IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Vol.-7, Special Issue-15, May 2019 E-ISSN: 2347-2693

Software Defect Prediction Using Data Mining Techniques

Swathi K[1*], Arun Biradar[2]

[1,2]Department of Computer Science and Engineering, VTU Belagavi, India

DOI: https://doi.org/10.26438/ijcse/v7si15.284287 | Available online at: www.ijcseonline.org

***Abstract*—**The accomplishment of any software framework completely relies upon the exactness of the consequences of the framework and whether it is with no blemishes. Software deformity prediction issues have an incredibly gainful research potential. Software defects are the serious issue in any software industry. Software defects diminish the software quality, increment costing yet it additionally suspends the improvement plan. Software bugs lead to off base and discrepant outcomes. As a result of this, the software ventures run late, are dropped or become untrustworthy after sending. Quality and reliability are the real difficulties looked in a protected software improvement process. There are real software cost overwhelms when a software item with bugs in its different segments is conveyed next to client. The software distribution center is generally utilized as record keeping vault which is for the most part required while including new highlights or fixing bugs. Numerous information mining strategies and dataset store are accessible to foresee the software defects. 'Bug prediction procedure' is a significant part in software building territory for most recent multi decade. Software bugs which identify at beginning period are straightforward and cheap for redressing the software. Software quality can be upgraded by utilizing the bug prediction strategies and the software bug can be decreased whenever connected precisely. Needy and autonomous variable are considered in Software bug prediction. To anticipate deformity dependent on software measurements software prediction model are utilized. Measurements based characterization sort part as faulty and non-inadequate.

***Keywords*—**Software defects, bugs, prediction, quality, reliability

I. INTRODUCTION

It is appropriate to make reference to that Software defects decrease software quality, increment costs just as postpone the advancement plan. Through software defects prediction strategies, a software advancement group can gauge the conceivable bug and their seriousness in the underlying phase of software improvement. Preceding the start of the testing stage, the way toward finding deficient segments in the software is known as Software deformity prediction. A standout amongst the most dynamic zones of research in software designing is software deformity prediction which prompts expanded consumer loyalty, progressively dependable software, decrease in the season of improvement and decline in revise exertion and cost-adequacy. The activity of determining ceaseless or requested qualities for a given information is known as prediction. In this way, so as to achieve software quality and to gain from prior slip-ups, the act of prediction of defects is viewed as amazingly huge.A common defect prediction process is shown in Figure

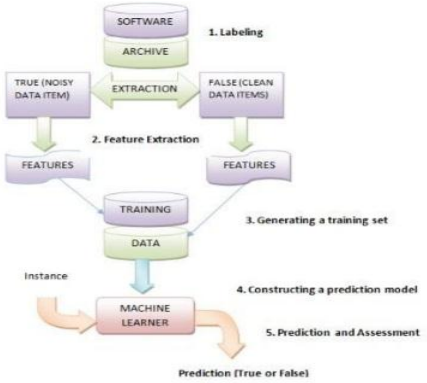


Figure 1. A Common Defect Prediction

II. LITERATURE SURVEY

Kursa notice his paper that Boruta is one of the significant element choice bundles to find the applicable component from an expansive dataset. The Feature choice procedure improves the speed of the calculation, yet in addition expands the exactness of AI calculation. Boruta utilizes a wrapper calculation which utilizes the Random Forest arrangement calculation and depends on the guideline of emphasis and rejects the highlights which are unimportant. The Wrapper calculation is superior to the channel technique in light of the fact that among the given highlights, there is

© 2019, IJCSE All Rights Reserved

no immediate relationship. The Wrapper technique classifier is utilized as a black box, restoring an element positioning. The R bundle Boruta is accessible at http://CRAN.R-project.org/package=Boruta). At the point when the R bundle Bouruta is run, its demonstrates the whole affirmed variable and the rejected variable in a dataset. At the point when the crate plot is drawn Green, Blue and Red box plot speak to a Z-scores of affirmed, negligible or normal rejected characteristic separately. Azeem and Usmani in their paper has called attention to that Software bug store is the fundamental asset for shortcoming inclined modules. There are diverse information mining calculations which are utilized to remove deficiency inclined modules from these stores. The group for Software advancement endeavoured to expand the software quality by diminishing the quantity of defects however much as could reasonably be expected. In this paper distinctive information digging methods are examined for recognizing issue inclined modules just as contrast the information mining calculations with discover the best calculation for imperfection prediction.

