Advances and Applications in Mathematical Sciences
Volume 20, Issue 12, 2021, Pages 3269-3287

AN INTEGRATED FRAMEWORK FOR SOFTWARE DEFECT ANALYSIS WITH AWARE OF SaaS PROVISIONING IN CLOUD ENVIRONMENT USING MADM METHODS

K. NITALAKSHESWARA RAO and CH. SATYANANDA REDDY

Department of Computer Science
and Systems Engineering
AU College of Engineering (A)
Andhra University, Visakhapatnam, A.P., India
E-mail: kolukulanitla@gmail.com

Abstract

The adamant issues in analysis of software defect raises copious solutions in a concur manner. The articulate procedures in cloud computing platform solves the service selection issues in an ameliorate way. Our research appraises the cognizance of Multi Attribute Decision Making methods in consort with our Proposed (MADM) Algorithm using TOPSIS Approach which is distinct in determining the factors of quality of services and quality assurance. This work edify the role of broker and software project developer with customized responsibilities in undertaking the software defect analysis for SaaS provisioning in cloud environment to an entail point. The gist in our research deliver the integrated solution for SaaS provisioning with decision making based on software data sets defects.

1. Introduction

The eloquence strategies in cloud computing from few decades represent prominent solutions for the present day computational needs. We know SaaS is the one of highly utilized service model where many small scale and large scale IT business enterprises adopt this feature for service sharing and consuming of resource utilization for economic benefits. Most of the impediments in SaaS are accountable to the cloud consumer. The variations

2010 Mathematics Subject Classification: 68.
Keywords: Machine intelligence, Quality assurance, Cloud broker, Software Project development, Software defect analysis, SaaS provisioning, Machine learning.
Received November 20, 2020; Accepted December 20, 2020

in consumer requirements for SaaS service consuming, obfuscate in judging of the services potentiality from the service provider. The untenable role of auditor in cloud computing states that a frequent audits enhances the performance and data consistence in the cloud using provenance data for performance assessment is most deciding factor for auditor [1]. The candour in analyzing software defects is a daunting work as the outcomes of software datasets exhibit correlated, complex sensitive information for post dissection studies as inputs to future software products. The software product development concern with different stakeholders, methodologies, models and infrastructure finally to output the efficient, effective and error free and economic product to the concerned software consumer [2]. We know software testing is different from software analysis, where testing is measurable attribute with different dependencies involved in software product and software analysis is a broad view of hypothesis solution about the product. Our concern is to develop an integrated framework synchronizing software defect analysis study with cloud computing platform in SaaS provisioning. The proposed framework targets the outcomes of SaaS provisioning based on software defects data set of in cloud computing platform. We involve SaaS broker for intermediating services between cloud consumer and cloud service provider and pass the software defect analysis report to the involved parties of our framework about the occurred software defects, and decision makings during utilization of services. The organization of our research scripts employs with six sections. The primary section introduction projects the narrowness of our research idea. The background section explains the usage of MADM methods and their uses in decision making. The review of literature renders the techniques and research ideas of different researches to our problem. The framework sections narrate our framework infrastructure graphically and lucidly. The methods section delivers the deployment of problem formulation and implementation procedures. The result analysis articulates our framework outcomes. The conclusion section portraits the further work highlights.

2. Background

In the contribution to our framework we had examined many research scripts, in finding an optimal solution and we targets in rendering of Multi attribute decision making methods (MADM). The elementary studies of multi criteria decision making methods involves with different methods i.e., max min, linear assignment, additive weighting and weighted product etc. MADM are unique synthesis criterion methods exhibit the advantages of logical planning of framework, synthesis of information with value based planning, Comparisons with inclusion of subjectivity of attributes to outcome and acceptable optimal solutions. The classification of MADM was grouped into individual and grouping where the individual may hold the certainty and uncertainty problem. The uncertainty group deals with Bayesian probabilistic, Demspter-Schafer evidence theories and Fuzzy set analysis. MADM used for decision making, the magnitude of these strategic process methods solves many uncertainties of data complexities. There are different MADM methods which can be employed depends on that computing situation. The analytical hierarchy process method helps in subject of decision making with available consistence preferences. A normalization procedure of attributes with pair wise comparing the attributes treated with inclusion of weights and random index (RI) values. The improved optimal method TOPSIS (Technique for Order Preference by Similarity to Ideal Solution method) produced by Hwang and Yon. Euclidean distance method incorporates on multiple chosen attributes to extract outcomes with positive and negative hypothetical solution. Data Envelopment Analysis method used for finding the multiple alternatives relative efficiency in decision making analysis. Hashimoto proposed ranking voting system from an aggregation of terms or attributes. In DEA method each attribute is compared with best alternative where these alternatives are grouped into higher alternative groups and lower value groups. The Gray relation space identified as more effectiveness technique for evaluation method (GRA) Grey relational analysis. Pre processing of data is clearly implemented and gray relational coefficient was calculated for all the chosen alternatives. Most of the studies says that traditional (GRA) method gives better results for the quantitative attributes than the qualitative attributes. The inception of attributes for (GRA) taken as larger the better, and smaller the better to that situation [3].

