Journal of Autonomous Intelligence (2024) Volume 7 Issue 5
doi: 10.32629/jai.v7i5.1510

ORIGINAL RESEARCH ARTICLE

Gradient integrated regression sustainable approach with machine learning towards software quality assurance

Alankrita Aggarwal[1,*], Vijay Bhardwaj[1], Sandeep Singh Bindra[2], Rajender Kumar[3], Preet Kamal[1]

[1] *Computer Science and Engineering, University Institute of Sciences, Apex Institute of Technology, Chandigarh University, Gharuan, Mohali, Punjab 140413, India*

[2] *Panipat Institute of Engineering & Technology, Samalkha, Haryana 132101, India*

[3] *Computer Science and Engineering, CUIET, Chitkara University, Punjab 140401, India*

*** Corresponding author:** Alankrita Aggarwal, alankrita.agg@gmail.com

ABSTRACT

With the increase of enormous software suites and modules, there is a need to evaluate the cumulative quality and overall performance of newly developed modules using advanced algorithms and approaches including machine learning, statistical analysis, deep learning, and many others. In this work, the software quality assurance with the dataset's evaluation using statistical analysis and regression integrated approach is presented. The aim is to acquire a much-enhanced acceptance of the subject matter by providing an investigation of software quality analysis using regression which is done in the last fifteen or twenty years. The combination of applying regression technique along with machine learning to outline its application sphere of influence, the type of metrics used, the application strategy, and the stage of the software development progression wherever they are useful. An outcome on or after going through around five hundred papers, a set of around more than fifty papers are unfolding the use of more than thirty software quality analysis can be identified. On the other hand, the lowest amount is given to maintain and apply software regression techniques in the industry and education. The graphical representation and the results showing the good and innovative performance of the gradient integrated regression approach with machine learning technology for increasing the quality of software.

Keywords: software quality assurance; regression; machine learning; gradient integrated regression; software project management; software quality sustainable analytics

ARTICLE INFO

Received: 25 December 2023
Accepted: 1 February 2024
Available online: 6 June 2024

1. Introduction

Software service management and code quality with web-based applications and code evaluations is an important task in developing distributed applications that can be accessed via the Internet. As a part of it, software risk management is a prominent area in engineering science that involves historical detection, processing and risk management, and software creation vulnerabilities.

In network engineering, software defect estimation is used to forecast software module deformity i.e., faults present during or after the distribution of the module. In this regard, the prediction methods for fault detection can be followed to assess faults in advance for the maintaining the good software.

Also, customer satisfaction is important for retaining high-quality software apps. Few major companies employ prediction progression schemes because of the regular release of revisions of their applications and consist of fewer resources instead of forecasting

software malformation process defects manually[1]. The performance and assorted software quality factors depend on multiple key points including throughput, robustness, accessibility, availability, interoperability, reliability, latency, accuracy, scalability, and integrity. When quality assurance optimization is done which is an important research area for many years and ages, new information in software development can be added, as well as in software quality has been observed and noticed that interesting new information in these areas such as incorporating psychosomatic factors, detecting pseudo experienced code, and detect code with lesser failure risk probability rate can be determined between various epochs and error rates. These are moderately important and widely integrated into software projects. The achievability and usefulness of fuzzy-based testing can be done for open-source projects and industrial standards with the help of various evolution methods of fuzzy. Many questions can be answered by using these fuzzification methods like if it covers the full branch coverage if test inputs are generated. Also, to check if a mutant vector is killed in the implementation of execution values between mutant values and final values[2].

Figure 1 depicts the multiple dimensions of regression testing of retesting software which is extended by

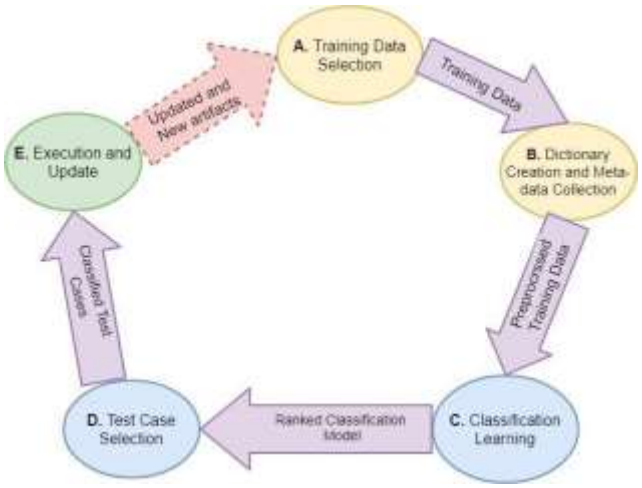

Figure 1. Machine learning-based test case selection for quality assurance.

