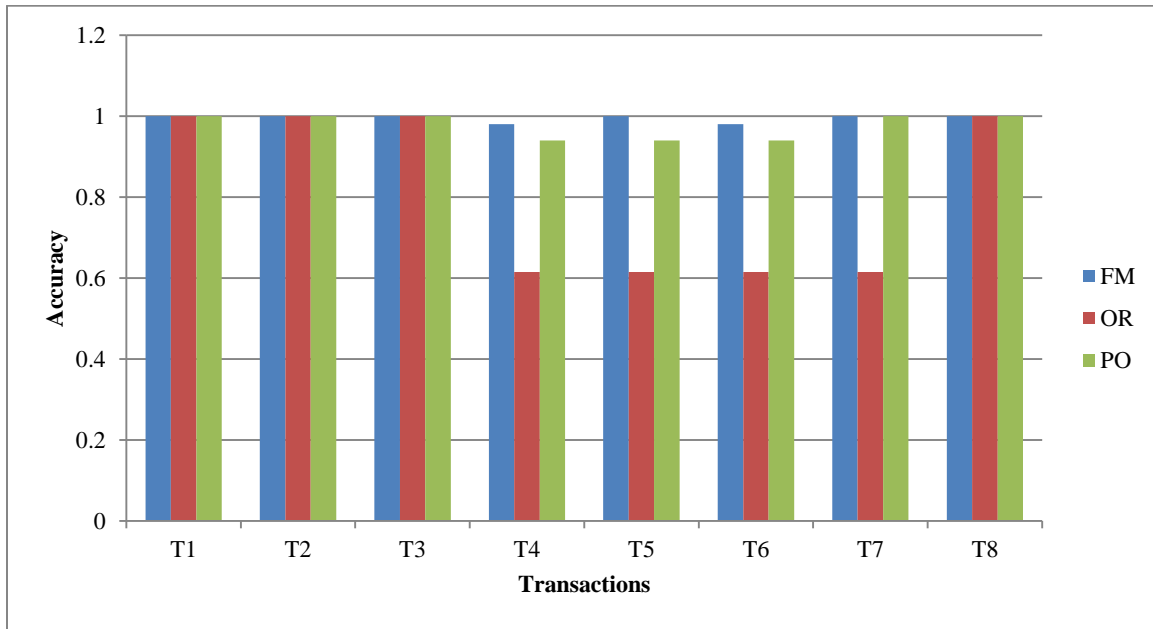


Figure 2 Overall optimized accuracy (First cycle)

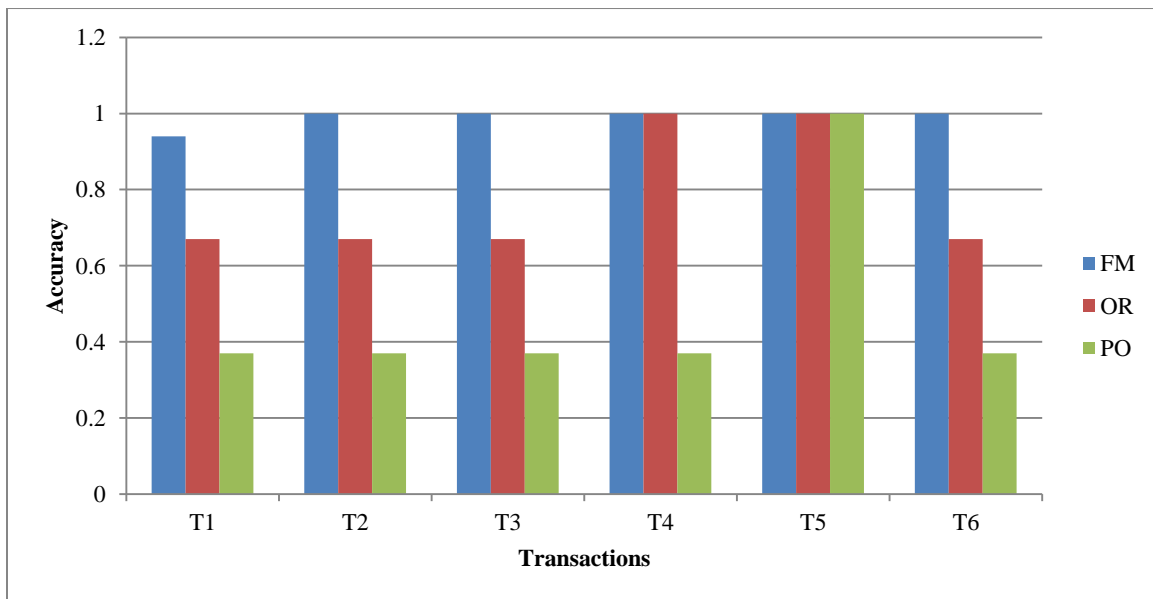


Figure 3 Overall optimized accuracy (Second cycle)