3. Review of Literature

The exposing of risks in software landscape discloses the defects and quality assurance factors. The judging of giving reliable quality by project manager was deployed with improved correlation over sampling method, which specifies the desired outcomes [4]. The quality of software address the issues of risk tendency, fault detecting which are depended on the input of the dataset values. The complexity arises between the selection software data set values are identically equal but not same. Many Artificial Intelligence (AI) techniques such as class imbalance methods are notified in this script. An under sampling strategy was implemented to sort out the software data set errors [5]. The prediction of quality software before testing triggers the economical benefit for the project managers. Many imbalanced, classification algorithms are rendered with simplified approaches i.e., Random over sampling, TOMEK, SMOTE, SMOTE and ENN. The above discussed techniques target the accuracy in incrementing software quality with fault predictions [6]. The consistency analysis between balanced and imbalanced sets reveals the performance factors of used classifiers. In this research the openly available data set was taken as sample and simulated with (DT) Decision tree method, Random forest (RF), Ada Boost (AdB), Gradient Boost (GB), Naive Bayes (NB), *K*-nearest neighbouring method (*K*-NN), and Artificial neural networks (ANN). A method of voting with logistic regression and support vector machine was used. [7].

The measurement of dataset defects collides with weighted per class, which tags the depth rate of inheritance, child nodes, computation involved in object based classes, class response status, analyses the cohesion, afferent verses efferent coupling, usage of public methods, analyzing the lack of cohesion in lines of code, aggregation rate, rate of function abstraction, pinning of cohesion in methods, classes and coupling the rate of inheritance, methods, complexity estimations and McCabe cyclomatic complexity [8]. In addressing of software quality of data characteristics influences more. The empirical software defect analysis strategies not always exhibit same optimal outcomes for the quality deficiencies off those particular projects which are based on bug reports.

The motivate spirit in this paper reveals the McCall software quality dimensions as revision of the product, transition of product and operation of the product. The modularity feature in software projects reveals the faults which affect the quality assurance during delivering of software product. A linear support vector machines model was applied in defect prediction effectively. A feature based prediction technique F-Score was used in determining the metrics in software modules [9]. The eminent usage of Fourier's learning algorithm with Boolean methods admits the prominent role compared to machine learning algorithm. Fourier's coefficient is calculates to obtain the predictive function. The predictive feature selection method evaluated with information gain, information entropy and conditional entropy [10]. A conceptual economic model with statistic approaches was used to find out the software defects monitoring and rectifying them in an economically beneficial manner for software companies. The two agendas in this script delivery statistical method for min of fault cost and defect finding improve software quality role during software development life cycle process [11].

The simplicity of finding the density of defects with statistical approaches evaluated with regression method was explained. The experimental results are circled with SPSS tool for normality test of data sets acquired from Promise software and final outcomes reveal the skewness factor and other fault prediction correlativeness. [12] The enormous growth in software developments and their defects are traced by knowledge discovery and data mining techniques from few years. MCDM play an alternative eminent role in selection criteria of different attributes based on empirical results and user preferences, TOPSIS, DEA and Elimination choice expression reality are taken as experimental study for software defect detection rather than tradition AI methods [13]. There are many methods in determine the reliability of software and quality. The limitations with Queue models, neural networks, Wavelet forms, elevate the scientific mathematical method called Grey forecasting method [14]. Representing TOPSIS method uses and recommending for decision making improve software project optimization in multi criteria analysis. Efficiency of software can be chosen to this method for better simulation desired outcomes. The program in the software must have specification of correctly, accurately, reliable, universal, compliance, secure useful, effectiveness, testable and adoptable. As this TOPSIS method embodied in find positive and negative hypotheses solution based on Euclidean distance [15]. In decision making scenario relative sensitive analysis between attributes are computed using Grey relational degree method. This approach is applied on real time congestion on urban traffic based on attributes of length of queue, parking rate, delay run, saturation slot, sharing time and speed [16].