Thomas proposed the utilization of measurable point models, for example, Latent Dirichlet Allocation (LDA) to consequently find structure in software vaults since these storehouses contain unstructured and unlabeled content that is fairly hard to dissect with customary systems. This paper tends to the difficulties of applying subject models to software storehouses. Wang et.al. demonstrates that in information mining, include choice system assumes an amazingly fundamental job. Through the utilization of highlight determination technique, software defects and hazard estimation may improve in the order model and the excess and insignificant information gets expelled. They have connected six channels bank rankers in three substantial software ventures. They have constructed an arrangement model in SVM, NB, KNN, LR, MLP students. They have determined the AUC execution metric. They inferred that the informational indexes assume a critical job in assessing the aftereffects of execution of rankers for example highlight choice method. They likewise presumed that in future undertakings, the investigation can be directed on an alternate dataset in various areas like software designing and other application spaces. They likewise referenced that in their future work, they would consider the strident and imbalance information.

Dhiauddinet.al. has proposed the prediction model for defects in framework testing. The fundamental reason for this prediction model is to make a quality pointer of the framework at whatever point any framework is going into a testing stage. They have connected relapse investigation in a chose measurements to foresee the deformity in a testing stage. They have additionally disclosed that to announce the model as fine, the prediction should fall between the greatest and least range. The investigation of the Pvalues must be under 0.5. The estimation of R squared must be over 85%. The Adjusted R squared must be over 85%.

III. PROBLEM FORMULATION

The achievement of any software framework is totally rely upon the precision of the aftereffects of the framework and whether it is with no blemishes, software imperfection prediction issues have an incredibly valuable research potential. Trick, explored the software bugs or defects have rendered significant commitments towards specialized clarifications for software venture disappointment. Mining of software vaults have a few research difficulties to be tended to. New software bug prediction models should be structured, viable software deformity measurements should be incorporated and gave them as contributions to different information digging strategies for removing arranged data so as to conceive the software blames in new software renditions and furthermore progressively created techniques are expected to diminish software cost overruns.The software measurements (for instance, process measurements and item measurements) lie at the center of bug prediction models. The primary goal of any association is to have deformity free software. Truth be told, before discovery of deformity would spare time and cost of the framework. The above articulation obviously portrays that we have to recognize the best AI model for software deformity prediction for which different execution parameters are accessible, for example, precision, mean square mistake, and relationship and R-Squared to contrast and other various models. Information is a significant piece of the framework. One of the greatest difficulties is to gain the privilege dataset and where in to classify the reliant and free factors. The more the information, the more mind boggling will the framework become and greater likelihood of the deformity showing up. Subsequently, it is constantly more secure to expel the immaterial factors from the dataset and decrease the free factor utilizing the element choice method. The unimportant factors have an immaterial effect to recognize the software bug. There are different kinds of highlight choice methods that are accessible to determine the huge and unimportant factors from the dataset. For the examination, the Wrapper and Filter strategy utilized in Feature Selection procedure are taken to locate the basic variable from the openly accessible Promise Repository. An AI strategy can be delegated a managed learning and unsupervised learning. Unsupervised learning takes a shot at a shrouded information and comprises of grouping; affiliation examination, concealed Markov model. The managed learning is utilized when there is a necessity to prepare the model for prediction. It contains relapse and characterization. A large portion of the looks into depend on arrangement methods and just a couple have been finished utilizing relapse systems utilized in an AI model. Research has been completed utilizing the relapse procedure

