THINK INDIA JOURNAL
ISSN: 0971-1260
Vol-22-NO-30 Special Issue-December-2019

Review and Comparison on Softwarebug Predictionusing Machine Learning

Aashish Gupta[1], Shilpa Sharma[2]

[1]Research Scholar, School of Computer Science & Engineering, LPU, Jalandhar, India.

[2]Assistant Professor, School of Computer Science & Engineering, LPU, Jalandhar, India.

Abstract

As the internet users increasing, the amount of information available on internet is also increasing. Web application is the backbone of today's economy. Almost everything which requires human effort or human presence at a place can be replaced by web. Web application is a type of software which can be written in some qualified language and can be accessed using a desired environment through a simple IP Address or domain name from anywhere in the world. While developing a web application the same SDLC Lifecycle is followed. In the early stages of development, it's a compulsory task to take care of the programmatical mistakes or bugs to save time and effort during testing phase and prevent any runtime crisis. In this review paper, we will be studying various techniques supported by machine learning to handle bug prediction in web application and we will be proposing our approach to handle this task by improving performance of existing algorithms. Automated bug detection is an idea to reduce the developer effort or resources in development of an application.

Keywords: Machine Learning, Dataset, Supervised Learning, Random Forest, SVM

1. Introduction

Now a days Software project success is a major challenge. A project manager biggest headache is defects or bugs. These Bugs occurs due to poor design and implementation of code. The major challenge in defect free code is the level of knowledge. Suppose a team which is working on a project which consist of 5-6 developers some of them are experienced and others are fresher. Now the new developer has very little experience of what type of defects can be occur in this code in real life scenarios. So, they just implement that project without caring about future bugs. Later that application is distributed among users and they will experience in the non-uniform environment that bugs which affects the application rating, customer engagement, performance. Later to fix the bugs a lot of effort and resources are spend. Sometime these bugs or defects can be viewed in the form of vulnerability and hackers will use this as a lead to exploit your website or application and sometimes you compromise with important information or money. So, it reveals the problem faced by software industry and the importance of predicting the software defects in its development phase. So that we can take necessary steps toward safety precaution before these effects come out. The biggest challenge in software industry is to develop a 100% bug free application. This issue is difficult to achieve by the software development companies even if they are kept on testing. Basically, any application developed by human is not an automated process so having defect is a common or natural thing Nevertheless, the software development companies focus on early defect detection though several inspections, testing procedures.

So, to resolve this issue we reviewed a various approach based on machine learning.

ISSN: 0971-1260
Vol-22-NO-30 Special Issue-December-2019

2. supervised Learning approach:

In supervised learning approach, we have dataset which contains label means given text along with the category it belongs to and then our classifier is trained on those examples and based on learning classifier determines the category of newly given input. To implement supervised learning approach, we have got so many classification algorithms but we cannot use them all because our input is text (sequential data) and data has to be very large to produce some quality result. So, in that case we can only use deep learning sequence to sequence algorithms such as recurrent neural network, long short-term memory and can also use convolution neural network which might increase the performance of our sequence to sequence algorithm.

2.1 Naïve Bayes- Naïve Bayes is one of the most popular algorithm for text classification task.it is based on the concept of Bayes theorem where there is no independence assumption between the features. Naïve Bayes performs good when its performed on less Noisy and small data.

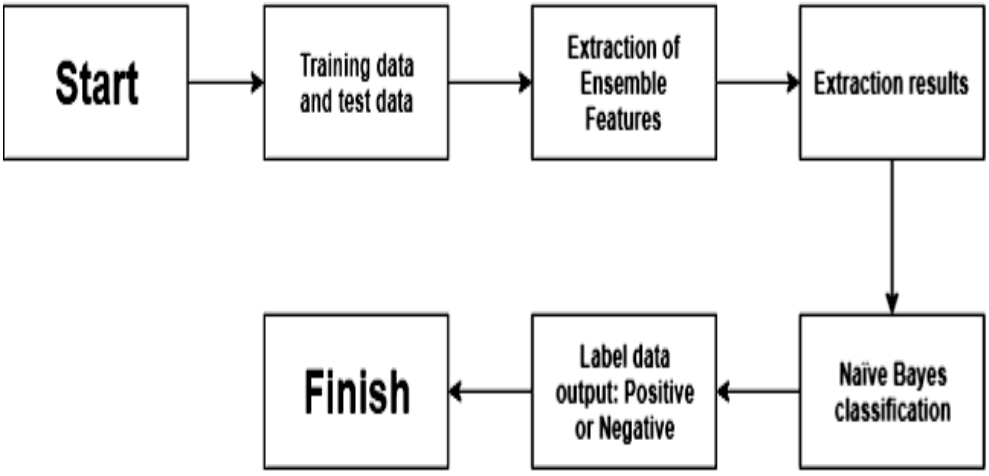

Fig 1.1 Naïve Bayes process

2.2 Random Forest- random forest is made of collection of decision trees.it is also known as ensemble learning algorithm. It is one of the most popular classifier due to its good performance on large and noisy dataset. It can automatically handle missing values and outliers but due to its complexity sometimes it tends to over fit on the training data which results in poor performance of the model. In most of the literature it is observed that it performed the best and most preferred for bug prediction feature.