One of the valid measures is comparison of software product with empirical completed software product which showcases the software project performance and quality. There are many pivot factor which form as a milestones in addressing software project quality. A relational degree is calculated for each pivot factor where the input data is uncertain [17]. Traditionally Grey relational analysis was used extensively as the statistical procedures are uncertain. Many types of GRA are proposed with non weighted linear, weights with max, nonlinear, distance based and correlation approaches. The proposed approaches performed the outcomes to the desired level [18]. This research script involves in tracking of defects at software requirement specification part and simulated in a tool which addresses most of the defects. Check list reading technique, Defect reading technique were compared with experimental simulation which also speaks about differentiation of defects [19]. The milestones in software defect analysis are aggregated with identification techniques of defects, affecting attributes that causes defects, and identifying the data used for defects. The triggering point of this paper predicts the defects in the software with a learning approach of different related research scripts, mechanisms and techniques [20].

4. Framework

We know the definition of defects embodied with error, fault, and failure. The misconception at the basic level of software requirement which may be present and involved in the thought process of user is called error. The unconditional support expressions of these errors are known as fault. An operational system which is not to the expected behaviour level status is called failure. The below figure depicts the conceptual view of framework about software defect analysis with aware of SaaS provisioning in cloud platform. The below figure differentiate with two levels and bridging the problem solving with and integration mode. The state holders and their roles in our framework are precisely stated, they are SaaS Consumer or cloud consumer, SaaS Broker, Service Provider, Software Facilitator, Software Projects Management, Software Project Analyzer / Tester and Software

Project Manager.

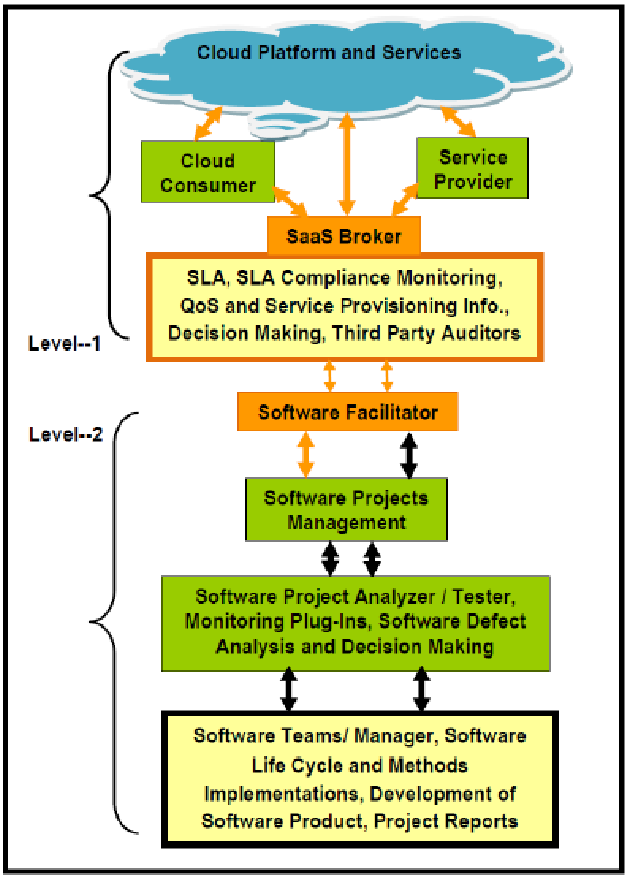


Figure 1. Conceptual view of framework about software defect analysis with aware of SaaS provisioning.

SaaS consumer or Cloud Consumer is a stake holder who utilizes the service provided by the service provider and pay as per the usage in cloud environment. SaaS provider is one who provides various services to the consumer in cloud environment. SaaS broker deals with intermediation of services between consumer and provider as both parties involves with other burdensome work [21]. The responsible role of SaaS broker was accepted by the consumer and provider mediates their services with regards to SLA, SLA compliance monitoring, QoS and Service provisioning information, decision making and involving with the third party auditors etc. The primary role of

SaaS broker facilitates the software products to service provider from software facilitator. We involve a software facilitator who forwards the software products to the SaaS broker from different software project managements. Software project management involves with software project analyzers, testers, monitoring plug-ins, software defect analysis report and decision making teams. The software project management was constituted with different software teams, managers, software life cycle methods, implementations and project reports.

