International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 7
DOI: https://doi.org/10.17762/ijritcc.v11i7.7836
***Article Received**: 23 April 2023 **Revised**: 19 June 2023 **Accepted**: 01 July 2023*

Enhanced Approach for Bug Severity Prediction: Experimentation and Scope for Improvements

Veena Kulkarni[1], Dr. Anand Khandare[2]
[1]Computer Engineering
Thakur College of Engineering and Technology
Mumbai, India
veena.kulkarni@thakureducation.org
[2]Computer Engineering
Thakur College of Engineering and Technology
Mumbai, India
Anand.khandare@thakureducation.org

Abstract—Software development is an iterative process, where developers create, test, and refine their code until it is ready for release. Along the way, bugs and issues are inevitable. A bug can be any error identified in requirement specification, design or implementation of any project. These bugs need to be categorized and assigned to developers to be resolved. the number of bugs generated in any large scale project are vast in number. These bugs can have significant or no impact on the project depending on the type of bug. The aim of this study is to develop a deep learning-based bug severity prediction model that can accurately predict the severity levels of software bugs. This study aims to address the limitations of the current manual bug severity assessment process and provide an automated solution using various classifiers e.g. Naïve Bayes, Logistic regression, KNN and Support vector machine along with Mutual information as feature selection method, that can assist software development teams in giving severity code to bugs effectively. It seeks to improve the overall software development process by reducing the time and effort required for bug resolution and enhancing the quality and reliability of software.

Keywords- Bug tracking systems, Bug priority, bug severity, NLP, data mining, mutual information.

I. INTRODUCTION

Software bugs are common in software development projects and can have varying degrees of impact on the system's functionality and performance. there is huge load on the tester/manager to assign bugs to developers e.g. Mozilla bug repository receives an average of 135 new bug reports each day (Liu et al., 2013). Bug severity is an important aspect that determines the urgency, need and priority of bug fixing. Identifying the severity of a bug accurately is essential for helping the manager in allocating resources, prioritizing bug fixes, and thereby ensures the timely software releases.

As per the Forbes Technology Council dated Dec 19, 2022, Brittany Greenfield, Wabbi has mentioned that as an important feature of any effective bug tracking system is to have a context based prioritization method. thus research on bug severity prediction in software engineering is a topic of interest and importance in the international research community. Many researchers around the world are working on developing effective techniques and models to predict bug severity accurately. The research findings and methodologies are often shared and discussed in international conferences, journals, and research communities.

The contribution of this paper is as follows

1. perform literature review of papers on bug severity prediction
2. to use different feature selection methods
3. To compare different methods in ML to accurately predict the severity of bugs

II. BACKGROUND KNOWLEDGE

A. Bug

A bug is an error or defect in a software program that causes the program to behave in an unintended way by not performing the function properly [1]. In Essential Scrum, the bug is defined as bug only if it is identified after a user story has been completed and accepted from product owner. The occurrence of bugs in the software can be due to various reasons like programming errors, inadequate testing, requirements are incomplete or the customer has given changes in requirements. Whenever a new bug is encountered, the user/tester adds the bug in any bug tracking system with the priority and severity value as per their subjective judgment.

B. Feature

Feature and bugs differ between the product we have and the product we need. It can be a new functionality which is absent in the product and adds value to customer. A per the

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 7
DOI: https://doi.org/10.17762/ijritcc.v11i7.7836
***Article Received**: 23 April 2023 **Revised**: 19 June 2023 **Accepted**: 01 July 2023*

Agile manifesto "The highest priority is the customer satisfaction with timely and continuous delivery", whether the input is a bug or a feature, if it adds to customer value, it needs to be prioritized. However, if it is of no customer value, it need not be prioritized. Therefore, it is necessary to classify the bugs into bugs or features or enhancement.

C. Bug Classification

the input bug reports are classified as software bugs or enhancement. This can be done by analyzing the bug description. The summary or description is searched for keywords which specify if input report is a bug or an enhancement.

D. Bug Categorization

Bug categorization is the process of classifying software defects or issues (bugs) into different predefined categories or groups based on specific characteristics or attributes. This categorization helps streamline the bug tracking and resolution process, making it easier for developers and QA teams to prioritize and address the reported issues effectively. It involves analyzing various aspects of a reported bug and assigning it to a relevant category. The categories can normally encompass areas like severity, priority, component affected, functionality impacted, and more. Bugs can be categorized as performance bugs, security bugs, UI bug and functionality bugs

E. Bug severity

Bug severity is a factor that decides whether the bug should be resolved immediately or later. It is the impact of the bug on the working of the project. there are 4 categories of bug severity
1) Block: this is the bug with critical severity.it affects the working of project. Most of the critical severities are of functional bugs. e.g. The item cannot be added to the cart; the application is crashing on submit button click
2) Major: this type of bug does not block the working of the project however it is frustrating to the user e.g. images are not visible on the site; page is taking time to load
3) Minor: this type of bug has low impact on the working of a project. These bugs need to be reported and given priority. e.g. blurry images, spelling mistakes, color combination in pages

4) suggestion: it is mainly for improvement.