new features during software development and the time taken in retesting the complete program may not be feasible with the amount of time and cost required so to overcome, a set of all test cases is executed for regression testing. For example, a test case execution prioritization. There are many methods for the selection of test cases based on the domain expertise of Testing Engineers or the subject experts. To automate the selection of test cases in order to avoid the chances of missing important test cases, work shows the selection of test cases with the help of the classification model. The work is presenting the effectual outcome and performance-aware results and these can be used in the advanced predictions associated with the bugs and software quality management. Also, metaheuristics are more widely used to solve problems of combinatory optimization, though of course, any topic may be recast in this manner, for example solving Boolean equations. In regular the commonly used metaheuristic approaches include: simulated annealing, threshold accepting, iterated local search, ant colony optimization, memetic algorithm, variable neighborhood, genetic algorithm, tabu search, scatter search, laguna, glover, and many others. The majority of frequently used metaheuristics are paying attention to combinatorial evolutionary as well as optimization problems but evidently, they can hold any difficulty that can be recast in that form and it can be used just like solving problems using Boolean equations.

2. Literature review

Various researchers, scholars, and practitioners are working in enormous efforts towards operating with suggestive remarks on the study of the same domain, yet there is tremendous scope for progress as the methods,

techniques, and paradigms need to be systematically analyzed. A large number of researchers are working on various domains with suggestions to maximize and assess the quality.

Takenen et al.[3] suggested a component-based model which is doing Asset-Centric-based assessment and evaluation of all areas of risks and their relation with metric-based analysis of risk in software components.

Odzaly et al.[4] emphasize the creation of software programs and effort stresses incorporate data collection from a working project context to efficiently interpret data collection. In this, the combination of data assimilated from live projects are used to highlight effectually by software agents for agile-based risk assessment analytics on data.

Palma et al.[5] have deployed an infrastructure as code (IaC) by offering automatic and configuration file infrastructure and tools to interact.

Zavvar et al.[6] emphasize comparing and projected results on the basis of the soft computing approach for optimization of results classifying risk related to the software development phase threats. The method suggested is a soft calculation approach to optimization of outcomes using support vector machine (SVM) and examines and region parameters of the curve called area under curve (AUC) and the assignment precisity rate (CAR). K-Means and findings can be correlated with self-organizing maps (SOM) and classification accuracy rate (CAR).

Ghobadi and Mathiassen[7] presented a representation for the removal of risks in agile development using four agile projects and two software companies researched risk analysis and knowledge sharing in software development for certain issues in software development, usefulness and created a resolution strategy to minimize the risks. Outcomes raised the knowledge-sharing risk management for different project performances.

Lu et al.[8] introduced the integration of decision models and risk avoidance by numerous tech firms in agile development. The usage of simulated annealing and GA methods for automated fault control for external ventures were integrated through nature-inspired soft computing approaches. The work assesses the output conditions of the GA, SA, and SAGA empirically and concludes that SAGA is as successful as any other conventional solution as its performance when compared to conventional methods.

Méndez Fernández et al.[9] had given viewpoint of evidentiary software risk control with an assessment of data of 228 organizations via the creation of the probabilistic network strengthens the method of requirements engineering (RE). For evaluation and outcomes, the method of testing datasets in tablets for validation of final results.

Rauter et al.[10] developed the asset-oriented risk evaluation of software components. The manuscript assesses the different aspects of threats and their relation with the metric analysis. In this project, the component-based paradigm is implemented to define the belief architecture for better protection and overall credibility of the software development phase.

Elzamely and Hussin[11] recommends and explains the usage of software-driven modeling methods based on fuzzy regression for risk management. The work used various risk factors correlated with the creation of the software project, which can be easily reversed with a rogue, multi-regressive strategy.

Krishna and Subrahmanyam[12] presented an approach and outlined a principal component analysis and Halstead technique (PCAHT) to software risk management. The study focuses on the estimation and detection of software threats utilizing Halstead, to ensure the availability of higher efficiency and optimum productivity for internal variables influential in software. Job for the association of culture, organization, technologies, and citizens is divided into four sections: the association of environment, organization, technology, and people and more phases will involve each phase estimate, inherent risk evaluation, and PCA incorporation.

Purandare[13] suggested the framework for software defects management and risk control system utilizing the entropy method focuses on the estimation of core risk factors in the creation of software projects and on the application of Shannon entropy for risk analysis. In this work which is related to each software development process, the danger in the particular phase is likely to occur. The work illustrates entropy's effective and extremely efficient usage of software creation risk control.