© 2019, IJCSE All Rights Reserved

and AI model utilized are Linear Regression, Decision Tree, Random Forest, Neural Network, Support Vector Machine and Decision Stump. Execution parameters were connected on these AI models to accomplish the best model to anticipate the software deformity. AI systems, Naïve Bayes, Decision Stump, SVM-Polykernel, and SVM-RBF were utilized by Selvarajet.al. to ascertain the software imperfection utilizing arrangement strategy and inferred that SVM-Polykernel achieve the best execution and the dataset taken was Promise Repository. Yogesh et.al has utilized the NASA dataset for the examination and found that the SVM model offers the best precision out of strategic relapse and the choice tree to foresee the software imperfection. Elishet.al too utilized the NASA dataset and connected eight measurable and AI models to foresee the imperfection inclined module and have announced that the SVM as the most ideal model as contrasted and Logistic Regression (LR), K-Nearest Neighbor (KNN), Multi-Layer Perceptrons (MLP), Radial Basis Function (RBF), Bayesian Belief Networks (BBN), Naive Bayes (NB), Random Forests (RF) and Decision Trees (DT). The greater part of the Researchers have inferred that the Support Vector Machine (SVM) is by a long shot the best model to foresee the software deformity yet some of them have utilized grouping strategy and connected on different datasets. The examination hole was to apply the AI model utilizing relapse system and to check whether in different datasets, it likewise accomplishing similar outcomes.

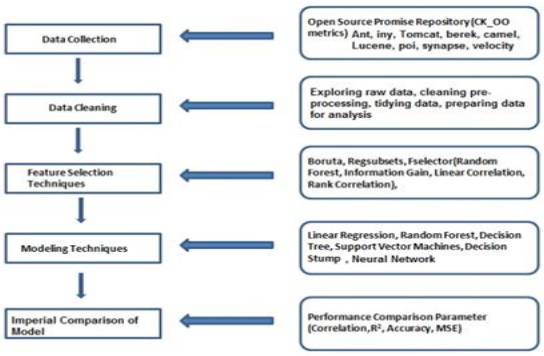


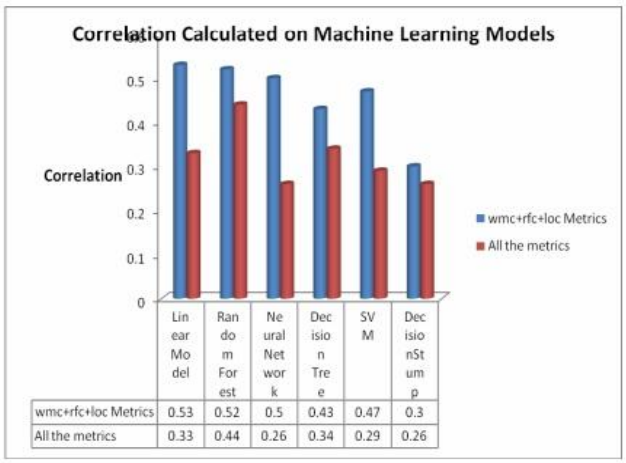
Figure 2. Frame Work For Software Defect Prediction UsingHistorical Databases

IV. RESULTS AND DISCUSSIONS

The comparative analysis was done by using the performance parameters on machine learning models, such as, Linear Regression, Random Forest, Decision Tree, Support Vector Machine, Neural Network and Decision Stump on software modules like Ant, Ivy, Tomcat, Berek, Camel, Lucene, POI, Synapse and Velocity. Two observations were analyzed which are described below:

