Acta Scientific COMPUTER SCIENCES

Volume 5 Issue 4 April 2023

Research Article

Comparative Study: To Analyze Software Defects Using Machine Learning Algorithms

Qasim Ali[1], Asma Zubedi[2], Fatima Najmuddin[1]*, Imran Memon[3] and Salahuddin Sadar[1]

[1]Mehran University of Engineering and Technology, Jamshoro, Software Department, Pakistan

[2]Beijing University of Post and Telecommunication, Management and Beijing, China

[3]Bahria University Karachi Campus, Computer Science Department, Karachi, Pakistan

***Corresponding Author:** Fatima Najmuddin, Mehran University of Engineering and Technology, Jamshoro, Software Department, Pakistan.

Received: March 22, 2023

Published: March 29, 2023

Abstract

The dependency on software has been increasing with each passing day, due to which reliability and quality of software has been becoming more and more crucial. The quality of product increases when the defects and faults will decrease. To find the defect in software product, many approaches were proposed but machine learning approach is very useful. Machine learning classify data into defective and non-defective modules

In this paper, 15 datasets from NASA promise repository named AR1, AR3, AR5, AR6, CM1, KC1, KC2, KC3, MC1, MC2, MW1, PC1, PC2, PC3, PC4 are analyzed using 7 machine learning algorithms (K-Nearest Neighbor, Linear regression, Random Forest, Naïve Bayes, Support Vector Machine (SVM), logistic regression and decision tree) and apply 10 k-fold cross validation in RapidMiner tool. In RapidMiner performance of ML algorithm in term of accuracy is calculated and the summary of the result shows that SVM perform best.

Similarly, these 15 datasets are also analyzed using 8 machine learning algorithms named Random Forest, Naïve Bayes, simple logistic, Sequential minimal optimization (SMO), K-star, REF (decision tree), K-Nearest Neighbor (KNN), decision table and apply 10 k-fold cross validation in WEKA simulation tool. In WEKA performance of ML algorithm in term of correctly classified instances is calculated and the summary of the result shows that Random Forest and simple logistic perform best.

Keywords: Software Defect; Machine Learning Algorithms; Weka; RapidMiner; Nasa Promise Datasets

Introduction

In this technology world, where almost everything is relied on computer, the software-based system is progressively grown [8,9]. Due to the extensive use of application software, software quality is still challenging problem for the developer. Software quality is the most critical element in software product, as if there are more defects in the software, then reputation of company will be affected and vice versa. Software quality refers to an attribute having reasonable defects or defects free [3]. Software defects is an error or fault in a program that generates improper output [4,7]. Defects significantly affects software quality. Existence of defects in software product affects software reliability, quality, and maintenance cost. If the defect is observed after formation, it creates a burden on development company as they have to re-create some software code or module, which effects the costs of overall product development.

The Software Defects Prediction detect defects before software products are released, as finding bugs after release is a tedious and time-consuming process. Software defects prediction is significant step in software development life cycle (SDLC). Because finding defects in early stages of development improve quality of software product and reduce maintenance cost and increase resource utilization. Many techniques are proposed for software defects prediction, but most familiar one technique is machine learning technique.

The research comprises on 15 datasets from NASA promise repository are downloaded and analyzed using 7 machine learning algorithms (K-Nearest Neighbor, Linear regression, Random Forest, Naïve Bayes, Support Vector Machine (SVM), logistic regression and decision tree) in RapidMiner tool.

And these 15 datasets are also analyzed using 8 machine learning algorithms named Random Forest, Naïve Bayes, simple logistic, Sequential minimal optimization (SMO), K-star, REF (decision tree), K-Nearest Neighbor (KNN), decision table in WEKA simulation tool.

The rest of the paper is organized as follows: Section II contains work related to software defects prediction. Section III contains detailed discussion of datasets used for analysis. Section IV explains machine learning algorithms used in this research. Section V is about results and discussions. Section VI contains conclusion based on the experiments and analysis.

Related Work

Machine learning techniques are widely used in previous researches to predict software defects. Instance-based learning algorithm was proposed to predict defects in dynamic/runtime data collected software and for that, intelligent software defect analysis tool (ISDAT) is design to monitor and assess defects in software module [1].

Multiple kernel ensemble learning (MKEL) approach [2] were used to predict software defect prediction and classification. NASA datasets were used as test data to examine the performance of all compared methods and result reveals that MKEL perform was good.