Choetkiertikul et al.[14] incorporates a delay and risk prediction network-dependent classification methodology for software projects that also provides the foundations for the design architectures of risk assessments. The paper tests the proposed solution to many parameters like accuracy, precision, retrieval, f-measurement, the region under the curve, ROC, to determine the coherence of the algorithm to calculate the consistency of the algorithm. The work includes a forecast for activities and components in software projects that raise the likelihood of slowdown and browbeaten.

Elzamely et al.[15] uses the linear step-wise discriminate analysis (LSDA) program risk prediction method and associated risks that could impact the success of software projects explicitly or indirectly. The main purpose and objective are to incorporate and adapt the LSDA method to forecast and further classification in the program danger groups as tiny, high, and medium. Risk assessment methods and strategies are often stressed and utilized to analyze the completeness of the projected strategy.

Mohanty and Ravi[16] implemented predictive analytics and estimation of program flaws in the work of machine learning and classification approach for categorizing device faults and the final predictive analysis the classification method would be used. Methods and approaches included in the exercise for the categorization of software defects.

Singh et al.[17] gave the idea of software defect prediction based on ensemble technique in association with random forest algorithm to forecast program defects. The effectual results are observed when compared to conventional methods gives efficient productivity of 6%. For test analytics and prediction mining, the benchmark data collection is taken from PC4 for mining of risks, and research analytics are underlined to analyze the integrity of the projected approach.

Ali et al.[18] suggested the work in the parallel cloud setting environment with a parallel setup on program defect prediction and metric selection and collection. In the framework, two numerous hybrids of parallel wrappers focused on the artificial neural relations network (ANN) were established using relay filters and created fisher and maximum relevance filters. Evaluations are done for actual data sets with all concurrent implementations of defect-prone applications and the experimentation depicts that parallel hybrid framework maintenance gets a better computational speedup with high prediction accuracy of defects and software metrics compared to independent filter indicating the proposed parallel hybrid system manages pace with a large prediction precision and less software associated to identify binding methods.

Osman et al.[19] developed the automated selection of features with the enhancement of the software system's prediction of error and overall accuracy of the system. In a Java-based framework, analysis of ridge, lasso, and the elastic net is implemented by linear and Poisson regressions the bug prediction approaches on the study of crustaceans, lasso, and elastic net carried out in linear and Poisson regression as bug estimation reaches the Java framework. Here comparison of Poisson regression and linear regression, mean squared error generated is done with the application of regular methods. The work revealed the regularization increases and improves the efficiency and decreases up to 50 percent of the root average squared error, thus increasing prediction stability.

Maddipati et al.[20] combined an inference system for defect prediction using the adaptive neuro-fuzzy inference system (ANFIS) using subtractive clustering trained using the hybrid learning rule. The classified AUC values from imbalanced datasets are compared with ANFIS and receiver operating characteristics (ROC) are cost-sensitive to create graphs in the results section.

Mittal et al.[21] presented a clustering approach which is a changed and updated single-pass clustering based on variable threshold method.

Xuan et al.[22] illustrates the device fault forecasts and analytics using the data reduction-based software methodology. We derive characteristics from historic faulty data sets and create a statistical replica for the current data set to assess how instance choices and function selections can be implemented.

Neufelder[23] presented the software reliability can be practically used by software developers in all aspects of all function sizes.

Van Deursen et al.[24] applied testing techniques called JPACMAN written in small coding of Java separated into different parts like getting familiar into environment and tools and follow after earning the points.

Leicht et al.[25] presented an approach for IT-enabled models manual testing is not practicably possible to implement. Therefore, the method of crowd testing utilizing crowdsourcing to apply testing by employing exterior throng of Internet users, workers, or engaging clientele.

Mittal et al.[26] described the rising of big data technologies and cloud computing environments and comparison of spark and Hadoop and resulting of limitations of Hadoop over spark can risk can be minimized by applying the same environments used by any kind of professionals.

Abdelrafe[27] depicted the fuzzy regression-based modeling methods with 49 risk factors related to software project development for risk analysis.

Elzamly et al.[28] uses an approach for the risk prediction called linear stepwise discriminate analysis for the identification of threats that affects the execution of software project to combine it for early prediction of software risks classified as low, high, and medium-class approaches for the risk control.

Yucalar et al.[29] suggested a manual risk forecasting method by using automated predictors to aim at error-prone components so that engineers can focus on the faulty component.