Bug severity prediction is the task of automatically predicting the severity level of a bug based on various attributes, such as bug description, the impacted components, and historical bug data. The goal is to develop predictive models and algorithms that can assist software developers, testers, and project managers in accurately assessing the severity of reported bugs

F. Bug Priority

Bug priority Bug priority is assigning the importance to the bugs reported. It is based on the severity and the impact the bug can have on the software and the business. The bugs can be prioritized as Critical, High, Medium and Low.

1. critical - e.g. bugs that cause the system to crash, cause data loss, or prevent the system from functioning properly. High- e.g. bugs that significantly impact system functionality.
2. Medium e.g. bugs that impact system functionality, but have less severe consequences.
3. low e.g. bugs that have minimal impact on system functionality, or are cosmetic in nature.

The importance of bug severity and priority prediction lies in its ability to optimize the bug fixing process. By predicting the severity, priority of bugs, the organizations can allocate resources effectively, focusing on critical and high-severity bugs that have a significant impact on system functionality and user experience. This helps in reducing the time and effort spent on low-severity bugs, allowing developers to prioritize critical issues and deliver high-quality software.

III. BUG TRACKING SYSTEMS

Bug tracking systems are software tools used in software development projects to manage and track reported issues or bugs. They help development teams to identify, document, prioritize, and resolve bugs in the software product. Bug tracking systems typically include features such as issue tracking, assignment, status updates, comments, and notifications. They enable efficient communication and collaboration among team members, as well as with stakeholders, by providing a centralized repository for bug reports and related information. By using bug tracking systems, software development teams can improve the quality and reliability of their software products, while also ensuring timely and effective bug resolution. following are some of the most used bug management/tracking systems.

A. Atlassian Jira

Jira is a very popular and widely used project management/ issue tracking system, in which issues can be either user stories, tasks or bugs. In Jira, projects can be created and team members can be added, given access to edit, comment on tasks. The issues can be tracked by manager and various reports can be generated.

B. GitHub Issues

GitHub Issues is also popular and useful bug tracking system that is built into the widely used software development platform, GitHub. It provides developers with a convenient and efficient way to track, manage, and resolve bugs and other issues in their code.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 7
DOI: https://doi.org/10.17762/ijritcc.v11i7.7836
***Article Received**: 23 April 2023 **Revised**: 19 June 2023 **Accepted**: 01 July 2023*

C. Bugzilla

Bugzilla is a robust and versatile bug tracking system that has gained a reputation as one of the most effective and reliable solutions for managing bugs and other issues in software development projects.

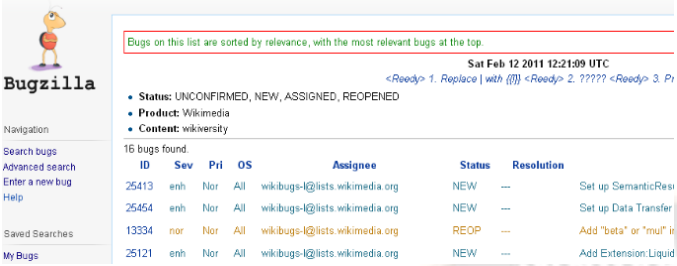

Figure 1. Bugzilla issue entries

As an open-source software solution, Bugzilla offers a range of features and capabilities that allow teams to effectively manage and track issues throughout the software development lifecycle. In these bug tracking systems, user can log bugs, assign description, severity and priority to each bug report, with help of which the bugs are triaged.

IV. RELATED WORK

X. Yang *et al.* [1] has Proposed a comprehensive taxonomy for prediction models. In this paper, 136 papers were considered for summarization. The key elements considered were the datasets, features, algorithms, and evaluation metrics. The author has suggested that a good prediction model relies on datasets it learns from. The factors to be considered for the quality are bias, noise, sixe and imbalance. big data the author proposed to consider features that are automatic and can reduce feature dimension while keeping quality. More elaboration for deep learning methods can be given

Y. Wei *et al.* [2] This paper combines knowledge intensified pre training i.e. understanding the internal knowledge of bug reports and contrastive learning i.e. understanding from global context. Six baseline approaches were considered, weighted f1 score is 30.68% up than others. Also open source datasets are considered.