Feature Selection Analysis

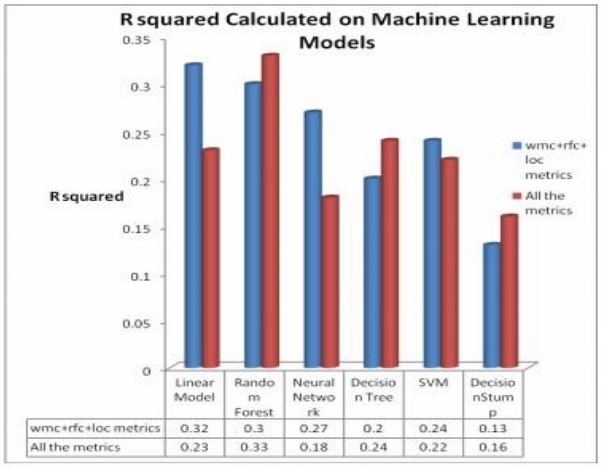
When the modeling technique was applied on the reduced variable, the result was either better or the same. The more the number of variables, the more complex would be the system. Another important factor of using Feature Selection technique is that if the number of variables is higher than optimal, then the Machine Learning Algorithm exhibits a decrease in accuracy.



Correlation Calculated on different Machine Learning Model using all the software metrics and with WMC, RFC and LOC software metrics

Figure 3. Correlation Calculated On Different Machine Learning Model

Above figure indicates that the result was much better when using only the optimal software metrics (WMC, RFC, and LOC). The best correlated machine learning model was with a Linear Regression value as 0.53 and least correlated was with a Decision Stump as 0.3.



R-Squared Calculated on different Machine Learning Models using all the software metrics and with WMC,RFC and LOC Software Metrics

Figure 4. R-Squared Calculated On Different Machine Learning Models

© 2019, IJCSE All Rights Reserved

The Linear Regression has the highest R-Squared value as 0.32 and least as 0.13 when using only optimal software metrics. The above Figure shows that using only the selected software metrics in Machine Learning Model was providing the best result comparatively.

V. CONCLUSION

The prior the imperfection is distinguished the cost included is decreased and the assets are completely used so it turns out to be a lot simpler to amend the deformity amid the underlying stage, Complexity measurements are better indicators of shortcoming potential in contrast with other understood recorded indicators of flaws, self-audit the code has a noteworthy commitment in averting the software imperfection, documents that have enemies of examples will in general have a higher thickness of bugs than others as enemies of examples can build the bugs later on. Enemies of examples can be expelled from frameworks utilizing refactoring.

REFERENCES

[1] H. Solanki, "Comparative Study of Data Mining Tools and Analysis with Unified Data Mining Theory," International Journal of Computer Applications, vol. **75**, no. **16**, pp. **23–28**, **2013**.

[2] A. E. Hassan and Tao,Xie, "Mining software engineering data", in Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering - Volume 2 (ICSE '10), Vol. **2**. ACM, New York, NY, USA, **2010**. pp. **503-504**.

[3]M. S. Rawat, and S. K. Dubey, "Software defect prediction models for quality improvement: A literature study." IJCSI International Journal of Computer Science Issues Vol.**9** No. **5**,pp. **295**, **2012**.

[4]M. Jureczko and L. Madeyski, "Towards identifying software project clusters with regard to defect prediction," Proc. 6th Int. Conf. Predict. Model. Softw. Eng. - PROMISE '10, pp **1-10**, **2010**.

[5]Y. Suresh, J. Pati, and S. K. Rath, "Effectiveness of software metrics for object-oriented system," Procedia Technologyvol. **6**, pp. **420–427**, **2012**.

