Data Flow Bug Detection and Analysis in Business Process

M.Thilagarani
Department of Computer Science and EngineeringVelalar College of Engineering and Technology Thindal, Erode – 638012
Ajitha V
Department of Computer Science and Engineering Velalar College of Engineering and Technology Thindal, Erode – 638012
Arun R P
Department of Computer Science and Engineering Velalar College of Engineering and Technology Thindal, Erode – 638012
Elakiya Sri G
Department of Computer Science and Engineering Velalar College of Engineering and Technology Thindal, Erode - 638012

Abstract-Software defect prediction helps businesses succeed by giving software team's useful results. Existing research, however, hasn't yet shown how to anticipate how many flaws a new product release will contain. 4 different datasets with 228 instances each were taken from the Cagle repository for the experiment. It pinpoints the modules that are prone to errors and demand thorough testing. In this manner, the testing resources can be used effectively while still adhering to the limitations. Although while SDP is a great tool for testing, it's not always simple to identify the faulty modules. With a correlation coefficient of 0.98, the average defect velocity and the total number of defects have a very significant positiveassociation. It is thus proven that this method can offer a guide for programmer testing in order to increase the efficiency of software development activities.

Keywords --- RMSE, MAE, SMO Algorithm

I. INTRODUCTION

The quantity of flaws is a crucial indicator of software quality that is frequently utilized in business. Unfortunately, it might be challenging to estimate defect density accurately. Sampling methods typically make the assumption that the discovered faults are a representative sample of all the faults already existent, which leads to erroneous estimations. Tools for measuring software testcoverage can quickly and precisely determine how often the software has been used. Software defect prediction, which provides in advance knowledge of the amount of flaws that are anticipated to be discovered in a new programmer, aids in ensuring thattesting and debugging continue to proceed quickly. Existing research, however, hasn't yet shown how to use derived variables to anticipate how many bugs will be in a forthcoming release. The ability to forecast software flaws would help software teams plan software testing and keep software standards up to date. Predicting the quantity of flaws in a new product release is crucial for ensuring cost- effectiveness. Such a prediction technique ought to concentrate on directing software testing efforts by foretelling the potential amount of software flaws ina new product release before testing gets started.

II. LITERATURE REVIEW

1. LEARNING TO DETECT COMMUNITY SMELLS IN OPEN-SOURCE SOFTWAREPROJECTS
Community odors are signs of social and organizational problems in the software development
Community, which frequently result in increased project costs. The rapid growth of software projects, which involves numerous clients and stakeholders, is one of its fundamental characteristics. Several scholars and practitioners demonstrated how crucial organizational aspect evolution is to preventing its degradation and, as a result, software project failures.

2. INTEGRATING PROCESS MANAGEMENTAND EVENT PROCESSING IN SMART FACTORIES
New frameworks and system designs for intelligent production systems have emerged as a result of Industry 4.0's new ideas for a greater digitization of industrial industries. Although the majority of approaches include a service-based architecture for selective integration with enterprise systems. BPM and IOT together have the potential to be beneficial for both research domains, albeit being in their early stages. Several use cases including process modelling, automation, and mining for a smart factory model serve to show the advantages of usingBPM technology to analyses and adjust discrete production processes in the IIoT.

1. IDENTIFYING A MINIMUM SEQUENCE OFHIGH-LEVEL CHANGES BETWEEN WORKFLOWS

There are several workflow variants as a result of adaptive workflow management systems' ability to adjust workflows during both the modelling and runtime phases. A basic but crucial problem is figuring out the smallest sequence of high-level modifications that should occur between two workflows. The most recent method searches for thebest answer using digital logic, however it does not scale well and may have problems when advanced workflow patterns (like loops) are included. Thisinitially suggest a basic solution to this issue, one thatapplies all legitimate modifications to one process upuntil the other workflow is located.