Defective software module can affect maintenance cost, quality, and reliability. In this regard, most popular Machine Learning algorithm Artificial Neural Network (ANN), Particle Swarm Optimization (PSO), decision tree, Naïve Bayes and linear classifier are applied on 7 NASA dataset using Keel tool and the conclusion shows that linear classifier has dominance over others [3].

The Research done by N. Kalaivani, Dr. R. Beena [4], discuss software defects, software management and software defects prediction and their approaches, techniques, and performance measures in detail.

The overall software success relies on software bug prediction as finding bugs increase reliability, efficiency, and quality. On historical data 3 Machine learning algorithm Naïve Bayes, Decision tree, Artificial neural network is applied. They conclude that ML algorithms are highly effective with a higher degree of accuracy and comparison measure shows that ML algorithm has better performance than other approaches [5].

In Ref. [6] datasets collected from promise repository are analyzed using Random Forest algorithm in Rapid Miner Machine Learning Tool. The results show that accuracy will increase when the number of trees were increased. The maximum accuracy was 99.59% and minimum was 85.96%.

The prediction of software defects in early SDLC has useful effect on software quality. Based on software metrics, many approaches have proposed. On 10 datasets, SVM, decision tree and random forest were applied, and experimental results shows that random forest performance was best as compare to others [7].

The hybridized approach was used in [8] that comprises on Software Framework (Random Forest, PCA, Naïve Bayes and the SVM), and 5 datasets (PC3, MW1, KC1, PC4, and CM1), are examine using the WEKA.

Research [9] comprises of seven datasets obtain from NASA repository. The result shows that the Neural Networks and Gradient Boosting classifier works better than other algorithms.

Datasets

In this research, 15 datasets available for software defects detection were downloaded from NASA promise repository that are publicly available were analyzed using machine learning algorithms. Table 1 describe the datasets.

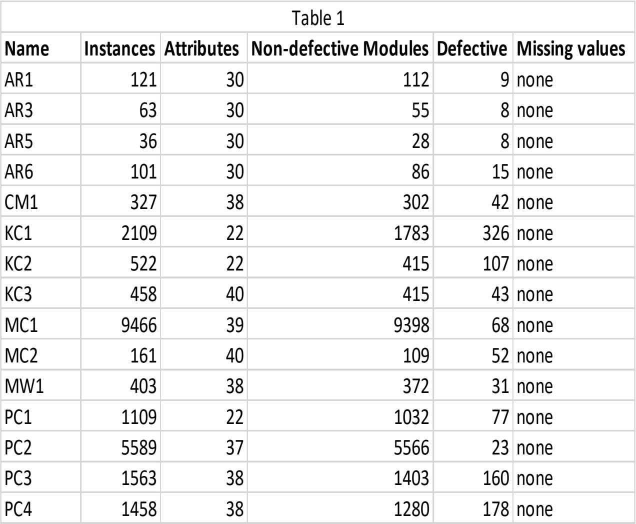


Table 1

Name	Instances	Attributes	Non-defective Modules	Defective	Missing values
AR1	121	30	112	9	none
AR3	63	30	55	8	none
AR5	36	30	28	8	none
AR6	101	30	86	15	none
CM1	327	38	302	42	none
KC1	2109	22	1783	326	none
KC2	522	22	415	107	none
KC3	458	40	415	43	none
MC1	9466	39	9398	68	none
MC2	161	40	109	52	none
MW1	403	38	372	31	none
PC1	1109	22	1032	77	none
PC2	5589	37	5566	23	none
PC3	1563	38	1403	160	none
PC4	1458	38	1280	178	none

Table 1: Datasets.

Machine learning algorithm

The research aims to analyze and assess Machine Learning algorithms, named K-Nearest Neighbor (KNN), Linear regression, Random Forest, Naïve Bayes, Support Vector Machine (SVM), logistic regression and decision tree, Sequential minimal optimization (SMO), decision table, K-star. Brief description of the selected ML algorithms is given below.

Its goal is to find all the nearest neighbors around a new unknown data point to find out which class it belongs to. It is a distance-based approach.

Linear regression

It accomplishes the task of predicting a dependent variable rely on a given independent variable(s). Thus, this algorithm finds a linear relationship between the dependent variable and the independent variables.