J Kim et al. [3] The authors used Latent Dirichlet Allocation (LDA) for Topic-based feature selection and CNN-LSTM algorithm is used for classification. Achieved accuracy of 80.6% for predicting bug severity. They used preprocessing on bug reports. the topic classification of documents is performed by applying LDA. the feature selection algorithm extracts feature by the severity of topics. The features are applied to the CNN-LSTM algorithm. the CNN-LSTM works better than the individual CMM and LSTM. LSTM degrades when the length of data is long. the performance is better than the baseline models

G. Rodríguez-Pérez *et al*.[4] In this paper, bugs are of two types intrinsic and extrinsic. Just in time(JIT) model is used. the extrinsic bugs due to API change, requirement change cannot find the bug introducing change for bug fix. So removing these extrinsic bugs can increase performance of the JIT bug prediction. In paper the bugs are analyzed manually. For bug description, NLP techniques can be used, however techniques are needed to understand the code in the bug

Meng, Fanqi *et al.* [5], NLP is used for text processing. BERT and TF-IDF are used to extract the features. five classifiers (including K-Nearest Neighbor, Naive Bayes, Logistic Regression, Support Vector Machine and Random Forest) are used. F-Measure achieves from 87.3% to 95.5% .in this paper, Deep learning techniques can be used.

H. A. Ahmed *et al.* [6], In this paper, Preprocessing steps are done with feature selection using TF-IDF . Further to handle class imbalance problems SMOTE technique is used. for classification, Naive Bayes, Decision tree. Random Forest and Logistic regression are used. The research achieved 88.78% accuracy for category prediction and 90.43 priority prediction. training dataset can be varied

Hani Bani-Salameh *et al.* [7] In this paper a five-layer deep learning RNN-LSTM neural network is used on Jira dataset of 2000 bug reports and comparing results with SVM and KNN for prediction of bug priority. It increases F measure by 3% compared with SVM and 15.2 % compared with KNN. other classifiers can be checked

Jia, X. Chen *et al*. [8] in this paper logistic regression model is used. this method improves performance of f measure by upto 5.19%. other classifiers can be checked.

Rashmi Agrawal *et al*. [9] in this paper, word2vec, word embedding model is used for word meanings. It examines the word2vec with different classifiers and empirically analyses the effect of hyperparameter values. Bugzilla and JDK Bug system is considered.it has been analyzed for KNN, SVM, SVM1, Naive Bayes, XgBoost and Random Forest. filtering out rare words that changes the severity from normal to other levels.

Dao, Anh-Hien *et al*. [10] In this paper, CNN and the content-aspect, sentiment-aspect, quality-aspect, and reporter-aspect features of bug reports are considered. to improve prediction performance. Datasets used were Mozilla and Eclipse. MASP outperforms CNN by Accuracy, Precision, Recall, F1-measure, and the Matthews Correlation Coefficient (MCC) by 1.83%, 0.46%, 3.23%, 1.72%, and 6.61%. the sentiment algorithm Senti4SD is not used with bug reports. The same results can be checked for other datasets

S. Fang *et al.* [11] In this paper, two challenges including words' nonconsecutive semantics and the imbalanced data. Graph convolution network on weighted loss function is used fir predicting priority and weight loss function in training phase is done. F measure is 13.22 by weighted average. Open source datasets are considered

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 7
DOI: https://doi.org/10.17762/ijritcc.v11i7.7836
***Article Received**: 23 April 2023 **Revised**: 19 June 2023 **Accepted**: 01 July 2023*

Based from above study it is found that most of the work is focused on varying feature selection methods and varying classification algorithms.

In this paper we have proposed a method that uses RNN [6] with mutual Information as feature selection method to explore the content aspect of the bug reports for predicting the severity of any bug.

1. For feature selection Mutual Information technique can be used. (Mutual information is a statistical measure used for feature selection)
    a. Mutual information [30] is
    b. Let (X,Y) be pair of random variables over space X X Y . Their joint distribution is P_{XY} and their marginal distribution is P_X and P_Y their mutual information is given by

2. $I(X,Y)=D_{KL}((P)X,Y) \parallel P(X)\otimes P(Y))$ (1)
    D_{KL} is the Kullback-Leibler divergence

3. Diverse datasets can be used from different projects. The various open source datasets available are eclipse_bug_dataset, Mozilla bug dataset and gnome dataset.
4. Bugs can be categorized and pattern or trend in types of bugs occurring can be identified
5. Imbalance in datasets are there, they can be handled

In this paper the following objectives are considered,

- How do the different classifiers perform with mutual information as feature selection method?
- Does combining the attributes in dataset help to achieve better results