ISSN: 0971-1260
Vol-22-NO-30 Special Issue-December-2019

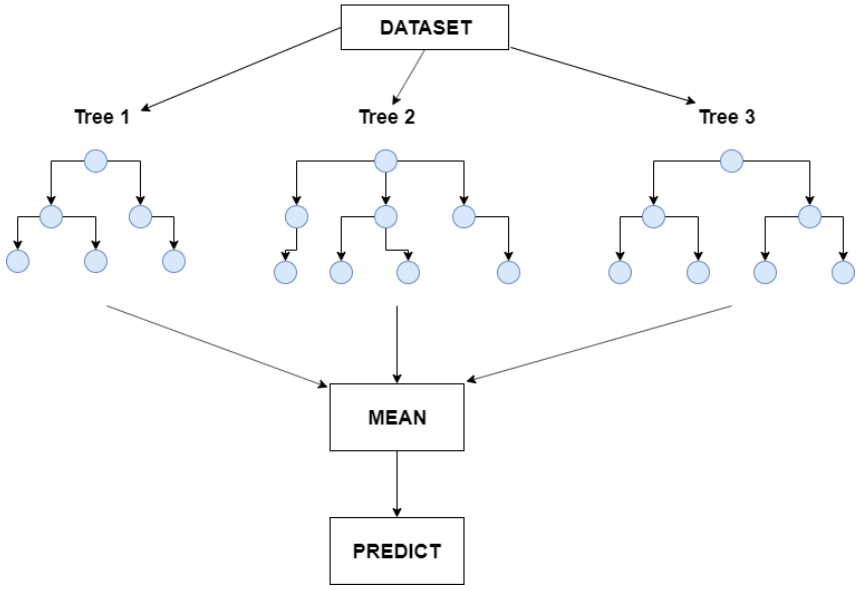

Fig 1.2 Random Forest

2.3 Decision Tree- It is a Supervised Machine Learning technique in which you explain what will be the input and the output corresponding to the data which is trained. In this technique data is splitted continuously according to a specific parameter. In this decision tree, we have two attributes that is parent nodes and leaves where leaves are the result and the parent nodes, where the data is being splitted. It is considered to be the best supervised learning algorithm due to which it is mostly used. It is basically a predictive model which provides us high accuracy, stability and ease of interpretation. Decision tree is very good at mapping non-linear relationship. Terminologies present in decision tree are splitting, decision node, root node, leaf node (Node which do not split), pruning, branch tree and parent and child node. Pros and cons of decision tree are - It is a layman algorithm, Person from non-analytical background can also implement this algorithm, it is the fastest way to significant variables and relation between multiple variables, it can deal with outliers and missing values, It can handle both numerical and categorical values, In case of small data set decision tree leads to over fitting, In case of continuous features decision tree does not performs well.

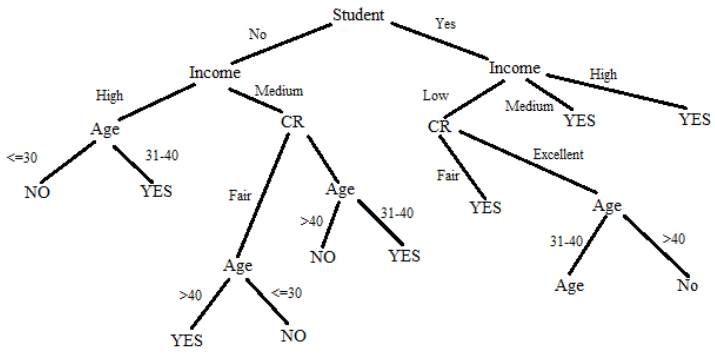

Fig 1.3 Decision Tree

ISSN: 0971-1260
Vol-22-NO-30 Special Issue-December-2019

2.4 Logistic Regression- Logistic regression is a type of classification algorithm. It gives the result in the form of binary outcome i.e. 1 or 0, True or False, Yes or No. It is a special case of linear regression, in linear regression our target outcome is continuous while in logistic regression our target outcome is categorical. It is used for prediction of outcome of dependent variable which is based on more than one independent variables (predictor). It is also used to detect if each feature is meaningful or not. It is a technique which is statistical which is used for calculating the relations between the variables. This model recognizes features and calculate the coefficient and weight for independent feature and then computes the category of tweet as a vector.