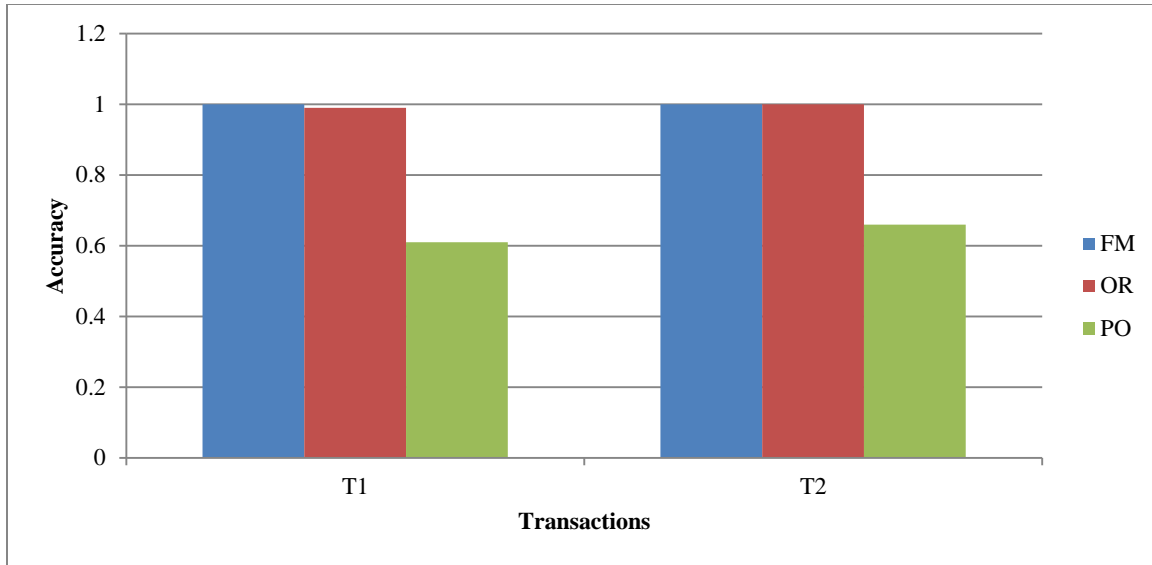


Figure 4 Overall optimized accuracy (Third cycle)

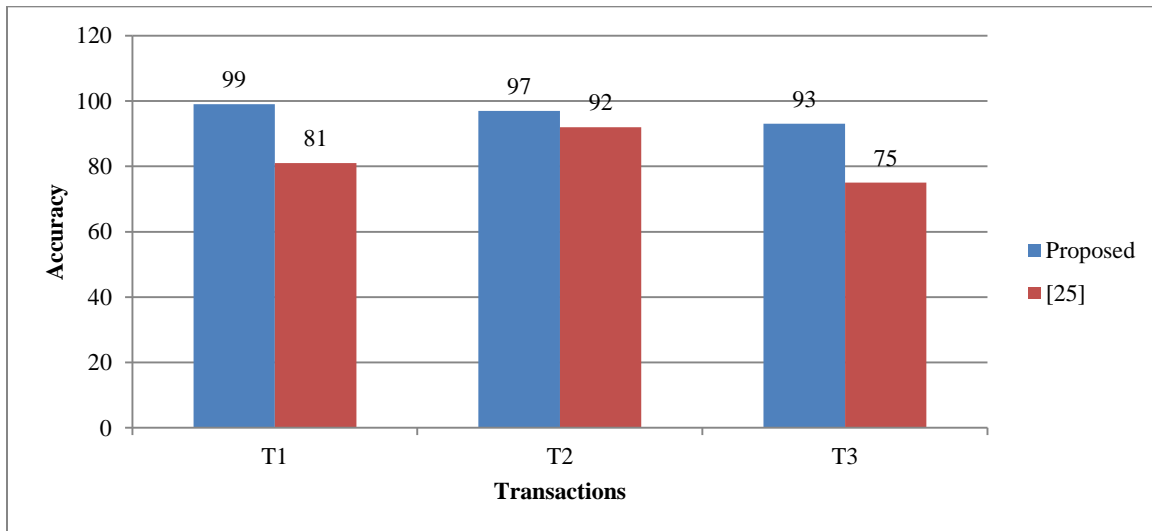


Figure 5 Comparisons from the previous method

5. Conclusion

In this paper a way for the object oriented measures quality estimation through binary particle swarm optimization (BPSO) have been presented. The four parameters have been considered which are class, object, inheritance and dynamic behaviour. The categorization has been partitioned in 10 block separation and accept large variant of modularity. It is separated by clustering algorithm. Then pre-processing has been applied based on object oriented measures. It is tested for the qualification by chi square test. For software quality estimation FM, PO and OR have been used. Then BPSO have been applied for the accuracy estimation. The comparative

results analysis and discussion shows the efficacy of our approach.