V. PROPOSED METHODOLOGY

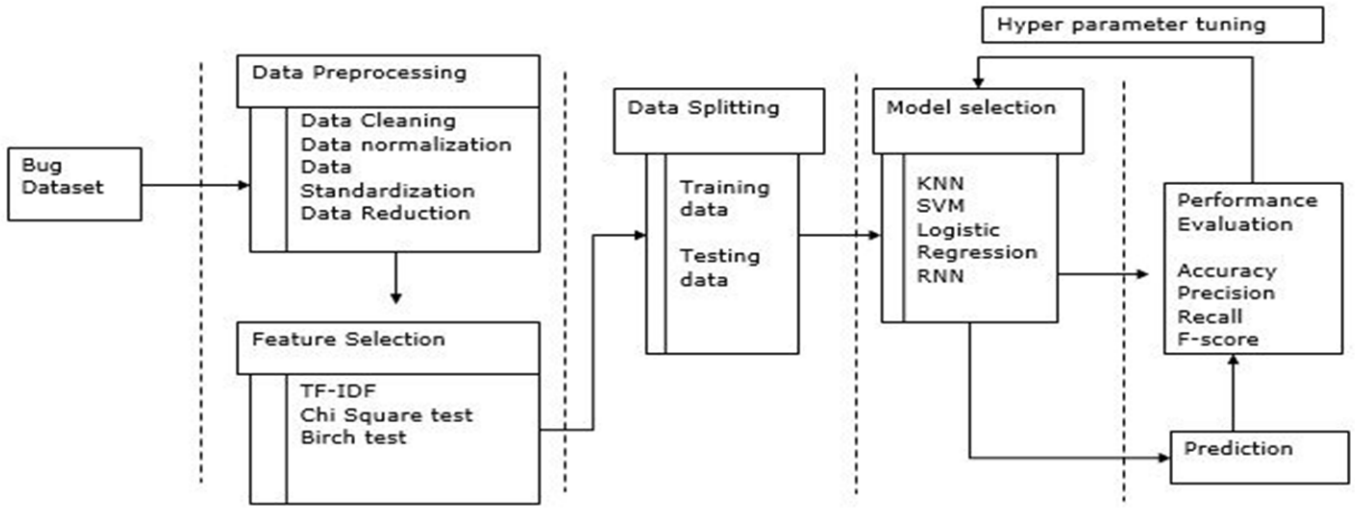

Figure 2. Proposed architecture

Based on the above mentioned topics for scope of improvement the following methodology is proposed. the bug dataset has been obtained from five projects i.e. Eclipse platform with 424767 bug reports, Firefox with 108750 bugs reports, Ellipse JDT with 45296 bug reports, mozilla_core with 1,83640 bug reports and Mozilla_thunderbird with 32552 bug reports.

The above proposed architecture is a generalized architecture for bug severity prediction. The goal of this research is to do the following,

1. consider bug reports in CSV format from heterogeneous data sources, there are various datasets online in which some datasets do not contain severity values, while for some datasets very few bug reports are marked with severity values. There are datasets which have both long and short descriptions of the bugs, the input data consists of bug reports that contain text descriptions and their associated severity levels.

2. The bug reports undergo text preprocessing to clean, tokenize, and normalize the text data
3. Next we can apply NLP preprocessing techniques like
    a. Tokenization - Split the text into individual words or tokens. This can be done using Python's NLTK library.
    b. Stop word removal- Remove common words that are not useful for classification such as "the", "a", "an", etc.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 7
DOI: https://doi.org/10.17762/ijritcc.v11i7.7836
***Article Received**: 23 April 2023 **Revised**: 19 June 2023 **Accepted**: 01 July 2023*

c. Stemming and Lemmatization- In this each word is reduced to its base form. e.g. 'running' and 'ran' will be reduced to 'run'.

4. Feature extraction is performed using various methods e.g. Term Frequency-Inverse Document Frequency (TF-IDF),

5. Mutual information to convert the text data into numerical vectors. There are many other feature extraction methods e.g. Birch test, Chi square test etc. However, in this paper we have compared various classifiers using TFIDF and Mutual information feature selection method.

6. Mutual information-based feature selection is applied to select the most informative features from the TF-IDF matrix.

7. The following classifiers are considered for bug severity prediction,

A. K-Nearest Neighbors (KNN),

The K-Nearest Neighbors (KNN), SVM, Logistic regression and Naïve Bayes classifiers are utilized for bug severity prediction, wherein KNN k represents the number of neighbors to consider. The KNN classifier predicts the severity level of new bug reports based on their similarity to the training data.

B. Support Vector Machine (SVM)

Support Vector Machine (SVM) SVM is used for both classification and regression tasks. In SVM goal is to find the optimal hyperplane that best separates data points belonging to different classes in a high-dimensional space. It aims to maximize the margin between the classes, which allows for better generalization to new data. SVM works well for both linearly separable and non-linearly separable datasets.

C. Logistic Regression

Logistic Regression is a classification algorithm used for binary classification problems it is a classification algorithm and not a regression algorithm. The model uses the logistic function (sigmoid function) to map the output to the range [0, 1], representing the probability of belonging to a particular Class. Logistic regression makes predictions by estimating the probability that an input data

D. Naïve Bayes

Naive Bayes is a probabilistic classification algorithm based on Bayes' theorem. in Naive Bayes the features are conditionally independent given the class label, which simplifies the calculations. Naive Bayes often performs well for text classification tasks, hence used for severity prediction.