3. Unsupervised Learning approach

In unsupervised learning approach, we have dataset which only contains input text not any label which represent the category that text belongs to. So, in unsupervised learning approach all the input is clustered or grouped based on their characteristics or we can say that similarities between other paragraphs or document. The similarities are identified by the number of similar rare words present in each text paragraph or document. To implement unsupervised learning approach, we have got so many clustering algorithms such as LDA, K-Mean clustering, hierarchal clustering, Auto encoder, Restricted Boltzmann machines and self-organizing maps.

4. Uses areas of Machine Learning:

Machine learning is used efficiently in numerous fields.

Some of them are listed below:

- Automotive industry
- High technology and industry
- E-commerce
- Telecommunication sector
- Medical field
- Retail industry
- Packaged consumer products
- Media and show business
- Travel and transport sector
- Financial services
- Social media and online services
- Public services
- Education and research
- Health services
- Law enforcement and defense industry

ISSN: 0971-1260
Vol-22-NO-30 Special Issue-December-2019

5. Key Factors of Software Failure

Software Application are the backbone which supports different sectors such as ecommerce, social networking, finance etc. all the tasks are handled by such applications nowadays. The biggest challenge in software industry is to develop a 100% bug free application. This issue is difficult to achieve by the software development companies even if they are kept on testing. Basically, any application developed by human is not an automated process so having defect is a common or natural thing Nevertheless, the software development companies focus on early defect detection though several inspections, testing procedures.

Project Challenged Factors	% of Responses
1. Lack of User Input	12.8%
2. Incomplete Requirements & Specifications	12.3%
3. Changing Requirements & Specifications	11.8%
4. Lack of Executive Support	7.5%
5. Technology Incompetence	7.0%
6. Lack of Resources	6.4%
7. Unrealistic Expectations	5.9%
8. Unclear Objectives	5.3%
9. Unrealistic Time Frames	4.3%
10. New Technology	3.7%
Other	23.0%

Fig 1.4 Project Challenges Factors

6. Related Work

Methodology compared most of the machine learning approaches including both supervised and unsupervised learning. WEKA Tool used for experiment and PROMISE -NASA Data set is used to train the model [1]. Introduced retrieval and classification model using (CNN) and Long Short-Term Memory (LSTM) for accurate detection [2]. Proposed a method by using Supervised Learning algorithm mainly logistic regression, Naïve Bayes, and Decision Tree using historical data set. And used K-Fold cross validation technique [3]. Focused on Outlier Detection and removal, followed by dimension reduction [4]. Proposed bug detection as binary classification problem e.g. - correct and incorrect, trained the classifier which distinguish incorrect code from correct code by using deep Bugs framework [5]. Proposed a tool or framework named as defect detector framework which works with various compiler and languages e.g. javac, gcc, visual studio. [6]. Proposed an approach which uses minimum and accurate no of performing metrices at a time by using marginal R square values. Uses chose Eclipse JDT Core dataset [7]. Proposed a one-class SFP (Software Fault Prediction) Model using One Class SVM [8]. Focused on Vulnerability prediction of Web application using machine learning. In this paper input validation and sanitation attributes are generated. It computes static backward slice for each sink. Program analysis is based upon control flow graph, control dependence graph and system dependence graph of a web program [9]. Approach suggested