Acknowledgment

None.

Conflicts of interest

The authors have no conflicts of interest to declare.

References

- [1] Coad P, Yourdon E. Object oriented analysis. Upper Saddle River, NJ, USA: Yourdon Press; 1991.
- [2] Wilde N, Matthews P, Huitt R. Maintaining object-oriented software. IEEE Software. 1993; 10(1):75-80.
- [3] De Brito H, Marques-Neto HT, Terra R, Rocha H,

- Valente MT. On-the-fly extraction of hierarchical object graphs. *Journal of the Brazilian Computer Society*. 2013; 19(1):15-27.
- [4] Yadav N, Saraswat P, Tripathi RP. Estimating the functionality of object oriented system using MCDM approach. In *international conference on image information processing 2017* (pp. 1-6). IEEE.
- [5] Cloutier R, Griego R. Applying object oriented systems engineering to complex systems. In *annual IEEE systems conference 2008* (pp. 1-6). IEEE.
- [6] Meyer B. *Object-oriented software construction*. New York: Prentice hall; 1988.
- [7] Dubey SK, Rana A. Analytical evaluation of usability of object oriented software system. *International Journal of Computer Applications*. 2012; 49(21):7-13.
- [8] Millham R, Dogbe E. Aspect-oriented security and exception handling within an object oriented system. In *computer software and applications conference workshops 2011* (pp. 321-6). IEEE.
- [9] 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(12):135-41.
- [10] Srivastav MK, Nath A. Mathematical model of various statements of C-type language. *International Journal of Advanced Computer Research*. 2013; 3(13):79-87.
- [11] Sünnderhauf N, Pham TT, Latif Y, Milford M, Reid I. Meaningful maps with object-oriented semantic mapping. In *international conference on intelligent robots and systems 2017* (pp. 5079-85). IEEE.
- [12] Rathore MN, Gupta R. A novel class, object and inheritance based coupling measure (COICM) to find better OOP paradigm using JAVA. *International Journal of Advanced Computer Research*. 2011; 1(1):1-6.
- [13] Muhammad S, Maqbool O, Abbasi AQ. Evaluating relationship categories for clustering object-oriented software systems. *IET Software*. 2012; 6(3):260-74.
- [14] Herraiz I, Rodriguez D, Harrison R. On the statistical distribution of object-oriented system properties. In *international workshop on emerging trends in software metrics 2012* (pp. 56-62). IEEE.
- [15] Tagoug N. Maintainability assessment in object-oriented system design. In *international conference on information technology and e-Services 2012* (pp. 1-5). IEEE.
- [16] Hussain SN, Hundewale N, Aljahdali S, Ahmed KA. A methodology for the abstraction of design components from the software requirement specification to the object oriented system. In *international conference on software engineering and service science 2012* (pp. 311-5). IEEE.
- [17] Singh R. Test case generation for object-oriented systems: a review. In *international conference on communication systems and network technologies 2014* (pp. 981-9). IEEE.
- [18] Khanna P. Testability of object-oriented systems: an AHP-based approach for prioritization of metrics. In *international conference on contemporary computing and informatics 2014* (pp. 273-81). IEEE.
- [19] Hu T, Ding G. A novel dynamic analysis of test cases to improve testing efficiency in object-oriented systems. In *international conference on computer science and network technology 2015* (pp. 457-61). IEEE.
- [20] Desai AB, Parmar JK. Refactoring cost estimation (RCE) model for object oriented system. In *international conference on advanced computing 2016* (pp. 214-8). IEEE.
- [21] Jha S, Ratha BK. OOMT-Object oriented metric technique towards predictive & qualitative software. In *international conference on infocom technologies and unmanned systems (Trends and Future Directions) 2017* (pp. 548-51). IEEE.
- [22] Mourad B, Badri L, Hachemane O, Ouellet A. Exploring the impact of clone refactoring on test code size in object-oriented software. In *international conference on machine learning and applications 2017* (pp. 586-92). IEEE.
- [23] Wang J, Ai J, Yang Y, Su W. Identifying key classes of object-oriented software based on software complex network. In *international conference on system reliability and safety 2017* (pp. 444-9). IEEE.
- [24] Bai L, Jia Z, Liu J. Reengineering object-oriented fuzzy spatiotemporal data into XML. *IEEE Access*. 2018; 6:12686-99.
- [25] Mekruksavanich S. An adaptive approach for automatic design defects detection in object-oriented systems. In *international conference on digital arts, media and technology 2017* (pp. 342-6). IEEE.



Moksha Thakur had completed her BE from Ujjain Engineering Collage, Ujjain, M.P. in 2009 in the Department of Computer Science and Engineering. Currently she is pursuing M.Tech in Computer Science from SSSIST, Sihore.

Email: thakur.moksha000@gmail.com