E. Recurrant Neural Network

RNN is an artificial neural network designed to process sequential data, such as time series or natural language. Unlike the traditional feedforward neural networks, RNNs have connections that form cycles, allowing information to persist and be shared across time steps. This capability makes RNNs effective for tasks where past context is crucial, such as language translation, speech recognition, and sentiment analysis. However, traditional RNNs suffer from vanishing and exploding gradient problems, which can be mitigated using variants like LSTM (Long Short-Term Memory) and GRU (Gated Recurrent Unit).

The performance of the bug severity prediction model is evaluated using various evaluation metrics such as accuracy, precision, recall, and F1-score. The hyper parameters can be identified and tuned.

Algorithm : Find Evaluation parameter values

- **Input:** File in .csv format
- Get the dataset in an output csv format
- Load dataset
- Preprocessing of data
- Vectorize dataset using TF-IDF
- For feature selection apply mutual information
- Split the dataset into training and testing datasets
- Apply various classifiers on the dataset
- Calculate the evaluation parameters e.g. Accuracy, Precision, Recall, F1 Score, RMSE and R2

Figure 3. Algorithm using Mutual Information

VI. EXPERIMENTATION

To validate the approach, the following experimentation is done. The table is shown for four classifiers mainly Naïve Bayes, Support vector machine, KNN and Logistic regression with and without using Mutual information as feature selection method. The results for various evaluation parameters, obtained are given in table,

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 7
DOI: https://doi.org/10.17762/ijritcc.v11i7.7836
***Article Received**: 23 April 2023 **Revised**: 19 June 2023 **Accepted**: 01 July 2023*

TABLE 1: CLASSIFIERS AND EVALUATION VALUES FOR DATASET WITHOUT USING MUTUAL INFORMATION

Evaluation parameters/ Classifiers without MI	Accuracy	Precision	Recall	f1-score
Naïve Bayes	0.62	0.21	0.33	0.26
Logistic Regression	0.73	0.77	0.73	0.69
SVM	0.79	0.79	0.79	0.69
KNN	0.45	0.45	0.45	0.4

TABLE 2: CLASSIFIERS AND EVALUATION VALUES FOR DATASET USING MUTUAL INFORMATION

Evaluation parameters/ Classifiers	Accuracy	Precision	Recall	f1-score
Naïve Bayes	0.82	0.164	0.2	0.18
Logistic Regression	0.83	0.79	0.83	0.76
SVM	0.82	0.83	0.83	0.76
KNN	0.82	0.75	0.82	0.76

From the results, we can conclude that the accuracy value improves on using mutual information as the feature selection method over using TFIDF in the graphs given below, the accuracy, precision, recall, f1-score for four methods namely Naïve Bayes, SVM, KNN and Logistic regression the following performance measures are considered,