ISSN: 0971-1260
Vol-22-NO-30 Special Issue-December-2019

is first classify the bugs based on their priorities based on severity and component attribute. Uses Xmean Clustering algorithm with Bayes Net Classifier [10]. Uses Supervised learning. Datasets Used: KC1, MC1,AR1,AC6,MC2 to train the model then compares the results of naïve Bayes and j48(Decision Tree Classifier) [11].Used Supervised learning on 10 Data Sets Provided by means of NASA especially classifiers used are Bagging, guide vector machines (SVM), choice tree (DS), and random wooded area (RF) classifiers [12]. Data is collected from an open Source Software where data will be in a form of object-oriented matrices. Model proposed is genetic based Classifier Systems [13]. Uses NASA dataset from PROMISE dataset repository. Uses 12Supervised Learning Algorithm [14]. Implemented a fault-susceptible module prediction device using textual content-filtering based technique named "Fault- Prone Filtering" [15]. Proposed an approach named as WS3D which uses machine learning techniques that consist of Support Vector Machine (SVM) and Simulated Annealing (SA) on over 600 Web Applications [16]. Predicts a presence or absence of bug using machine learning classification models that can be deployed on Cloud. Datasets used "Churn of CK and other 11 object-oriented metrics over 91 versions of the system" and "Change metrics (15) plus categorized (with severity and priority) post-release defects" uses two class decision jungle and two class averaged perceptron algorithms [17] deep learning approach for multiclass severity classification using convolutional neural network(CNN) and Random Forest with Boosting. It uses natural language methods for pre-processing then n-gram for feature extraction then CNN for Feature pattern formation then lastly, random forest for multiple bug severity classes [18]. Based on supervised machine learning. Uses Naïve Bayes (NB), Decision Tree (DT) and Artificial Neural Networks (ANNs). Uses 3 datasets which are DS1, DS2, DS3 [19]. Introduces semi supervised dictionary learning technique and propose a cost sensitive kernelized semi supervised dictionary learning (CKSDL) approach to provide a unified and effective solution for both CSDP (Cross Project defect prediction) and WSDP (within-project semi supervised defect prediction) problems [20]. Instead of using single classifier/clustering. It shows that Soft Computing Techniques are much more efficient. Integrated Approaches by using Combination of Genetic algorithm, Fuzzy c-means based Random forest (GA-FCM based RF) instead of using standalone provides more accurate results by using NASA Dataset [21]. Proposed an approach to identify ageing related bug on two publicly available datasets of Linux and MySQL. Applied Extreme Learning Machine (ELM) with linear, polynomial and RBF kernels against different feature selection techniques [22]. Proposes a bug classification model using Multi-Class Semi Supervised SVM using different Kernel functions and tested against traditional SVM [23]. Proposed a novel approach for providing feedback on syntax errors that uses recurrent neural networks (RNNs). Tested against a dataset Of Edx which consist data of 140000 students [24]. Proposed a tree based convolutional neural network approach for programming language processing and compared with traditional NLP Models. The model is trained by 8000 programs [25]. Proposed an approach to improve the 13
Software development cycle in order to predict the bugs in the early stages of development, Smart Paste uses Deep Neural Network which improves the writing of code. Multiple datasets are used to train model which have around 1000k rows of each language and it learns on every successful compilation of program [26]. Proposed a semiautomatic approach named as BUGAID, it's used for detecting bug pattern. It works on unsupervised learning. By using data mining on 105k projects they used to find bug patterns [27]. Proposed an attentional neural network that detect features in context-dependent way. Using these features, the model generates a summary. Tested on 10 most popular programs on GitHub [28]. Proposed a Bayesian based framework for learning from large, unstructured code to predict bugs. First it prioritizes the

ISSN: 0971-1260
Vol-22-NO-30 Special Issue-December-2019

specification of program. Uses neural network and topic model [29]. Introduces a deep learning-based approach to bridge the gap between program semantics and bug prediction feature. Specifically, they have used Deep Belief Network (DBN) to learn features which is obtained from extracted token vectors from code. Used top ten open source java projects [30].