3. Proposed algorithm integrated with gradient integrated approach and statistical analysis

A brief introduction of all the statistical methods is covered below which draws attention to the variety of statistical and machine learning methods used for quality and defect reduction. Software testing and analyzing using the regression techniques authenticate the preceding functionalities whenever there is change or modification in software or new functionalities are added. Previous surveys suggested that the cost reduction metrics which are mostly used are quality, cost and failure detection along with time[30]. This can be observed that software quality was done for the obsolete or agile projects only. Statistical methods for software quality prediction can be done using regression analysis methods in which the values of any unidentified variable are predicted which are dependent on one or more known variables values. The basic introduction of regression approaches and machine learning are explained below:

- Regression-based approach

The types of regression like linear and logistic regression. The difference between both types of regression methods is given below.

- Linear regression & logistic regression

The former is a statistical technique that creates a linear straight-line relationship among different variables used when errors are dispersed over an extensive variety of classes. Linear regressions of two types: (a) univariate linear regression, (b) multivariate linear regression. The latter is used to forecast the result of dependent variables on independent variables and a dependent variable can acquire two values, therefore a

class of dependent variables containing errors is classified into two categories in which the first category contains no bugs and others have at least one bug. Logistic regression can again be of two types: (a) univariate logistic regression, (b) multivariate logistic regression.

- Machine learning methods

These methods are used in addition to the statistical method various machine learning techniques can be applied to forecast the exactness velocity in software quality fault minimization. **Figure 2** represents the merging of statistical and machine learning techniques

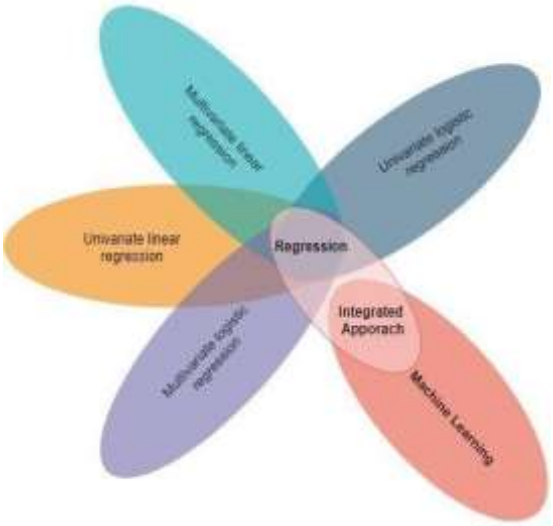

Figure 2. The amalgamation and integration of regression and machine learning techniques.

Stochastic gradient-based approach and regression integrated analysis is a slope plunge streamlining technique for limiting an objective capacity that is made as a whole out of differentiable capacities and makes clusters of the same or different data[31].

As the calculation clears through the arrangement set, it plays out the above upgrade for each readiness test. A couple of ignores can be made the planning set until the calculation blends[32,33]. If this is done, the data can be reworked for each go to maintain a strategic distance from cycles. Average utilization may use an adaptable learning rate so the calculation joins together[34].

Inclination descent is a first-orchestrate improvement approach and is used for statistical evaluations. To find a close-by least of a capacity using slope plunge, one gains ground comparing to the negative of the inclination (or of the evaluated angle) of the capacity at the present point. In case rather one gains ground comparing to the positive of the slope, one approaches a close by generally outrageous of that capacity; the procedure is then known as inclination rising[35–37].

Figure 3 is depicting the flow of work and the steps mentioned are an iterative process with an assortment of phases in which existing paradigms and technologies for quality of service (QoS) factors for the software services are studied and considered.

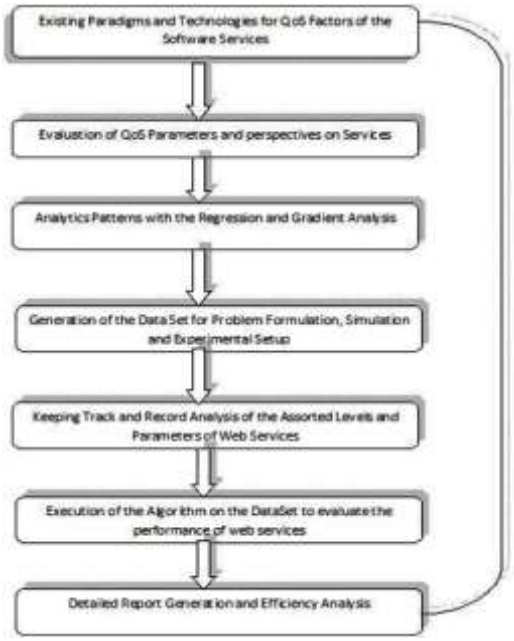

Figure 3. Algorithmic flow of work.