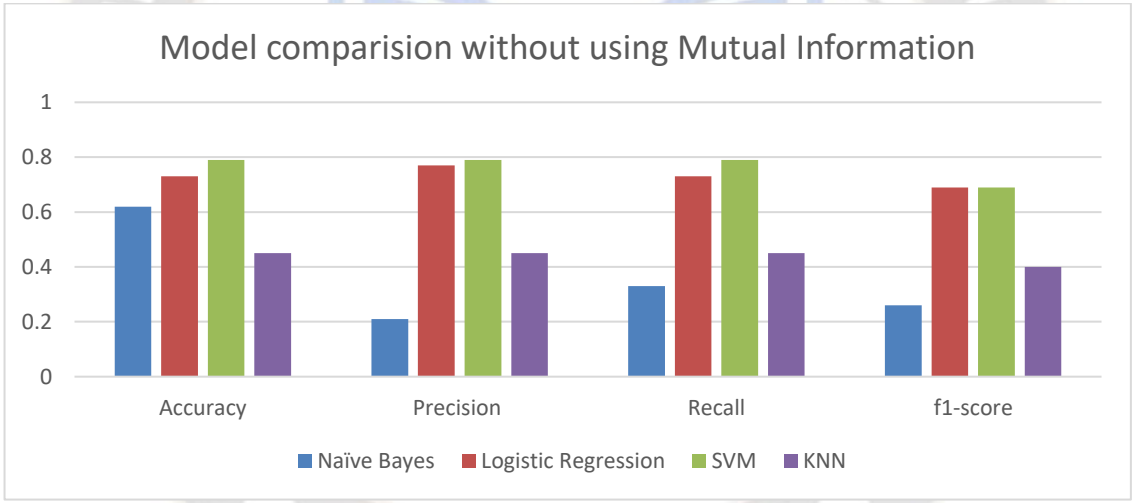

Figure 4. Model comparison without using Mutual information

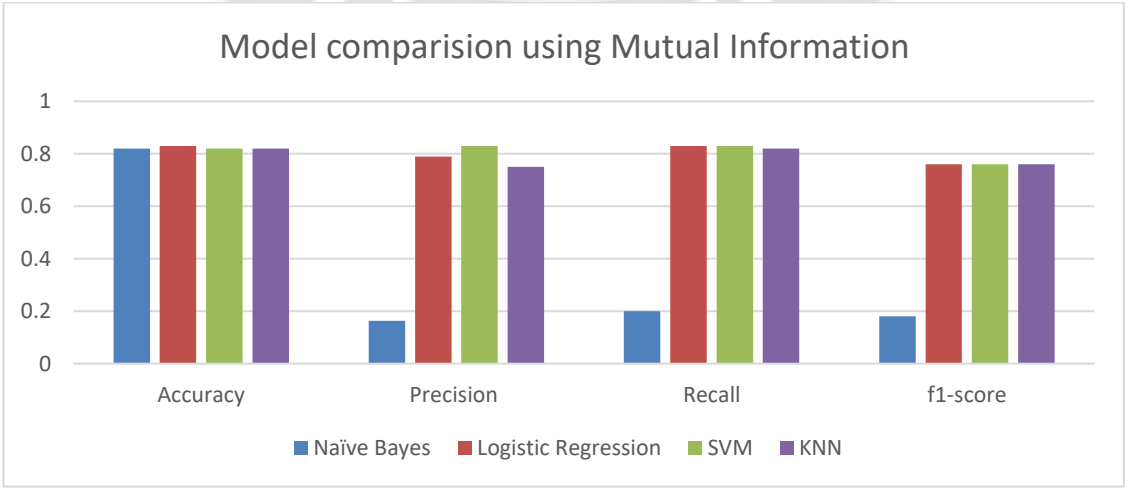

Figure 5. Model comparison using Mutual information

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 7
DOI: https://doi.org/10.17762/ijritcc.v11i7.7836
***Article Received**: 23 April 2023 **Revised**: 19 June 2023 **Accepted**: 01 July 2023*

Confusion Matrix: the performance of any classification algorithm can be measured by a table called confusion matrix. It summarizes the performance of a classification model by showing the number of true positives, false positives, true negatives, and false negatives

	Actually positive	Actually negative
Predicted positive	True positive (TP)	False Positive (FP)
Predicted Negative	False Negative (FN)	True Negative (TN)

Accuracy: It measures the percentage of correctly classified instances in the dataset.it can be said as the proportion of correctly predicted bugs from all bug reports given in a dataset.

$$Accuracy=(TP+TN)/(TP+FP+TN+FN) \quad (2)$$

Precision: It measures the proportion of true positive instances among all the predicted as positive. It can be said as the proportion of correctly predicted bugs of a specific severity level out of all the bugs predicted as that severity level.

$$Precision= TP/(TP+FP) \quad (3)$$

Recall: It measures the proportion of true positive instances that are correctly identified by the model out of all positive instances.it can be said as the ability of the model to identify all actual high severity bugs in the bug report datasets.

$$Recall= TP/ (TP+FN) \quad (4)$$

F1 Score: It is the harmonic mean of precision and recall, which gives a balanced measure of the two.

$$F1\ score=2*((Precision\ *Recall)/(Precision+Recall)$$

$$= TP/ (TP+½(FP+FN)) \quad (5)$$

F1 score value ranges from 0 to 1. A high value of F1 score indicates the model is able to predict the correct severity levels for bug reports

VII. EXPERIMENTAL RESULTS

In this paper we have compared the results of different classifiers mainly SVM, KNN Naïve Bayes and Logistic regression with using TF IDF as feature selection method and same classifiers with mutual information as feature selection methods. It shows improved accuracy, precision, recall and F1-score values for KNN, SVM and Logistic regression classifiers.

VIII.CONCLUSION

In this paper we have studied about the severity and priority of the bugs, importance of bug tracking systems and compared various bug tracking systems e.g. Jira, Trello, GitHub issues etc. based on their features. From the study it was concluded that a method is required which automatically predicts the severity of the bug and recommends it while submission of the bug in the bug tracking system. We have studied papers from various journals and conferences for bug severity prediction. The experimentation was done in which the bug reports were preprocessed and input to classifier after summarization, the results show that there is a decrease in accuracy after summarization. A method is proposed that uses bug reports from heterogeneous data sources and uses mutual information as a feature selection process and deep learning classifiers for varying sizes of datasets for finding the severity of a bug report.

IX. FUTURE SCOPE

The feature selection methods are applied on machine learning classifiers Logistic regression, SVM, KNN and Naïve Bayes, in future we can extend this method for deep learning algorithms like Recurrent neural network or its variant LSTM and GRU

ACKNOWLEDGMENT

The authors would like to that the inputs given by various reviewers that helped to improve the paper.