7. General Methodology:

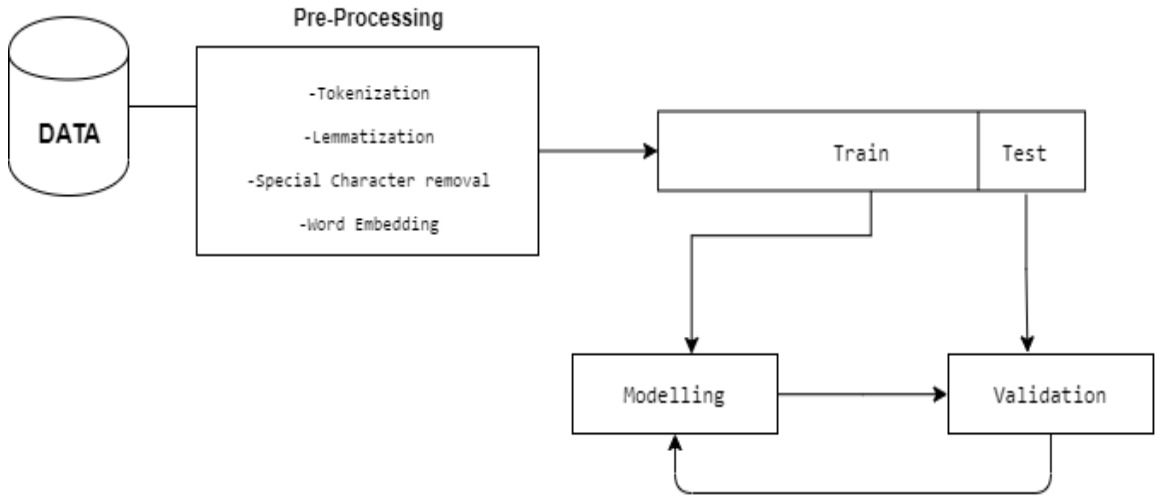

Fig 1.3 Supervised Learning model

Data- Online Data Collected from Different repository.
Which is being built from data collected from top Kernels e.g. – Visual Basic, Javac, Javascript.
Preprocessing- In this stage first part is tokenization. It means generation of tokens from sentences or source code. E.g. - "<? phpecho "<h2>same output</h2>";?> "
Tokens generated for the above sample line will be – '<', '?','php', 'echo', '"', '<', 'h2', '>''same', 'output', '<','/','h2', '>', ';', '?', '>'.
After tokenization, lemmatization is done.
Lemmatization means finding the words which are similar to each other and converting it to its first form of verb or its root form.
For e.g. - Programming is converted to program, if there are movie and movies in same sentence then movies will be converted to movies.
After lemmatization, Special character removal is done. In this process, only special characters are removed. For e.g. - '@', '#', '$' etc.
Last process in pre- processing is word embedding's. In this, word vectors are projected in 2D space and find the relation between them. For e.g. - boy and girl, Agree and disagree, car and truck etc.
Train / Test – we will split our dataset in two parts i.e. train and test in the ratio of 8:2. 80% of the data will be used for training and 20% will be used for testing of the model.

ISSN: 0971-1260
Vol-22-NO-30 Special Issue-December-2019

Modelling - we will be using machine learning algorithms for training like logistic regression, naïve Bayes, random forest, decision tree, support vector machine and many more to improve the performance of the model.
Validation – we will be validating performance of our model on test data set.

8. Role of Machine Learning in field of Software Engineering:

Machine Learning is expanded its feet to almost every field nowadays so it's no strange about its usage in field of software development. In any field where the need of future prediction or intelligent decision is required machine learning is being used. Machine learning find its usage in field of software engineering to help in reducing its complexity, development time, efforts. According to previous researches machine learning can resolve various software development issues.
In the table below we will look at different machine learning approaches for various software development lifecycle tasks:

SE tasks	**Applicable type(s) of Learning methods**
Requirement gathering	AL, BBN, LL, DT, ILP
Rapid Prototype	GP
Component Reuse	IBL (CBR4)
Cost/effort prediction	IBL(CBR), DT, BBN, ANN
Defect Prediction	BBN
Test oracle generation	AL (EBL5)
Test Data adequacy	CL
Validation	AL
Reverse Engineering	CL

Table 1: Role of ML in SDLC

ISSN: 0971-1260
Vol-22-NO-30 Special Issue-December-2019

9. Literature Survey

Reference	Year	Methodology	Result
1	IJSEA, Vol.6, No.3, May 2015	Compared most of the machine learning approaches including both supervised and unsupervised learning. WEKA Tool used for experiment and PROMISE -NASA Data set is used to train the model.	Naïve Bayes:83.47% on MC1,PC2, and PC5 and PC3 MLP: 89.14% Decision Tree:88.47% Random Forest:89.08% k-means:87.29% Bagging:89.386%
2	2017 IEEE	Proposed retrieval and classification model using Convolutional Neural Network and Long Short-Term Memory (LSTM) to increase detection accuracy.	achieved accuracy of 90% on datasets from Bugzilla which are based on OpenOffice,Eclipseand NetBeans
3	2019 IEEE	Proposed a method by using Supervised Learning algorithm mainly Logistic regression, Naïve Bayes, and Decision Tree using historical data set. And used K-Fold cross validation technique.	Logistic Regression:96%, Decision Tree:96 %, Naive Bayes 92%
4	IJIRCCE 2015	Focused on Outlier Detection and removal, followed by dimension reduction	Used Regression analysis approach in neural network
5	arXiv :2018	Proposed bug detection as binary classification problem e.g. - correct and incorrect, trained the classifier which distinguish incorrect code from correct code by using deep Bugs framework	Tested on 150,000 .js files with accuracy of 89-95%, and takes around only 20ms per file and found 102 programming mistakes with 68% true positive rate

ISSN: 0971-1260
Vol-22-NO-30 Special Issue-December-2019

6	NASA: 2019	Proposed a tool or framework named as defect detector framework which works with various compiler and languages e.g. javac, gcc, visual studio.	Used two-fold classification approach - Back Propagation, and graph edit distance measure technique. Results are not published yet
7	ScienceDirect,2016	Proposed an approach which uses minimum and accurate no of performing metrices at a time by using marginal R square values. Uses chose Eclipse JDT Core dataset	Results shows that the proposed Model is being tested against simple regression model and multiple regression model in which its concluded that proposed model works better than other two
8	2016	proposed a one-class SFP (Software Fault Prediction) Model using One Class SVM	Tested on 6 Data Sets named as CM1, KC3,MC1, MW1, PC1,PC2 and achieved highest Accuracy of 68.8%
9	IJSER,2017	Focused on Vulnerability prediction of Web application using machine learning. In this paper input validation and sanitation attributes are generated. It computes static backward slice for each sink. Program analysis is based upon control flow graph, control dependence graph and system dependence graph of a web program.	Used supervised and semi-supervised learning for predictor. And achieved good accuracy but for certain types only
10	Springer, 2015	Approach suggested is first classify the bugs based on their priorities based on severity and component attribute. Uses Xmean Clustering algorithm with Bayes Net Classifier	Performance improvement of 9.24% is achieved with Xmean and Bayes Net Classifier, Simple K-Mean an improvement of 5.67% is obtained and With Expectation Maximization, the improvement was 3.49%.