2. PREDICTING PRIORITY OF REPORTEDBUGS BY MULTI-FACTOR ANALYSIS

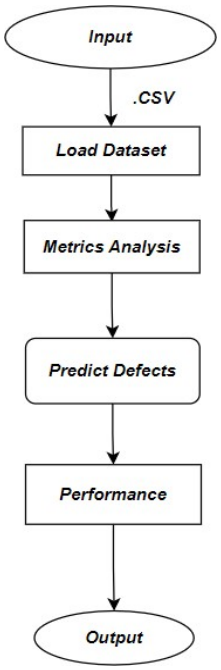
There are lots of bugs. Developers frequently allow users to report issues they find in software using a bug-tracking tool like Bugzilla in an effort to increasesoftware quality. Input from users would include, among other things, a description of the defect, the component it affects, and its severity. Due to a lack of resources, bug reports would be looked into according to their priority ratings. Yet, this prioritization procedure is a manual one. In this research, machine learning-based automated method that would suggest a priority level based on data from bug reports. This method takes into account a number of variables, including temporal, textual, author, related report, severity, and product, that may have an impact on a bug report's priority level. This performance can surpass baseline approaches in terms of average F-measure by a relative improvement of 58.61%, according to tests on more than 100,000 bug reports from Eclipse.

III. METHODSPROPOSED SYSTEM

The findings of a prediction study may be impacted by class imbalance, a well-known problem in machine learning investigations. A properly cleaned and pre-processed dataset can aid in the prediction of errors early in the Tri Model method, freeing the software team to focus on producing
Better outcomes. Hence, these actions might promotebetter software quality. Recent defect prediction research have called into question the validity of the datasets used in defect prediction and have also demonstrated the necessity of properly pre- processing such datasets. The development of the proposed modelling approach is thoroughly described. The basis for the modelling method is the Rayleigh distribution curve, which depicts the progression in defect frequency over time in a project

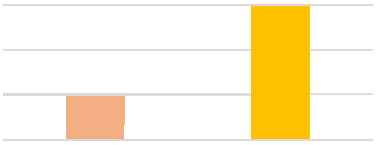
IV.FLOW CHARTDATA PRE-PROCESSING

The data preprocessing step, which tackles class imbalance and data cleanness, is the first stage of the suggested methodology. The class imbalance is a well-known issue in machine learning studies thatmight have an impact on a prediction study's results.Datasets that have been properly cleaned and pre- processed can make it easier to predict problems

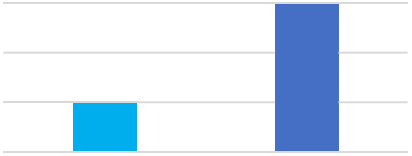


Yearly in the Tri Model method, allowing the software team to focus on producing better outcomes. As a result, such efforts might promote better software quality. The validity of the datasets used for defect prediction has recently come under scrutiny, and studies on the subject have also demonstrated the need for sufficient pre-processing of such datasets.

V.PROPOSED MODELING TECHNIQUE



MAE PREDICTION



DEFECT PREDICTIONSMO ALGORITHM

For a support vector machine to be trained, a very large quadratic programming (QP) optimization issue must be solved. SMO breaks down this significant QP challenge into a series of minor QP problems.

$$min_\alpha \ \frac{1}{2}\sum_i \sum_j \alpha_i \alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j - \sum \alpha_i$$

$$\text{s.t.} \sum_i \alpha_i y_i = 0, \alpha_i \in [0, \mathbf{C}] \quad \text{(equality constraint)}$$

EQUATION IN SMO ALGORITHM

The inner loop that would have required a time-consuming numerical QP optimization is not requiredin order to handle these minor QP concerns analytically. Due to the fact that memory usage growslinearly with training set size, SMO can handle very large training sets. Due to the avoidance of matrix computing, SMO scales between linear and quadratic
in the training set size for various test problems, as opposed to the linear and cubic scaling of the conventional chunking SVM algorithm. Linear SVMs and sparse data sets benefit from SMO's speedbecause SVM assessment consumes a significant percentage of its computing time. For sparse data setsobserved in the real world, SMO can be a thousand times faster than the chunking method.