REFERENCES

[1] G. Yang, T. Zhang, and B. Lee, ‘‘Towards semi-automatic bug triage and severity prediction based on topic model and multi-feature of bug reports,’’ in Proc. IEEE 38th Annu. Comput. Softw. Appl. Conf., Jul. 2014, pp. 97–106

[2] X. Xia, D. Lo, E. Shihab and X. Wang, "Automated Bug Report Field Reassignment and Refinement Prediction," in IEEE Transactions on Reliability, vol. 65, no. 3, pp. 1094-1113, Sept. 2016, doi: 10.1109/TR.2015.2484074.

[3] Yang, Xinli, Jingjing Liu, and Denghui Zhang. 2023. "A Comprehensive Taxonomy for Prediction Models in Software Engineering" Information 14, no. 2: 111. https://doi.org/10.3390/info14020111

[4] J. Kim and G. Yang, "Bug Severity Prediction Algorithm Using Topic-Based Feature Selection and CNN-LSTM Algorithm," in IEEE Access, vol. 10, pp. 94643-94651, 2022, doi: 10.1109/ACCESS.2022.3204689.

[5] H. A. Ahmed, N. Z. Bawany and J. A. Shamsi, "CaPBug-A Framework for Automatic Bug Categorization and Prioritization Using NLP and Machine Learning Algorithms,"

in IEEE Access, vol. 9, pp. 50496-50512, 2021, doi: 10.1109/ACCESS.2021.3069248.

[6] Hani Bani-Salameh, Mohammed Sallam and Bashar Al shboul, "A Deep-Learning-Based Bug Priority Prediction Using RNN-LSTM Neural Networks", In e-Informatica Software Engineering Journal, vol. 15, no. 1, pp. 29–45, 2021. DOI: 10.37190/e-Inf210102.

[7] Y. Jia, X. Chen, S. Xu, G. Yang and J. Cao, "EKD-BSP: Bug Report Severity Prediction by Extracting Keywords from Description," 2021 8th International Conference on Dependable Systems and Their Applications (DSA), Yinchuan, China, 2021, pp. 42-53, doi: 10.1109/DSA52907.2021.00014.

[8] Rashmi Agrawal, Rinkaj Goyal, "Developing bug severity prediction models using word2vec",International Journal of Cognitive Computing in Engineering, Volume 2, 2021, Pages 104-115, ISSN 2666-3074, https://doi.org/10.1016/j.ijcce.2021.08.001.

[9] Dao, Anh-Hien, and Cheng-Zen Yang. 2021. "Severity Prediction for Bug Reports Using Multi-Aspect Features: A Deep Learning Approach" Mathematics 9, no. 14: 1644. https://doi.org/10.3390/math9141644

[10] Pundir, Prachi & Singh, Satwinder & Kaur, Gurpreet. (2019). A Machine Learning Based Bug Severity Prediction using Customized Cascading Weighted Majority Voting. International Journal of Computer Sciences and Engineering. 7. 1345-1350. 10.26438/ijcse/v7i5.13451350.

[11] W. Y. Ramay, Q. Umer, X. C. Yin, C. Zhu and I. Illahi, "Deep Neural Network-Based Severity Prediction of Bug Reports," in IEEE Access, vol. 7, pp. 46846-46857, 2019, doi: 10.1109/ACCESS.2019.2909746.

[12] H. Osman, M. Ghafari and O. Nierstrasz, "Hyperparameter optimization to improve bug prediction accuracy," 2017 IEEE Workshop on Machine Learning Techniques for Software Quality Evaluation (MaLTeSQuE), Klagenfurt, Austria, 2017, pp. 33-38, doi: 10.1109/MALTESQUE.2017.7882014

[13] Jin, Kwanghue & Dashbalbar, Amarmend & Yang, Geunseok & Lee, Byungjeong & Lee, Jung-Won. (2016). Improving predictions about bug severity by utilizing bugs classified as normal. Contemporary Engineering Sciences. 9. 933-942. 10.12988/ces.2016.6695.

[14] G. Catolino, F. Palomba, A. Zaidman, and F. Ferrucci, "Not all bugs are the same: Understanding, characterizing, and classifying bug types," Journal of Systems and Software, vol. 152, pp. 165–181, Jun. 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0164121219300536

[15] S. Gujral, G. Sharma, S. Sharma and Diksha, "Classifying bug severity using dictionary based approach," 2015 International Conference on Futuristic Trends on Computational Analysis and Knowledge Management (ABLAZE), Greater Noida, India, 2015, pp. 599-602, doi: 10.1109/ABLAZE.2015.7154933

[16] Ashima Kukkar, Rajni Mohana,"A Supervised Bug Report Classification with Incorporate and Textual field Knowledge" ,Procedia Computer Science,Volume 132,2018,Pages 352-361,ISSN 1877-0509,https://doi.org/10.1016/j.procs.2018.05.194

[17] N. Kanti-Singha Roy and B. Rossi, "Towards an Improvement of Bug Severity Classification," 2014 40th EUROMICRO Conference on Software Engineering and Advanced Applications, Verona, Italy, 2014, pp. 269-276, doi: 10.1109/SEAA.2014.51.