ISSN: 0971-1260
Vol-22-NO-30 Special Issue-December-2019

11	IRJET, 2019	Uses Supervised learning. Datasets Used: KC1,MC1,AR1,AC6,MC2 to train the model then compares the results of naïve Bayes and j48(Decision Tree Classifier)	Found that Decision tree classifier provides better results than naïve Bayes algorithm. Where naïve Bayes have accuracy of 77.07 and Decision Tree(J48) have 86.27
12	JSEA, 2019	Used Supervised learning on 10 Data Sets Provided by NASA mainly classifiers used are support vector machines (SVM), decision tree classifier (DS), and random forest classifier.	Results found that Random Forest Performs Better than All other Supervised Classifiers with the accuracy of 91%
13	IEEE,2019	Data is collected from an open Source Desktop Application where data will be in a form of object oriented matrices. Model proposed is genetic based Classifier Systems	Results found that genetic based Classifier Systems which uses artificial neural network Performs Better than Back Propagation and Levenberg-Marquardt
14	2017	Uses NASA dataset from PROMISE dataset repository. Uses Supervised Learning Algorithm	SVM and Random Forest Provides the best Results
15	IEEE,2015	Introduced a fault-inclined module prediction tool the use of text-filtering based approach named “Fault-Prone Filtering”.	Study performed on 3 open source project databases. And by using this tool they achieved an accuracy of 0.67, precision of 0.63 and recall of 0.9
16	IEEE,2017	Proposed an approach named as WS3D which makes use of gadget gaining knowledge of strategies that include Support Vector Machine (SVM) and Simulated Annealing (SA) on over six hundred Web Applications.	Results show that WS3D achieves a median of 91% and 94% of precision and recall that is an awful lot higher than any other set of rules

ISSN: 0971-1260
Vol-22-NO-30 Special Issue-December-2019

17	2018	Predicts a presence or absence of bugs the usage of machine studying classification fashions that may be deployed on Cloud. Datasets used "Churn of CK and other 11 object-oriented metrics over 91 versions of the system" and "Change metrics (15) plus categorized (with severity and priority) post-release defects" uses two class decision jungle and two class averaged perceptron algorithms	achieved F1 score on Dataset 1 is 91.5% and on Dataset 2 is 90.7%
18	2019	Deep learning knowledge of technique for multiclass severity classification the use of convolutional neural network (CNN) and Random Forest with Boosting.It uses natural language methods for pre-processing then n-gram for feature extraction then CNN for Feature pattern formation then lastly, random forest for multiple bug severity classes	Achieved average accuracy of proposed model is 96.34%. F-measure of proposed BCR and existing model is 96.43 and 84.24 respectively
19	Research Gate 2018	Based on supervised machine learning of. Uses Naïve Bayes Classifier, Decision Tree (DT) and Artificial Neural Networks.Uses 3 datasets which are DS1, DS2,DS3	Results found that Average accuracy on the 3 provided datasets of Naïve Bayes is 93.4%, Decision Tree is 97.1% and ANN 95.1%. So, Decision Tree Performs Best over the other methods
20	IEEE,2018	Introduces semi supervised lexicon learning technique and propose a value sensitive kernelized semi supervised dictionary learning (CKSDL) approach to produce a unified and effective resolution for each CSDP (Cross Project defect prediction)and WSDP(within-project semi supervised defect prediction) issues.	Performed experiment on widely used 16 projects and observed that CKSDL is Performed better than CSDP & WSDP both. As on NASA Dataset the Observed Accuracy is 76% whereas the accuracy of KSDL is 73%.