[6]M. S. Rawat, and S. K. Dubey, "Software defect prediction models for quality improvement: A literature study." IJCSI International Journal of Computer Science Issues Vol.**9** No. **5**,pp. **289**, **2012.**

[7]M. S. Rawat, and S. K. Dubey, "Software defect prediction models for quality improvement: A literature study." IJCSI International Journal of Computer Science Issues Vol.**9** No. **5**,pp. **292**, **2012.**

[8]S. Kim, H. Zhang, R. Wuand L. Gong. "Dealing with noise in defect prediction "in Proceedings of the 33rd International Conference on Software Engineering (ICSE '11). ACM, New York, NY, USA, pp**. 481-490**, 117,**2011.**

[9]M. Shepperd, Q. Song, Z. Sun, and C. Mair, "Data Quality: Some Comments on the NASA Software Defect Datasets," IEEE Transactions on Software Engineering, vol. **39**, no. **9**, pp. **1208–1215**, **2013.**

[10] M. Jureczko and L. Madeyski, "Towards identifying software project clusters with regard to defect prediction," Proc. 6th Int. Conf. Predict. Model. Softw. Eng. - PROMISE '10, pp **2-4**, **2010.**

[11]R. Goyal, P. Chandra, and Y. Singh, "Identifying influential metrics in the combined metrics approach of fault prediction," Springerplus, vol. **2**, no. **1**, pp**. 1–8**, **2013**.

[12]R. Subramanyam and M. Krishnan, "Empirical analysis of CK metrics for object-oriented design complexity: implications for software defects," IEEE Transactions on Software Engineering, vol. **29**, no. **4**, pp. **297–310**, **2003**.

[13] F. Provost and R. O. N. Kohavi, "Guest Editors' Introduction: On Applied Research in Machine Learning," New York, vol. **132,** no. **1998**, pp. **127–132**, **1998.**

[14] A. Pradesh and A. Pradesh, "The Importance of Statistical Tools in Research Work," Int. J. Sci. Innov. Math. Res., vol. **3**, no. **12**, pp. **50–58**, **2015**.

[15] T. Zimmermann, R. Premraj, N. Bettenburg, S. Just, A. Schröter, and C. Weiss, "What makes a good bug report?," IEEE Trans. Softw. Eng., vol. **36,** pp. **618–643**, **2010**.

[16]H. Wang, "Software Defects Classification Prediction Based On Mining Software Repository," Dissertation, **2014.**

[17] M. Jureczko. "Significance of different software metrics in defect prediction," Softw. Eng. An Int. J., vol. **1**, no. **1**, pp. **86–95,2011**

Authors Profile

Ms. Swathi K is a Ph.D Research Scholar at Department of Computer Science and Engineering Research Center, EWIT, Visvesvaraya Technological University, Karnataka, India. She received her M.Tech degree in Computer Science and Engineering from EWIT, Vivesvaraya Technological University in the year 2014. She obtained her B.E. Degree in Computer Science & Engineering from KSIT, Visvesvaraya Technological University in the year2012. She is currently pursuing Ph.D degree in Computer Science and Engineering under VTU. Her research area is Software Engineering. Having 5 years of teaching experience, She is Currently working as Assistant Professor in Department of Computer Science and Engineering, K.S. Institute of Technology, Bengaluru. She has Co-authored three text books being published in Lambert Publishing, Germany. She has delivered number of technical talks in different institutes in the field of Software Engineering and Networking.

Dr. Arun Biradar, is currently working as Professor & Head in Department of Computer Science & Engineering at East West Institute of Technology, Bengaluru. He obtained his Ph.D, M.Tech and B.E degree in Computer Science and Engineering. He is author/co-author of over 80+ International/National Publications. He is Recognized Research Supervisor in Computer Science & Engineering, Visvesvaraya Technological University, Belagavi. He is currently serving in department of Computer Science & Engineering at East West Group of Institutions, Bengaluru. His Interested areas of Research are Software Engineering, Computer Networks, Cloud Computing, Wireless Ad-Hoc Networks, IOT, Genetic Algorithms & Machine Learning. He guided/guiding 30+ Projects. He is a Professional Member of ISTE, CSI and IE. He is a Research Supervisor for 8 Ph.D Research Scholars under VTU. He has delivered number of expert talks in the field of Networks.

© 2019, IJCSE All Rights Reserved