4. Generation and adoption of the dataset with the benchmark properties

i. Fragmentation of the DataSet into further subsets
   a. training data
   b. validation data
   c. testing data

ii. loop block initiates for each category:
   a. add a label to output
   b. label class allocation
   c. evaluation of the key factors

iii. end loop block

iv. loop block for the epochs to variable levels
   a. loop block activated and initiated to the max. size of training data
      i. if (iteration < or > 1)
         1) create initial inference K-means clustering
         2) add a label to each data point
         3) for each data point from the training set
      ii. measure centroid and standard deviation
         end if
      iii. if success=true
         1) normalize standard deviation
         2) evaluate and analyze gradient
         3) adapt X
      iv. end if
   b. end for
   c. update step size
   d. evaluate and analyze Root Mean Square Error (RMSE)
   e. measure performance
   f. evaluate and analyze fuzzy integrated weights
   g. initialize q: = 0

h. loop block to the max. size of Input Data
i. increment wi × xi with q
1) end loop block
j. measure error factor
k. learn weights

v. end loop block

vi. for each data in Dataset
a. evaluate and analyze predicted using trained weights
b. evaluate performance using error from the actual value
c. store error

vii. end for

viii. plot performance

The approaches of regression and statistical analytics are required so that the predictions on the test datasets can be done effectively. For these integrations, the usage of advanced benchmark datasets is required and widely integrated into the software products. Software quality aspects for the projects and test management are as follows:

Availability	- integer
Best practice	- integer
Success	- integer
Documentation	- Floating Point
Compliances	- integer
Response-time	- Floating Point
Throughput	- Floating Point
Reliability	- Floating Point
Latency	- Floating Point
Web Service Name	- String
Software Service Classification	- integer
URL/URI of the Software Module	- String

The technique and algorithmic approach can be integrated with assorted metaheuristic approaches for higher accuracy and global optima for futuristic approaches. Some of the metaheuristics are commonly used or applied for problems where there are no reasonable problem-specific algorithm or heuristic thumb rules for which the problem is not sufficient/impractical or were applying such an approach is not feasible.

5. Results and analysis

The work proposes a qualified revision of diverse quality prediction methods for error minimization based on any project and its related data sets. Then there are factors that are evaluated on the basis of QoS parameters and perspective on services. After that, the patterns are analyzed the regression and gradient analysis methods. The next step will be the generation of data sets so that problems can be formulated and along with that simulation and experimental setup can be done for keeping track and record analysis of the assorted levels and parameters of Web Services. The next step will be the execution of the algorithm on the dataset to evaluate the performance of web services. In the last step of the algorithm, there will be the detailed report generation and efficiency analysis of all the steps with the visualization of data through the graphs. Following is the analytics of the simulation done on the datasets and thereby the evaluation patterns are presented as shown in **Table 1**.

Table 1. Performance analytics on stability points.

Iterations	Execution time (in Microseconds)	Stability point (error evaluation)
10	1.21	8
30	1.29	23
60	1.20	24
70	1.42	25
100	1.39	74
120	1.32	75

The evaluation of training and testing with the validation is required to analyze and reduce the error factors. The training error is evaluated to analyze the overall patterns of the datasets under evaluations and presented in **Figures 4** and **5**.

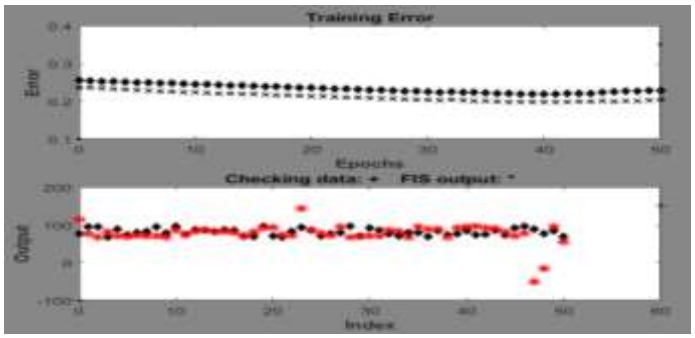

Figure 4. Performance analytics output on stability diverse points.

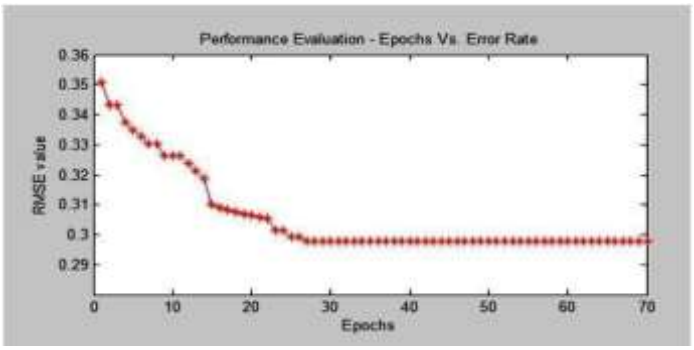

Figure 5. Epochs and performance evaluation.