ISSN: 0971-1260
Vol-22-NO-30 Special Issue-December-2019

21	ICECDS-2017	Instead of Using single classifier/clustering. It shows that Soft Computing Techniques are much more efficient. Integrated Approaches by using Combination of Genetic algorithm, Fuzzy c-means based Random forest (GA-FCM based RF) instead of using standalone provides more accurate results. Used NASA Dataset	Achieved Accuracy of 98.23% when used combined soft computing techniques whereas standalone technique e.g. fuzzy c achieved accuracy of 76.2%
22	IEEE,2017	Proposed an approach to identify ageing related bug on Two publicly available datasets of Linux and MySQL. Applied Extreme Learning Machine(ELM) with linear, polynomial and RBF kernels against different feature selection techniques.	Results shows that the proposed approach successfully achieves better performance of 0.59 AUC than Without SMOTE of 0.53AUC
23	IEEE,2012	proposes a bug classification model using Multi-Class Semi Supervised SVM using different Kernel functions and tested against traditional SVM	accuracy obtained is 81%
24	arXiv,2016	Proposed a novel method for offering feedback on syntax mistakes that uses Recurrent neural networks (RNNs).Tested against a dataset Of Edx which consist data of 140000 students	This method can generate repairs for approximately 32% of submissions on the large MOOC Courses dataset
25	arXiv,2015	Proposed a tree based CNN approach for programming language processing and compared with traditional NLP Models. The model is trained by 8000 programs.	proposed model achieved accuracy of more than 87.06% whereas bag-of-words feature yields an accuracy of 62.03%

ISSN: 0971-1260
Vol-22-NO-30 Special Issue-December-2019

26	arXiv,2017	Proposed an approach to improve the software development cycle in order to predict the bugs in the early stages of development, Smart Paste uses Deep Neural Network which improves the writing of code. Multiple datasets are used to train model which have around 1000k rows of each language and it learns on every successful compilation of program	achieves 58.6% accuracy
27	FSE-2016	Proposed a semiautomatic approach named as BUGAID, it's used for detecting bug pattern. It works on unsupervised learning. By using data mining on 105k projects they used to find bug patterns	Using BugAID, they performed an evaluation of 105,133 commits from 134 server-aspect JavaScript tasks. They discovered 1,031 BCTs and 219 alternate kinds.
28	arXiv,2016	Proposed an attentional neural network that detect features in context-dependent way. Using these features, the model generates a summary. Tested on 10 most popular programs on GitHub	Achieved a better accuracy than existing approach as the F1- score of existing approach is 45.2 and the observed F1-Score of proposed deep learning approach is 59.6
29	arXiv,2017	Proposed a Bayesian-based framework for studying from huge, unstructured code to expect bugs.First it prioritizes the specification of program. Uses neural network and topic model	When compared to the non-Bayesian methods the observed accuracy is 53% and that of Bayesian (proposed) approach its 80%. When tested over a large data
30	IEEE,2016	Introduces a deep getting to know-primarily based technique to bridge the space among software semantics and defect prediction function. Specifically, they have got used Deep Belief Network to examine functions that are acquired from extracted token vectors from code. Used the top ten open source java	Significantly improves within-project defect prediction (WPDP) and cross-project defect prediction (CPDP) compared to traditional features. WPDP is improved by 14.2 % F1 and CPDP by 8.9% F1.

ISSN: 0971-1260
Vol-22-NO-30 Special Issue-December-2019

		initiatives.	

Table 1: Literature Review

10. Conclusion:

The biggest challenge in software industry is to develop a 100% bug free application. This issue is difficult to achieve by the software development companies even if they are kept on testing. Basically, any application developed by human is not an automated process so having defect is a common or natural thing Nevertheless, the software development companies focus on early defect detection though several inspections, testing procedures. So, to resolve this issue we propose a novel approach based on machine learning with greater accuracy to tackle such type of this which will help the developers to save a lot of effort and time.

The proposed framework will distinguish incorrect from correct code. it can able to provide an intelligent prediction with greater accuracy and speed as we train our model with large set of database with is consist of large amount of data about the bugs founded in previous real life software's from where our model can learn and it can also learn on the go means whenever the program successfully compiles our model can take it as an example and learn from it In addition to this we don't need to hardcode every statement or condition check as there are thousands of condition.

The issues that we faced is less availability of latest data source for implementing because most of the data available in online repositories is unlabeled so it's difficult to train our model which such data if we do then it will lead to less accuracy.

References:

[1] Saiqa Aleem, Luiz Fernando Capretz and Faheem Ahmed "Benchmarking machine learning technique for software defect detection" IJSEA, Vol.6, No.3, May 2015.

[2] Jayati Deshmukh, Annervaz K M, Sanjay Podder, Shubhashis Sengupta, Neville Dubash "Towards Accurate Duplicate Bug Retrieval using Deep Learning Techniques"2017 IEEE.

[3] S. Delphine Immaculate,M. Farida Begam,M. Floramary "Software Bug Prediction Using Supervised Machine Learning Algorithms" 2019 IEEE.