The framework involves and exposes the crucial roles of two major stake holders, software facilitator and broker. We know SaaS is cloud service model which involved between cloud consumer and provider with an intermediating service of SaaS broker. The consumer requests for the services in utilization of resources from service provider through broker. In the context of quality of service and quality assurance the service provider admits requests to broker about software products quality assurance that are provisioned for the consumer preferences through concern project of software product reports. The SaaS broker with the help of software facilitator gathers the necessary software product and collects its software defects analysis reports and forwards to the service provider. The delegate role of software facilitator was given more importance by software projects managements which involves with different operational, management and decision making activities.

5. Methods

We know that SaaS services in cloud computing prone to little outages and performance issues which reflects the QoS and Quality assurance factor. To study this quality assurance narrowly we identified the problem of software defects from software is a major issue. The cerebration of solving identified problem in cloud computing is another issue. We propose an integrated framework in an altruistic way for crescendo of quality assurance. In our framework we consort with many research scripts in detonation of this problem and cull an optimal solution with optimal MADM methods.

Table 1. Snapshot of Software Defects Datasets (SDD).

S.NO	NAME OF THE DATA SET (PROMISE REPOSITORY)	# ATTRIBUTE-1 (FEATURES)	# ATTRIBUTE-2 (MODULES)	# ATTRIBUTE-3 (DEFECTS)	# ATTRIBUTE-4 (IMBALANCE RATIO)
1	AR1	29	121	9	13.4
2	AR3	29	63	8	7.87
3	AR5	29	36	8	4.5
4	CM1	37	327	43	7.78
5	DATATRIEVE	9	130	11	11.81
6	DESHARNAIS	11	81	10	8.1
7	JM1	21	7782	1672	4.65
8	KC1	21	2109	326	6.46
9	KC1-DEFECTIVE	94	145	60	2.41
10	KC1-TOP5PER	94	145	8	18.12
11	KC3	39	194	36	5.38
12	MC1	38	1988	46	43.21
13	MC21	39	125	44	2.84
14	MW1	37	253	27	9.37
15	PC1	37	705	61	11.55
16	PC3	37	1077	134	8.03
17	REUSE	27	24	9	2.66

The above table is acquiring from the research script for our problem solving [4]. The Table 1: is the snapshot data which is used for problem input data for solving.

The limn of our framework process as follows

a. Service provider acquires software products and their defective dataset reports with help of SaaS broker from software facilitator.

b. The decision making and monitoring services of broker identifies the consumer requests helps in provisioning of resources from provider towards consumer preferences by awarding weights to SDD.

c. The decision making of the broker checks for the QoS, and Quality assurance factors with regards to software defects analysis of software products.

d. Based on QoS and Quality assurance factors of software defects analysis was carried by broker through MADM method and justify the best software product and analysis their defects through reports.

e. The obtained defect reports are shared to the service provider and the service provider may have an option to continue or discontinue the product acquiring from that particular Software developer.

The above framework process projects our research work process in step by step order. The framework process steps states (a) Connectivity involvements through web portals. (b) broker awards weights to the SDD, (c) Brokers elevate the Quality assurance factors and analyze the given SDD table (d) MADM method is implemented to compute closeness and ideal solutions (e) Broker pushes the outcome reports to service provider.

Note. The Table 1: consists of six columns with serial number of Datasets, Name of the Datasets, #Attribute 1(features), #Attribute 2(modules), #Attribute 3(defects), #Attribute 4(imbalance ratio). The whole table consists of 17 datasets with attributes. The broker awarded weights (cumulative addition) should be equal to one.

The broker award weights on the (SDD) attributes for decision making analysis i.e., for #Attribute 1(features) 0.305, #Attribute 2(modules)0.267, #Attribute 3(defects)0.197, #Attribute 4(imbalance ratio)0.231.

Algorithmic representation of Proposed (MADM) Algorithm using TOPSIS Approach for our Framework

Step 1. The attributes form the (SDD) attributes segmented in the form of decision matrix (DM) where (Q) represent the attributes of (SDD).