The performance levels are evaluated with the epochs and found the effective outcomes shown in **Figure 6**.

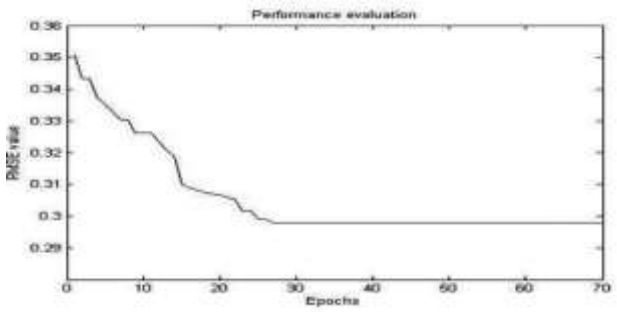

Figure 6. RMSE and performance analytics.

The stability points and their integration with the dataset are used for the cumulative performance that underlines the effectiveness in the multiple dimensions and is presented in **Figure 7**.

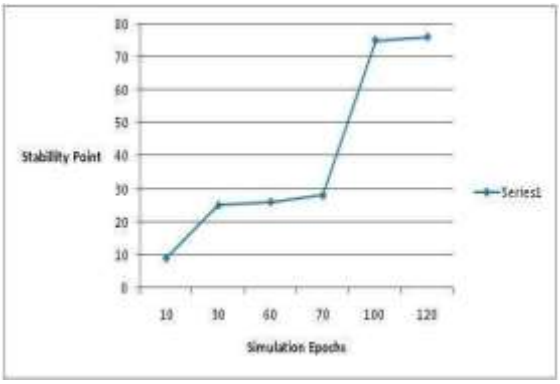

Figure 7. Stability points and simulation epochs.

The scatter plot as in **Figure 8** is presenting the cumulative evaluation of the presented approach. The cost factor is reducing in the iterations and giving the effectual outcomes.

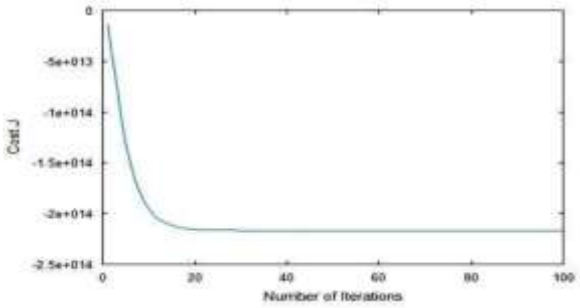

Figure 8. Scatter Plot for cumulative evaluation.

The plot in **Figure 9** depicts the training data and the associated linear regression fit. The line is presenting the performance levels and there is huge consistency.

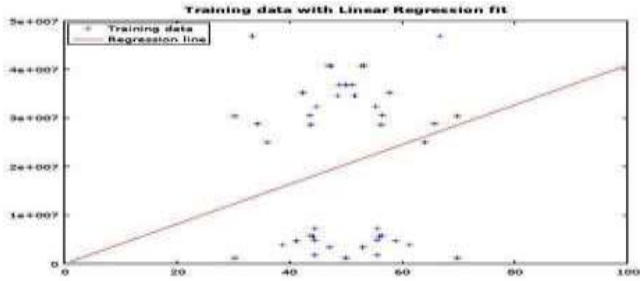

Figure 9. Training data with linear regression.

6. Conclusion and future scope

The domain of software quality is quite prominent whereby the software suites are required to be rigorously tested. With the expansion of tremendous programming suites and modules, there is a need to assess the combined quality and in the general execution of recently created modules utilizing propelled calculations and approaches including AI, factual investigation, profound learning, and numerous others. In this work, the product quality affirmation with the datasets assessment utilizing factual examination and relapse coordinated methodology is introduced. In this, the programmer can use the datasets to predict and classify the error-prone areas in the project. The work is displaying the effectual result and execution aware outcomes and these can be utilized in the propelled expectations relationship with the bugs and programming quality administration. In this, the programmer can use the datasets to predict and classify the error-prone areas in the project. The right number of errors of faults prediction is the prime goal of software quality analysis. The application of statistical and machine learning usage in error forecasting requires a colossal quantity of statistics and

moreover to analyze this massive data is essential with the help of a better forecast model. There are many approaches to check on software quality but using the statistical regression approach is giving better results as compared to the other analytical approaches. This is done to reduce the cost when statistical techniques were applied manually and were cumbersome and time-consuming and attention is to reduce cost and bring efficient results**.** Earlier the software quality was checked only for the developed programs to check that for corrective and maintenance activities. Present researchers and people from the research community or system analysts are making use of forecast models in order to categorize error levels is actually defective or not as much broken down. Earlier statistical methods were deployed for doing the process but present work requires a more dependable approach so that the modeling for the prediction of software quality can be done. The application of statistical and machine learning usage in error forecasting requires a colossal quantity of statistics and analyzing this massive data is essential with the help of a better forecast model.