[4] Surbhi Parnerkar, Ati Jain, Vijay Birchha "An Approach to Efficient Software Bug Prediction using Regression Analysis and Neural Networks" IJIRCCE 2015.

[5] MICHAEL PRADEL, KOUSHIK SEN "DeepBugs: A Learning Approach to Name-based Bug Detection" arXiv :2018.

[6] Markland J. Benson "Toward Intelligent Software Defect Detection" NASA: 2019.

[7] Shruthi Puranika, Pranav Deshpandea, K Chandrasekarana "A Novel Machine Learning Approach for Bug Prediction" ScienceDirect, 2016.

[8] LIN CHEN, BIN FANG, ZHAOWEI SHANG "Software fault prediction based on One-Class SVM" IEEE -2016.

[9] Vignesh M, Dr. K. Kumar "Web Application Velnerability prediction using machine learning" IJSER, 2017.

[10] Neetu Goyal, Naveen Aggarwal, and Maitreyee Dutta "A Novel Way of Assigning Software Bug Priority Using Supervised Classification on Clustered Bugs Data" Springer, 2015.

[11] Meenakshi, Dr. Satwinder Singh "Software Bug Prediction using Machine Learning Approach" IRJET, 2019.

[12] Abdullah Alsaeedi, Mohammad, Zubair Khan "Software Defect Prediction Using Supervised Machine Learning and Ensemble Techniques: A Comparative Study" JSEA, 2019.

[13] Amod Kumar,Ashwni Bansal "Software Fault Proneness Prediction Using Genetic Based Machine Learning Techniques" IEEE,2019.

[14] Meiliana, Syaeful Karim,Harco Leslie Hendric Spits Warnars, Ford Lumban Gaol, Edi Abdurachman,Benfano Soewito "Software Metrics for Fault Prediction Using Machine Learning Approaches" IEEE-2017.

[15] Keita Mori and Osamu Mizuno "An Implementation of Just-In-Time Fault-Prone Prediction Technique Using Text Classifier" IEEE, 2015.

[16] Ali Ouni, Marwa Daagi, Marouane Kessentini, Salah Bouktif, Mohamed Mohsen Gammoudi. "A Machine Learning-Based Approach to Detect Web Service Design Defects" IEEE, 2017.

[17] Uma Subbiah, Muthu Ramachandran and Zaigham Mahmood "Software Engineering Approach to Bug Prediction Models using Machine Learning as a Service (MLaaS)" IEEE-2019.

[18] Ashima Kukkar , Rajni Mohana , Anand Nayyar , Jeamin Kim , Byeong-Gwon Kang and Naveen Chilamkurti "A Novel Deep-Learning Based Bug Severity Classification Using Convolutional Neural Networks and Random Forest with Boosting" IEEE-2019.

[19] Awni Hammouri, Mustafa Hammad, Mohammad Alnabhan, Fatima Alsarayrah "Software Bug Prediction using Machine Learning Approach" Research Gate 2018.

[20] Fei Wu, Xiao-Yuan Jing, Ying Sun, Jing Sun, Lin Huang, Fangyi Cui, and Yanfei Sun "Cross-Project and Within-Project Semisupervised Software Defect Prediction: A Unified Approach" IEEE,2018.

[21] Pushphavathi T P "An Approach for Software Defect Prediction by Combined Soft Computing" ICECDS-2017.

[22] Lov Kumar,Ashish Sureka "Aging Related Bug Prediction using Extreme Learning Machines" IEEE,2017.

[23] Ayan Nigam,Bhawna Nigam,Chayan Bhaisare,Neeraj Arya "Classifying the Bugs Using Multi-Class Semi Supervised Support Vector Machine" IEEE,2012.

[24] Sahil Bhatia,Rishabh Singh "Automated Correction for Syntax Errors in Programming Assignments using Recurrent Neural Networks" arXiv,2016.

[25] Lili Mou,Ge Li, Lu Zhang, Tao Wang, Zhi Jin "Convolutional Neural Networks over Tree Structures for Programming Language Processing" arXiv,2015.

[26] Miltiadis Allamanis,Marc Brockschmidt "SMARTPASTE: Learning to Adapt Source Code" arXiv,2017.

[27] Quinn Hanam,Fernando S. de M. Brito,Ali Mesbah "Discovering Bug Patterns in JavaScript" FSE-2016.

[28] Miltiadis Allamanis,Hao Peng,Charles Sutton "A Convolutional Attention Network for Extreme Summarization of Source Code" arXiv,2016.

[29] Vijayaraghavan Murali,Swarat Chaudhuri,Chris Jermaine "Finding Likely Errors with Bayesian Specifications" arXiv,2017.

[30] Song Wang,Taiyue Liu and Lin Tan "Automatically Learning Semantic Features for Defect Prediction" IEEE,2016.