Random forest

Random forest creates framework that predicts value function by using various input factors. Input parameters are reflected by all internal node. Tree has leaves that describes main factors that states input factor parameter traverse from the root to the leaf.

Naïve bayes (NB)

Naïve Bayes algorithm is based on Bayes theorem that gives probability of an event by using the previous knowledge. It contains families of algorithms that considers that the presence or absence of a distinct property of the class is not related to the presence and absence of any other property.

Support vector machine (SVM)

The support vector machine algorithm aims to build best decision boundary known as hyperplane in a n-dimensional space so that new datapoint can uniquely organize in future.

Logistic regression

It is a statistical model which uses a logistic function to model a binary dependent variable.

Decision tree (DT)

Decision Tree generates representation of all possible results of a decision in a graphical way. No more than one 'or' condition are included in decision tree. The aim of the decision tree is to give a worthwhile means to visualized and subsequent understanding of on hand decision's possibilities along with range of possible outcomes.

Sequential minimal optimization (SMO)

Sequential Minimal Optimization is commonly used in training step of support vector machine (SVM). Quadratic programming issue in support vector machine can be resolved using sequential minimal optimization (SMO) algorithm. SMO algorithm or software package for SVM were implemented to used SVM on real-world applications.

K-Star

K-star is an instance-based classifier, in which the test instance class is based upon those training instances class which are resemble to it, by using some similarity function. It uses an entropy-based distance function, therefore it differs from another instance-based learner.

Decision table

Decision Table represent all possible condition and actions in a tabular form. Decision table can easily derive from decision tree. More than one 'or' condition are included in decision table. Based on the data entered in the table, the decision table generates the rules for structuring logic.

Result and Analysis

In this research we used two data analyzing tool named WEKA and RapidMiner.

Rapid miner

Datasets obtained from NASA repository is downloaded in comma separated format (.CSV), format which is supported by RapidMiner tool. The algorithm used in RapidMiner tool are K-Nearest Neighbor, Linear regression, Random Forest, Naïve Bayes, Support Vector Machine (SVM), logistic regression and decision tree and 10 k-cross validation were applied on it. In RapidMiner performance of ML algorithm in term of accuracy is calculated and the summary of the results are shown in table 2.

Datasets	KNN	Linear Regression	Random Forest	Naive Bayes	SVM	Logistic Regression	Decision Tree
AR1	89.29%	90.13%	90.96%	77.69%	91.79%	86.79%	88.46%
AR3	85.71%	81.43%	88.57%	85.71%	84.13%	70.24%	88.57%
AR5	86.67%	83.33%	80.83%	84.17%	80.83%	80.83%	75.00%
AR6	84.18%	86.09%	84.18%	84.09%	87.18%	74.27%	86.18%
CM1	85.47%	86.64%	86.04%	82.24%	87.81%	85.76%	86.34%
KC1	81.32%	86.06%	85.30%	82.46%	85.30%	85.63%	85.21%
KC2	82.74%	82.55%	82.36%	83.51%	84.67%	83.12%	83.70%
KC3	88.21%	90.39%	89.51%	84.71%	90.84%	90.19%	90.84%
MC1	99.39%	99.28%	99.44%	93.64%	99.29%	99.35%	99.28%
MC2	66.51%	63.97%	72.16%	73.35%	72.10%	68.90%	73.38%
MW1	91.08%	91.82%	91.07%	82.85%	92.32%	90.84%	91.55%
PC1	92.79%	92.97%	93.87%	89.09%	92.97%	92.25%	92.15%
PC2	99.59%	99.55%	99.57%	97.23%	98.94%	99.46%	99.55%
PC3	88.17%	89.89%	89.64%	46.20%	89.76%	89.57%	89.12%
PC4	85.80%	90.81%	88.55%	87.31%	89.85%	91.90%	88.89%

Table 2: Rapid miner.