Step 2. Normalization of (DM) is carried.

Step 3. Weighted Normalized decision matrix is obtained form the preferences of the broker.

Step 4. Positive and Negative ideal solutions are Calculated.

Step 5. Finding a Euclidean distance for the above obtained Positive and Negative solutions of criteria.

Step 6. Calculate the relative closeness from the above obtained alternatives and rank them.

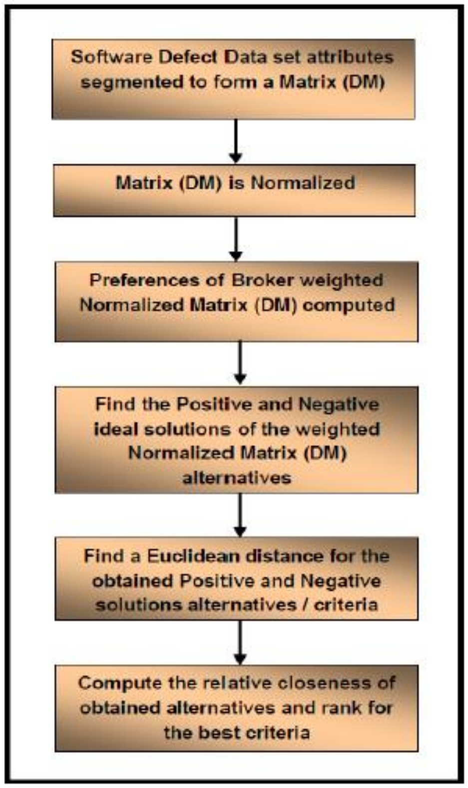


Figure 2. Diagrammatic view of Proposed (MADM) Algorithm using TOPSIS Approach for our Framework.

Mathematical formulations for Proposed (MADM) Algorithm using TOPSIS Approach for our Framework problem (Technique for Order Preference by Similarity to Ideal Solution method)

Step -1. Establish a decision matrix for the given list of Software defect Dataset (SDD) attributes

$$DM = \begin{pmatrix} a_{11}, a_{12}, \ldots, a_{1n} \\ a_{21}, a_{22}, \ldots, a_{2n} \\ a_{m1}, a_{m2}, \ldots, a_{mn} \end{pmatrix}$$

where j = criteria $(Q)(1, 2, 3, \ldots, n)$ $i = SDD$ $(1, 2, 3, \ldots, m)$.

Step -2. Normalize the (DM) Decision matrix, normalized value r_{ij} of each variable a_{ij} is calculated through

$$r_{ij} = r_{ij}/\text{square root of} \sum_{i=1}^{m} a_{ij}^2 (j = 1, 2, \ldots, n).$$

Step -3. Obtain the weighted normalized decision matrix

$$e_{ij} = w_j * r_{ij} \quad i- = 1, 2, 3, \ldots, m \text{ and } j = 1, 2, 3, \ldots, n$$

w_j is the weight of j^{th} criteria.

Step -4. Identify the Positive and Negative ideal solutions

Positive $A^+ = (e_1^+, e_2^+, \ldots, e_n^+) = \{\max (e_{ij})/j_e J1, \min (e_{ij})/j_e J2\}$.

Negative $A^- = (e_1^-, e_2^-, \ldots, e_n^-) = \{\min (e_{ij})/j_e J1, \max (e_{ij})/j_e J2\}$.

Step -5. Euclidean distance from Positive and Negative ideal solutions of each alternative (SDD) can be calculated as

$$Ed_i^+ = \text{square root of} \sum_{j=1}^{n} (e_{ij} - e_j^+)^2 i = 1, 2, 3, \ldots, m.$$

$$Ed_i^- = \text{square root of} \sum_{j=1}^{n} (e_{ij} - e_j^-)^2 i = 1, 2, 3, \ldots, m.$$