Author contributions

Conceptualization, AA; methodology, AA; software, AA; validation, AA, VB and AA; formal analysis, SSB; investigation, SSB resources, RK; data curation, RK; writing—original draft preparation, AA; writing—review and editing, AA; visualization, PK; supervision, PK; project administration, PK; funding acquisition, AA. All authors have read and agreed to the published version of the manuscript.

Conflict of interest

The authors declare no conflict of interest.

References

1. Lewis WE, Dobbs D, Veerapillai G. Software Testing and Continuous Quality Improvement. Auerbach Publications; 2017. doi: 10.1201/9781439834367
2. Goyal LM, Mittal M, Sethi JK. Fuzzy model generation using Subtractive and Fuzzy C-Means clustering. CSI Transactions on ICT. 2016; 4(2-4): 129-133. doi: 10.1007/s40012-016-0090-3
3. Takanen A, Demott JD, Miller C, Kettunen A. Fuzzing for software security testing and quality assurance. Artech House. 2018.
4. Odzaly EE, Greer D, Stewart D. Agile risk management using software agents. Journal of Ambient Intelligence and Humanized Computing. 2017; 9(3): 823-841. doi: 10.1007/s12652-017-0488-2
5. Palma SD, Di Nucci D, Palomba F, Tamburri DA. Towards a Catalogue of Software Quality Metrics for Infrastructure Code. Journal of Systems and Software. 2020; 170: 110726.
6. Zavvar M, Yavari A, Mirhassanni SM, et al. Classification of risk in software development projects using support vector machine. Journal of Telecommunication, Electronic and Computer Engineering (JTEC). 2017; 9(1): 1-5.
7. Ghobadi S, Mathiassen L. Risks to Effective Knowledge Sharing in Agile Software Teams: A Model for Assessing and Mitigating Risks. Information Systems Journal. 2016; 27(6): 699-731. doi: 10.1111/isj.12117
8. Lu F, Bi H, Huang M, et al. Simulated Annealing Genetic Algorithm Based Schedule Risk Management of IT Outsourcing Project. Mathematical Problems in Engineering. 2017; 2017: 1-17. doi: 10.1155/2017/6916575
9. Méndez Fernández D, Tießler M, Kalinowski M, et al. On Evidence-based Risk Management in Requirements Engineering. arXiv. 2017.
10. Rauter T, Höller A, Kajtazovic N, Kreiner C. Asset-Centric Security Risk Assessment of Software Components. In: Workshop on MILS: Architecture and Assurance for Secure Systems; 2016.
11. Elzamly A, Hussin B. Managing Software Project Risks (Analysis Phase) with Proposed Fuzzy Regression Analysis Modelling Techniques with Fuzzy Concepts. Journal of Computing and Information Technology. 2014; 22(2): 131. doi: 10.2498/cit.1002324
12. Krishna BC, Subrahmanyam K. A Decision Support System for Assessing risk using Halstead approach and Principal Component Analysis. Journal of Chemical and Pharmaceutical Sciences. 2016; 9(4): 3383–3387.
13. Purandare P. An entropy-based approach for risk factor analysis in a software development project. International Journal of Applied Engineering Research. 2016; 11(4): 2258-2262.
14. Choetkiertikul M, Dam HK, Tran T, et al. Predicting Delays in Software Projects Using Networked Classification (T). In: 2015 30th IEEE/ACM International Conference on Automated Software Engineering (ASE). Published online November 2015. doi: 10.1109/ase.2015.55
15. Elzamly A, Hussin B, Abu-Naser SS, Doheir M. Predicting Software Analysis Process Risks Using Linear Stepwise Discriminant Analysis: Statistical Methods.Int. J. Adv. Inf. Sci. Technol. 2015; 38: 108–115.