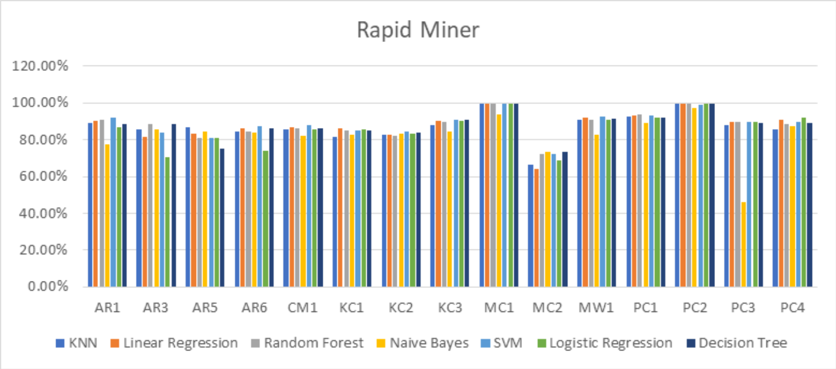


Figure 1: Bar Chart of Rapid Miner.

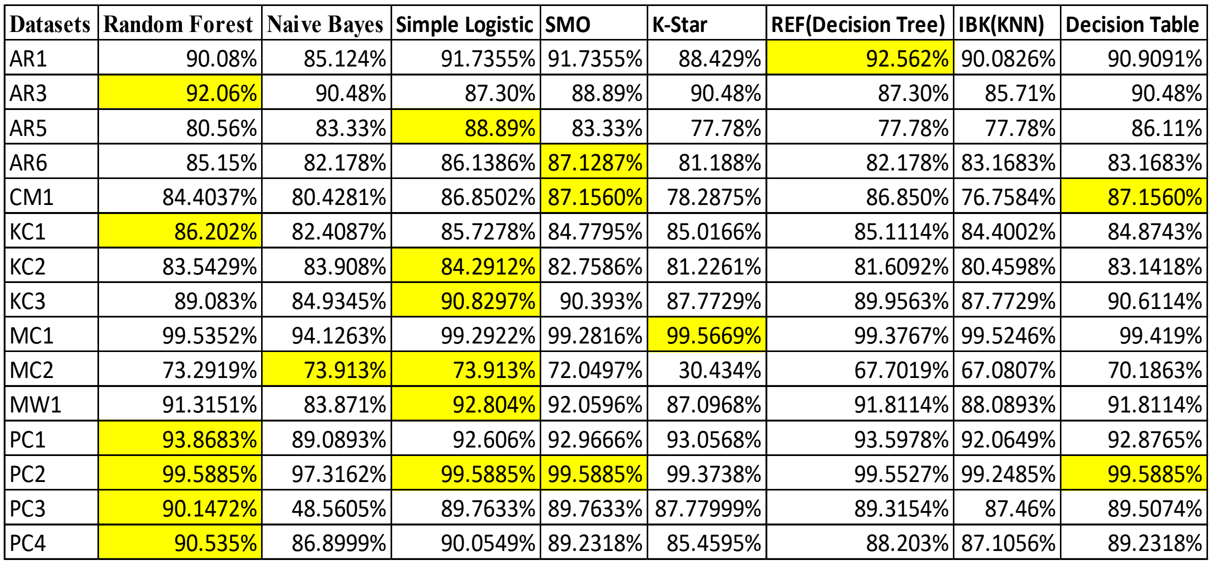
Citation: Fatima Najmuddin., *et al.* "Comparative Study: To Analyze Software Defects Using Machine Learning Algorithms". *Acta Scientific Computer Sciences* 5.4 (2023): 103-108.

Based on above results, it has been clearly visualized that the SVM provides better results in 06 out of 15 selected datasets and rest of the algorithms like logistic regression perform good 01 dataset, KNN and Linear Regression perform good in 02 datasets, random forest and decision tree in 03 datasets and Naïve Bayes in none of the datasets.

WEKA

Datasets obtained from NASA repository is downloaded in attribute-relation file format (.ARFF) format which is supported by WEKA tool. The algorithm used in WEKA tool are K-Nearest Neighbor (KNN), Sequential minimal optimization (SMO), Random Forest, Naïve Bayes, Support Vector Machine (SVM), simple logistic and decision table, K-star and REF tree (decision tree) and 10 k-cross validation were applied on it. In WEKA tool, the performance of ML algorithm in term of correctly classified instances is calculated and the summary of the results are shown in table 3.