Step -6. Calculate the relative closeness to the ideal solutions

$$IS_j = ED_j^- / Ed_j^+ + Ed_j^- \;\; j = 1, 2, \ldots, m.$$

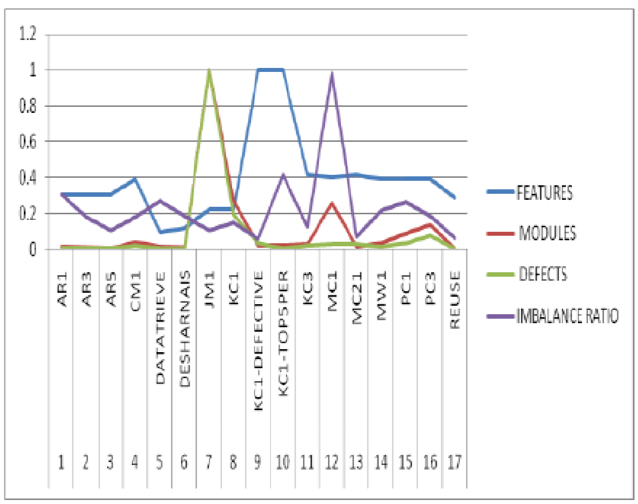
Largest IS_j should be ranked as top

5. Results and Discussions

The results section is organized with data tables and graphs. The conceptual explanation of the above defined method was tabulated with mathematical computations. The review of our framework outcomes are demonstrated in tables and graphs with precise touch.

Table 2. Normalized Software Defects Datasets (SDD).

S.NO	NAME OF THE DATA SET	FEATURES	MODULES	DEFECTS	IMBALANCE RATIO
1	AR1	0.308510638	0.0155487	0.0053828	0.304545455
2	AR3	0.308510638	0.0080956	0.0047847	0.178863636
3	AR5	0.308510638	0.0046261	0.0047847	0.102272727
4	CM1	0.393617021	0.04202	0.0257177	0.176818182
5	DATATRIEVE	0.095744681	0.0167052	0.0065789	0.268409091
6	DESHARNAIS	0.117021277	0.0104086	0.0059809	0.184090909
7	JM1	0.223404255	1	1	0.105681818
8	KC1	0.223404255	0.27101	0.1949761	0.146818182
9	KC1DEFECTIVE	1	0.0186327	0.0358852	0.054772727
10	KC1-TOP5PER	1	0.0186327	0.0047847	0.411818182
11	KC3	0.414893617	0.0249293	0.0215311	0.122272727
12	MC1	0.404255319	0.2554613	0.027512	0.982045455
13	MC21	0.414893617	0.0160627	0.0263158	0.064545455
14	MW1	0.393617021	0.0325109	0.0161483	0.212954545
15	PC1	0.393617021	0.0905937	0.0364833	0.2625
16	PC3	0.393617021	0.1383963	0.0801435	0.1825
17	REUSE	0.287234043	0.003084	0.0053828	0.060454545



Graph 1. Normalized Software Defects Datasets (SDD).

X-axis represents the Name of the Datasets

Y-axis represents the Normalized scale from 0 to 1.2.

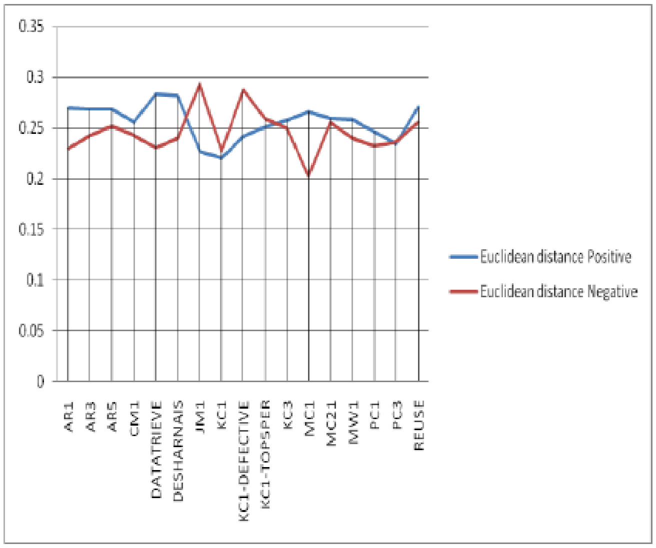
For admitting the consistency in data computation Table 2: represents the normalized values of attributes i.e., features, modules, defects and imbalance ratio. The computations of normalized values of all the provided attributes lie between 0 and 1.

The Graph 1: represents the normalized value of software defect datasets with the attributes in a normalized view.

Table 3. Positive and Negative solutions of Datasets (SDD).