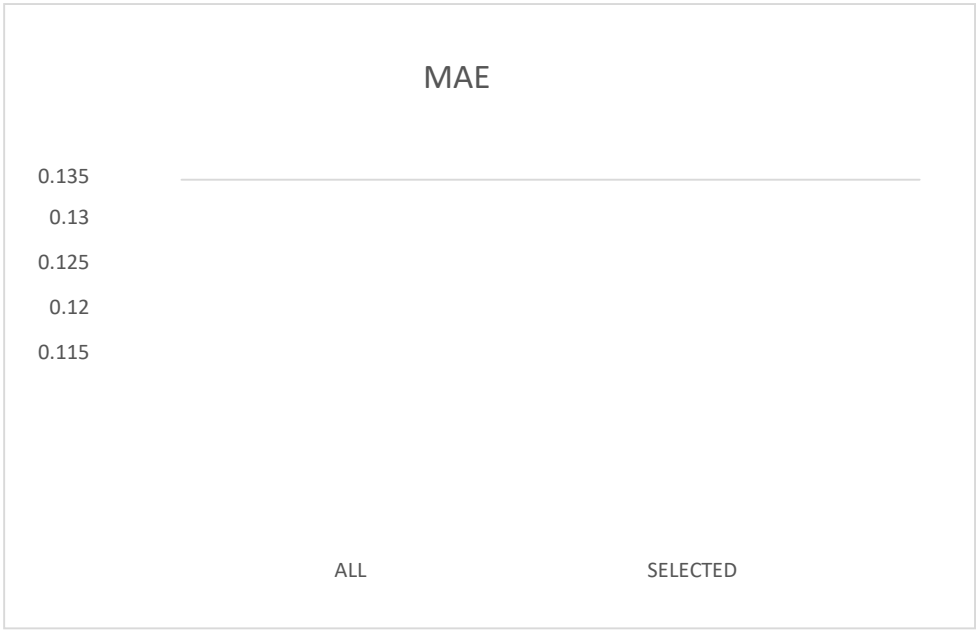
VI.RESULT

The experiment on three different datasets from the Gaggle repository showed that the proposed framework can reliably forecast the amount of software flaws. These results show that the suggested structure is capable of providing the strategy forprogrammer testing to improve the viability of programming improvement exercises.
The framework can be used with the Tri Model method to identify the modules that require extensivetesting and are prone to bugs. By doing this, testing resources are employed efficiently while staying within budget. The suggested system can assist programming teams in achieving more advanced outcomes and advancing higher programming quality

VII.CONCLUSION AND FUTURE WORKS

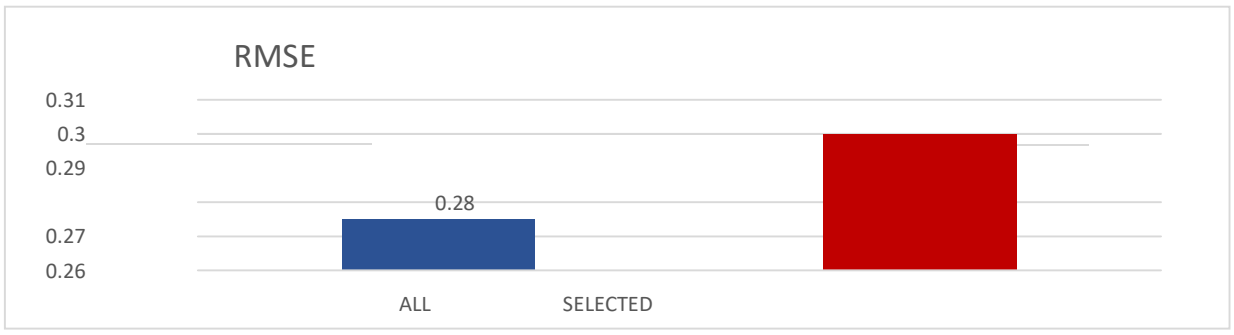
Prediction of software defects brings up a variety of problems that must be solved. In this paper, a Tri Model technique was presented to forecast the quantity of errors in a future software product using predictor variables. Software managers may therefore pay attention to how quickly a project moves from one phase to the next over time in order to decrease defects. To be effective for defect prediction, this approach needs to be tested against

That these variables are related to the quantity ofdefect



VIII.VISUALIZATION OF FAULT DETECTION

Although cross-validation sampling outperforms random or leave-one-out sampling for large datasets. It was employed in trials to assure accurate results. Estimates of model performance are used to foretell how effectively a model will function with unseen data. The authors of employed a cross- validation sampling technique to assess their performance. By using separate training and validation datasets, cross-validation reduces the dependence of one on the other; as a result, this technique is known to be nearly unbiased. Yet smaller samples might have more volatility. Furthermore, when working with small samples, the authors of claim that neither cross validation norbootstrapping are reliable However, unstable results from cross-validation can be stabilized by repeating the validation method. The defect is then graphicallydepicted by modelling it as a chart using a few rendering functions after the forecast. Chart library classes were used to construct the chart. Javasupports a number of XY charts and bar charts.