[18] P. Achimugu, A. Selamat, R. Ibrahim, and M. N. Mahrin, "A systematic literature review of software requirements prioritization research," Information and software technology, vol. 56, no. 6, pp. 568–585, 2014.

[19] A. Fawzi, D. Al-Shdaifat, M. Hammad, and E. E. Abdallah, "Severity prediction of software bugs," 2016 7th International Conference on Information and Communication Systems (ICICS), Apr. 2016. [Online]. Available: https: //ieeexplore.ieee.org/abstract/document/7476092

[20] Solanki, S. ., Singh, U. P. ., Chouhan, S. S. ., & Jain, S. . (2023). Brain Tumour Detection and Classification by using Deep Learning Classifier. International Journal of Intelligent Systems and Applications in Engineering, 11(2s), 279 –. Retrieved from https://ijisae.org/index.php/IJISAE/article/view/2624

[21] Tao Zhang, Jiachi Chen, Geunseok Yang, Byungjeong Lee, Xiapu Luo," Towards more accurate severity prediction and fixer recommendation of software bugs", Journal of Systems and Software,Volume 117,2016,Pages 166-184,ISSN 0164-1212,https://doi.org/10.1016/j.jss.2016.02.034.

[22] G. Fan, X. Diao, H. Yu, K. Yang, and L. Chen, "Software defect prediction via attention-based recurrent neural network," Scientific Programming, Vol. 2019, 2019.

[23] N. Serrano and I. Ciordia, ''Bugzilla, ITracker, and other bug trackers,'' 589 IEEE Softw., vol. 22, no. 2, pp. 11–13, Mar. 2005.

[24] Filip Pasarič and Maja Pušnik. "Comparison of Project Management Tools." SQAMIA 2022: Workshop on Software Quality, Analysis, Monitoring, Improvement, and Applications, September 11--14, 2022, Novi Sad, Serbia.

[25] https://upload.wikimedia.org/wikipedia/commons/6/65/Bugzilla_search_ex.png

[26] Sayeda Shamma Alia; Md. Nazmul Haque; Sadia Sharmin; Shah Mostafa Khaled; Mohammad Shoyaib "Bug Severity Classification Based on Class-Membership Information" 2018 Joint 7th International Conference on Informatics, Electronics & Vision (ICIEV) and 2018 2nd International Conference on Imaging, Vision & Pattern Recognition (icIVPR),25-29 June 2018 , Kitakyushu, Japan

[27] https://www.forbes.com/sites/forbestechcouncil/2022/12/19/14-essential-features-and-qualities-of-an-effective-bug-tracking-system/?sh=1d9fb7ff366e

[28] N. K. Singha Roy and B. Rossi, "Cost-Sensitive Strategies for Data Imbalance in Bug Severity Classification: Experimental Results," 2017 43rd Euromicro Conference on Software Engineering and Advanced Applications (SEAA), Vienna, Austria, 2017, pp. 426-429, doi: 10.1109/SEAA.2017.71.

[29] Tabassum, Nadia, Abdallah Namoun, Tahir Alyas, Ali Tufail, Muhammad Taqi, and Ki-Hyung Kim. 2023. "Classification of Bugs in Cloud Computing Applications Using Machine

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 7
DOI: https://doi.org/10.17762/ijritcc.v11i7.7836
***Article Received**: 23 April 2023 **Revised**: 19 June 2023 **Accepted**: 01 July 2023*

Learning Techniques" Applied Sciences 13, no. 5: 2880. https://doi.org/10.3390/app13052880

[30] S. Rastkar, G. C. Murphy and G. Murray, "Automatic Summarization of Bug Reports," in IEEE Transactions on Software Engineering, vol. 40, no. 4, pp. 366-380, April 2014, doi: 10.1109/TSE.2013.2297712.

[31] https://machinelearningmastery.com/information-gain-and-mutual-information/#:~:text=The%20mutual%20information%20can%20also,marginal%20probabilities%20for%20each%20variable.&text=%E2%80%94%20Page%2057%2C%20Pattern%20Recognition%20and,X)%20*%20p(Y))

[32] H. Mahfoodh and Q. Obediat, "Software Risk Estimation Through Bug Reports Analysis and Bug-fix Time Predictions," 2020 International Conference on Innovation and Intelligence for Informatics, Computing and Technologies (3ICT), Sakheer, Bahrain, 2020, pp. 1-6, doi: 10.1109/3ICT51146.2020.9312003