NAME OF THE DATA SET	Euclidean distance Positive	Euclidean distance Negative
AR1	0.269612009	0.2299164
AR3	0.268388257	0.2430144
AR5	0.268359885	0.2516352
CM1	0.255572994	0.2424654
DATATRIEVE	0.283301677	0.2309492
DESHARNAIS	0.281307016	0.2400048
JM1	0.226640992	0.2937794
KC1	0.220545339	0.2276017
KC1-DEFECTIVE	0.242192987	0.2876089
KC1-TOP5PER	0.250679919	0.2593781
KC3	0.257611417	0.2497064
MC1	0.266179863	0.2028422
MC21	0.259384021	0.255704
MW1	0.258371292	0.2399162

PC1	0.246480012	0.2325272
PC3	0.234273252	0.2360084
REUSE	0.269921247	0.2560463



Graph 2. Positive and Negative solutions of Datasets (SDD).

X-axis represents the Name of the Datasets

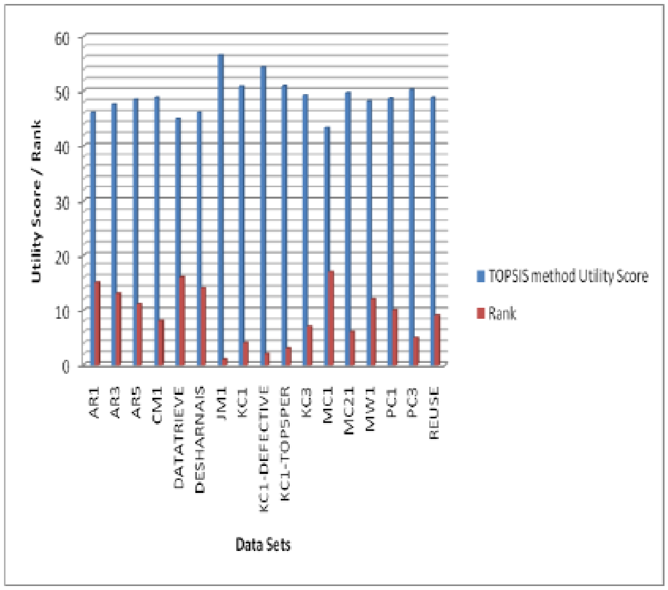
Y-axis represents the Positive and Negative solutions.

The Table 3 represents the positive and negative solution of the attributes of a particular (SDD) and the Graph 2 represents the positive and negative solutions lie between 0.2 to 0.3.

Table 4. Utility Score in acquiring ideal solution of the particular Software Defects Datasets (SDD).

NAME OF THE DATA SET	TOPSIS method Utility Score	Rank
AR1	46.02669186	15
AR3	47.51919338	13

AR5	48.39184072	11
CM1	48.68408011	8
DATATRIEVE	44.90983007	16
DESHARNAIS	46.03862205	14
JM1	56.45039968	1
KC1	50.78728055	4
KC1-DEFECTIVE	54.28612217	2
KC1-TOP5PER	50.8526629	3
KC3	49.22089987	7
MC1	43.2479064	17
MC21	49.64277417	6
MW1	48.14814498	12
PC1	48.54357433	10
PC3	50.18448371	5
REUSE	48.68100923	9



Graph 3. Utility Score with ideal solutions of particular Software Defects Datasets (SDD).

X-axis represents the Name of the Datasets

Y-axis represents the Utility Score and Ranks given to the particular (SDD).

Table 4. represents the ideal solutions for the input(SDD) which is notified as Utility Score. Utility Score projects the value of each dataset credibility in terms of ranking QoS and analyzing the defect datasets. The ranks are based on the Utility Score. Depending on Utility Score of the particular dataset the considering issues may be in terms of service provisioning, quality assurance, and analyzing rate of software defect datasets can be reviewed.

6. Conclusions and Future Scope

In the goal of achieving the quality assurance our integrated framework is a new idea to deal with problem of software defect analysis with aware of SaaS provisioning in cloud environment. The advantages of MADM method enlights the path for decision making to solve the uncertainties in software defect analysis. In the contribution to our framework many research ideas/techniques was examined but TOPSIS (MADM) method fit to the mark. The conceptual view of our framework admitted with simple step by step process. The framework outcomes are transparent and stood with mathematical baselines to elevate the problem solving innovatively. The result of the framework materializes in computing of software dataset positive and negative solutions and finding the closeness in the datasets with an ideal solution. The future scope promotes and announces the applicative advantages of MADM methods to software defect analysis problem. To aggregate the desired outcome different MADM method used to showcase their optimality in solution. Finally our research opens the doors to solve the problem in an integrated manner.