RMSE PREDICTION

The findings of this research. Further research can validate this strategy for estimating the amount of faults in an upcoming product release while alsotaking into account additional predictor variables,using the most recent datasets from any software company. The incorporation of additional variables ormetrics that can help increase the predictive model accuracy could be one topic of future research. Also,adding data from additional sources, such user reviews or production environment logs, may help toenhance the prediction models. Investigating the application of different machine learning algorithms or methodologies for foretelling data flow issues is another possible subject for future research. The suggested method can also be expanded to manage more intricate business procedures involving several systems or outside apps.

REFERENCES

[1] Z.-W. Zhang, X.-Y.Jing, and T.-J. Wang, "Label propagation based semi-supervised learning for software defect prediction," Automated Software Engineering, vol. 24, no. 1, pp. 47–69, 2021

[2] B. F. Manly and J. A. N. Alberto, Multivariate statistical methods: a primer. CRC Press, 2020

[3] C. Tantithamthavorn, S. McIntosh, A. E. Hassan,and K. Matsumoto, "An empirical comparison ofmodel validation techniques for defect prediction models," IEEE Transactions on Software Engineering, vol. 43, no. 1, pp. 1–18, 2020.

[4] Q. Yu, S. Jiang, and Y. Zhang, "The performancestability of defect prediction models with class imbalance: An empirical study," IEICE TRANSACTIONS on Information and Systems, vol.100, no. 2, pp. 265– 272, 2020.

[5] O. Loyola-Gonzalez, M. A. Medina-P ´ erez, J. F. Mart ´ ´ınez-Trinidad, J. A. Carrasco-Ochoa, R. Monroy, and M. Garc´ıa-Borroto, "Pbc4cip: A new contrast pattern-based classifier for class imbalance problems," Knowledge-Based Systems, vol. 115, pp. 100–109, 2020.

[6] U. Riemann, "Benefits and challenges for business process management in the cloud", Int. J.

[7] Organizational Collective Intel., vol. 5, no. 2, pp. 80-

[8] 104, 2015.

[9] W. Song and H.-A. Jacobsen, "Static and dynamic process change", IEEE Trans. Serv. Compute., vol. 11, no. 1, pp. 215-231, Jan./Feb. 2018.

[10] C.Nagarajan and M.Madheswaran - 'Experimental verification and stability state space analysis of CLL-T Series Parallel Resonant Converter' - Journal of ELECTRICAL ENGINEERING, Vol.63 (6), pp.365-372, Dec.2012.

[11] C.Nagarajan and M.Madheswaran - 'Performance Analysis of LCL-T Resonant Converter with Fuzzy/PID Using State Space Analysis'- Springer, Electrical Engineering, Vol.93 (3), pp.167-178, September 2011.

[12] C.Nagarajan and M.Madheswaran - 'Stability Analysis of Series Parallel Resonant Converter with Fuzzy Logic Controller Using State Space Techniques'- Taylor & Francis, Electric Power Components and Systems, Vol.39 (8), pp.780-793, May 2011.

[13] Nagarajan and M.Madheswaran - 'Experimental Study and steady state stability analysis of CLL-T Series Parallel Resonant Converter with Fuzzy controller using State Space Analysis'- Iranian Journal of Electrical & Electronic Engineering, Vol.8 (3), pp.259-267, September 2012.

[14] Nagarajan C., Neelakrishnan G., Akila P., Fathima U., Sneha S. "Performance Analysis and Implementation of 89C51 Controller Based Solar Tracking System with Boost Converter" Journal of VLSI Design Tools & Technology. 2022; 12(2): 34–41p.

[15] C. Nagarajan, G.Neelakrishnan, R. Janani, S.Maithili, G. Ramya "Investigation on Fault Analysis for Power Transformers Using Adaptive Differential Relay" Asian Journal of Electrical Science, Vol.11 No.1, pp: 1-8, 2022.

[16]

[17] J. Koehler and J. Vanhatalo, "Process anti- patterns: How to avoid the common traps of business process modeling", IBM WebSphere Developer Tech. J., vol. 10, no. 2, 2007.

[18] G.Neelakrishnan, K.Anandhakumar, A.Prathap, S.Prakash "Performance Estimation of cascaded h-bridge MLI for HEV using SVPWM" Suraj Punj Journal for Multidisciplinary Research, 2021, Volume 11, Issue 4, pp:750-756

[19] G.Neelakrishnan, S.N.Pruthika, P.T.Shalini, S.Soniya, "Perfromance Investigation of T-Source Inverter fed with Solar Cell" Suraj Punj Journal for Multidisciplinary Research, 2021, Volume 11, Issue 4, pp:744-749