16. Mohanty R, Ravi V. Machine Learning Techniques to Predict Software Defect. Artificial Intelligence.: 1473-1487. doi: 10.4018/978-1-5225-1759-7.ch059
17. Singh R, Raja R, Chopra J. Software Defect Prediction Using Averaging Likelihood Ensemble Technique. International Journal. 2017; 4: 213-223.
18. Ali MM, Huda S, Abawajy J, et al. A parallel framework for software defect detection and metric selection on cloud computing. Cluster Computing. 2017; 20(3): 2267-2281. doi: 10.1007/s10586-017-0892-6
19. Osman H, Ghafari M, Nierstrasz O. Automatic feature selection by regularization to improve bug prediction accuracy. 2017 IEEE Workshop on Machine Learning Techniques for Software Quality Evaluation (MaLTeSQuE). Published online February 21, 2017. doi: 10.1109/maltesque.2017.7882013
20. Maddipati SS, Pradeepini G, Yesubabu A. Software Defect Prediction using Adaptive Neuro Fuzzy Inference System. International Journal of Applied Engineering Research. 2018; 13(1): 394-397.
21. Mittal M, Sharma RK, Singh VP. Modified single pass clustering with variable threshold approach. International Journal of Innovative Computing Information and Control. 2015; 11(1): 375-386.
22. Xuan J, Jiang H, Hu Y, et al. Towards Effective Bug Triage with Software Data Reduction Techniques. IEEE Transactions on Knowledge and Data Engineering. 2015; 27(1): 264-280. doi: 10.1109/tkde.2014.2324590
23. Neufelder AM. Ensuring Software Reliability. Published online October 8, 2018. doi: 10.1201/9781315217758
24. Van Deursen A, Aniche M, Boone C, et al. Software Quality and Testing. Delft University of Technology. 2019.
25. Leicht N, Blohm I, Leimeister JM. Leveraging the Power of the Crowd for Software Testing. IEEE Software. 2017; 34(2): 62-69. doi: 10.1109/ms.2017.37
26. Mittal M, Balas VE, Goyal LM, et al. Big Data Processing Using Spark in Cloud. Springer Singapore; 2019. doi: 10.1007/978-981-13-0550-4
27. Abdelrafe MS. Managing Software Project Risks Using Stepwise and Fuzzy Regression Analysis Modeling Techniques. 2016.
28. Elzamly, Hussin B, Naser SSA, Doheir M. Predicting Software Analysis Process Risks Using Linear Stepwise Discriminant Analysis: Statistical Methods. Int. J. Adv. Inf. Sci. Technol. 2015; 38: 108–115.
29. Yucalar F, Ozcift A, Borandag E, et al. Multiple-classifiers in software quality engineering: Combining predictors to improve software fault prediction ability. Engineering Science and Technology, an International Journal. 2020; 23(4): 938-950. doi: 10.1016/j.jestch.2019.10.005
30. Rothermel G. Improving regression testing in continuous integration development environments (keynote). Proceedings of the 9th ACM SIGSOFT International Workshop on Automating TEST Case Design, Selection, and Evaluation. Published online November 5, 2018. doi: 10.1145/3278186.3281454
31. Podgorny IA, Cessna J, Gielow CC, Cannon M. U.S. Patent No. 10,162,734. Washington, DC: U.S. Patent and Trademark Office. 2018.
32. Mittal M, Sharma RK, Singh VP. Validation of k-means and threshold based clustering method. International Journal of Advancements in Technology. 2014; 5(2): 153-160.
33. Podgorny IA, Cessna J, Gielow CC, Cannon M. US Patent 10162734 Method and system for crowdsourcing software quality testing and error detection in a tax return preparation system. U.S. Patent No. 10,162,734, 25 December 2018.
34. Felderer M, Fourneret E. A systematic classification of security regression testing approaches. International Journal on Software Tools for Technology Transfer. 2015; 17(3): 305-319. doi: 10.1007/s10009-015-0365-2
35. Aggarwal A, Dhindsa KS, Suri PK. A Pragmatic Assessment of Approaches and Paradigms in Software Risk Management Frameworks. International Journal of Natural Computing Research. 2020; 9(1): 13-26. doi: 10.4018/ijncr.2020010102
36. Malik M, Prabha C, Soni P, et al. Machine Learning-Based Automatic Litter Detection and Classification Using Neural Networks in Smart Cities. International Journal on Semantic Web and Information Systems (IJSWIS). 2023; 19(1): 1-20. doi: 10.4018/IJSWIS.324105
37. Paliwal M, Soni P, Chauhan S. Digit Recognition using the Artificial Neural Network. In: 2023 International Conference on Advancement in Computation & Computer Technologies (InCACCT); 5–6 May 2023; Gharuan, India. pp. 145-149. doi: 10.1109/InCACCT57535.2023.10141703