References

[1] Vadlamudi, Divya and K. Thirupathi Rao, Provenance Aware Audit Trail Framework For Saas Providers Efficiency Assessment In Cloud Environment, Journal of Critical Reviews 7(12) (2020), 1191-1196.

[2] Kiranbabu, Movva Naga Venkata and K. V. V. Satyanarayana, A productive survey on resource usage and migration by client towards csp resource allotments in risk state of cloud computing based on sla's, International Journal of Pure and Applied Mathematics 116(21) (2017), 439-447.

[3] Rao and Ravipudi Venkata, Decision making in the manufacturing environment: using graph theory and fuzzy multiple attribute decision making methods, Springer Science and Business Media, 2007.

[4] Rao, K. Nitalaksheswara and Ch Satyananda Reddy, An efficient software defect analysis using correlation-based oversampling, Arabian Journal for Science and Engineering 43(8) (2018), 4391-4411.

[5] Rao, K. Nitalaksheswara and Ch Satyananda Reddy, A novel under sampling strategy for efficient software defect analysis of skewed distributed data, Evolving Systems 11(1) (2020), 119-131.

[6] Pradeep Singh, Learning from Software defect datasets, 2019 5th International Conference on Signal Processing, Computing and Control (ISPCC). IEEE, 2019.

[7] Md Fahimuzzman Sohan, et al., Revisiting the class imbalance issue in software defect prediction, 2019 International Conference on Electrical, Computer and Communication Engineering (ECCE). IEEE, 2019.

[8] Bachmann, Adrian and Abraham Bernstein, When process data quality affects the number of bugs: Correlations in software engineering datasets, 2010 7th IEEE Working Conference on Mining Software Repositories (MSR 2010). IEEE, 2010.

[9] Sonali Agarwal and Divya Tomar, A feature selection based model for software defect prediction, assessment 65 (2014).

[10] Yang, Kang, et al., Software Defect Prediction Based on Fourier Learning, 2018 IEEE International Conference on Progress in Informatics and Computing (PIC). IEEE, 2018.

[11] Marandi, Arun Kumar and Danish Ali Khan, An Approach of Statistical Methods for Improve Software Quality and Cost Minimization, International Journal of Applied Engineering Research 12(6) (2017), 1054-1061.

[12] Mandhan, Neeraj, Dinesh Kumar Verma and Shishir Kumar, Analysis of approach for predicting software defect density using static metrics, International Conference on Computing, Communication and Automation, IEEE, 2015.

[13] Peng, Yi, Guoxun Wang and Honggang Wang, User preferences based software defect detection algorithms selection using MCDM, Information Sciences 191 (2012), 3-13.

[14] Denghua Mei, Software reliability estimation in grey system theory, 2007 IEEE International Conference on Grey Systems and Intelligent Services IEEE, 2007.

[15] Shafagat Mahmudova, Application of the TOPSİS method to improve software efficiency and to optimize its management, Soft Computing 24(1) (2020), 697-708.

[16] Mao, Shuhua and Xinping Xiao, Attribute Relative Sensitivity Analysis Based on Grey Relational Degree, 2010 International Conference on Computational Intelligence and Software Engineering. IEEE, 2010.

[17] Slavek, Ninoslav and Alan Jović, Application of grey system theory to software projects ranking, Automatika: časopis za automatiku, mjerenje, elektroniku, računarstvo i komunikacije 53(3) (2012), 284-293.

[18] Hsu, Chao-Jung and Chin-Yu Huang, Comparison of weighted grey relational analysis for software effort estimation, Software Quality Journal 19(1) (2011), 165-200.

[19] Alshazly, A. Amira, Ahmed M. Elfatatry and Mohamed S. Abougabal, Detecting defects in software requirements specification, Alexandria Engineering Journal 53(3) (2014), 513-527.

[20] Singh, Pradeep Kumar, Dishti Agarwal and Aakriti Gupta, A systematic review on software defect prediction, 2015 2nd International Conference on Computing for Sustainable Global Development (INDIACom), IEEE, 2015.

[21] Elarbi Badidi, A broker-based framework for integrated sla-aware SaaS provisioning, arXiv preprint arXiv:1605.02432, (2016).