Based on above results, it has been clearly visualized that the Random Forest and Simple Logistics provides better results in 06



Datasets	Random Forest	Naive Bayes	Simple Logistic	SMO	K-Star	REF(Decision Tree)	IBK(KNN)	Decision Table
AR1	90.08%	85.124%	91.7355%	91.7355%	88.429%	92.562%	90.0826%	90.9091%
AR3	92.06%	90.48%	87.30%	88.89%	90.48%	87.30%	85.71%	90.48%
AR5	80.56%	83.33%	88.89%	83.33%	77.78%	77.78%	77.78%	86.11%
AR6	85.15%	82.178%	86.1386%	87.1287%	81.188%	82.178%	83.1683%	83.1683%
CM1	84.4037%	80.4281%	86.8502%	87.1560%	78.2875%	86.850%	76.7584%	87.1560%
KC1	86.202%	82.4087%	85.7278%	84.7795%	85.0166%	85.1114%	84.4002%	84.8743%
KC2	83.5429%	83.908%	84.2912%	82.7586%	81.2261%	81.6092%	80.4598%	83.1418%
KC3	89.083%	84.9345%	90.8297%	90.393%	87.7729%	89.9563%	87.7729%	90.6114%
MC1	99.5352%	94.1263%	99.2922%	99.2816%	99.5669%	99.3767%	99.5246%	99.419%
MC2	73.2919%	73.913%	73.913%	72.0497%	30.434%	67.7019%	67.0807%	70.1863%
MW1	91.3151%	83.871%	92.804%	92.0596%	87.0968%	91.8114%	88.0893%	91.8114%
PC1	93.8683%	89.0893%	92.606%	92.9666%	93.0568%	93.5978%	92.0649%	92.8765%
PC2	99.5885%	97.3162%	99.5885%	99.5885%	99.3738%	99.5527%	99.2485%	99.5885%
PC3	90.1472%	48.5605%	89.7633%	89.7633%	87.77999%	89.3154%	87.46%	89.5074%
PC4	90.535%	86.8999%	90.0549%	89.2318%	85.4595%	88.203%	87.1056%	89.2318%

Table 3: WEKA.

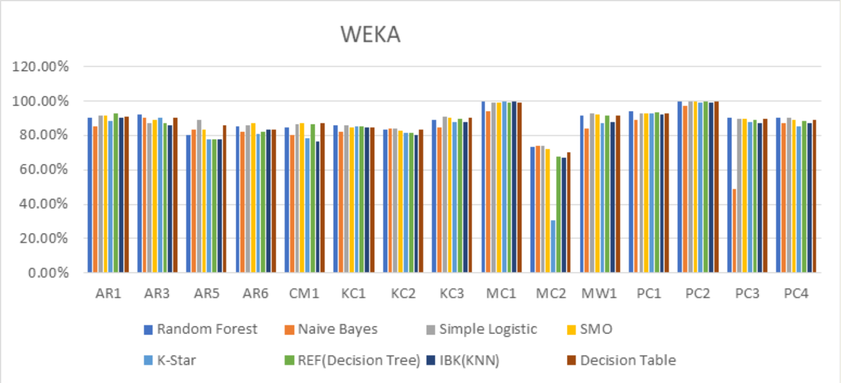


Figure 2: Bar Chart of WEKA.

out of 15 selected datasets and rest of the algorithms like Naïve Bayes, REF and K-star provides better results in 01 dataset, Decision table provides better results in 02 datasets, SMO in 03 datasets, KNN in none of the datasets.

Conclusion

Software defects prediction is an essential process in software development. Software defects is inversely proportion to software quality, increase in number of defects will decrease quality of the software product.

In this regard, we perform experiment on publicly available datasets from NASA Promise repository and analyzed them by using machine learning algorithm in two different tools named Rapid Miner and WEKA. Experimental result shows that, in Rapid Miner Support Vector Machine gives highest accuracy and in WEKA simple logistic and random forest perform best among all other algorithm. This field is still in much more research cap. Researcher may apply these more algorithm on these datasets or create a hybrid approach of these machine learning algorithm for predicting defects in software product.

Bibliography

1. Challagulla B., *et al*. "Empirical assessment of machine learning based software defect prediction techniques". *International Journal on Artificial Intelligence Tools* 17 (2008): 389-400.

2. Z Zhang., *et al*. "Multiple kernel ensemble learning for software defect prediction". *Automatic Software Engineering* 23 (2016): 569-590.

3. P Deep singh and A Chug. "Software defect prediction anlysis using machine learning algorithm". in 2017 7th International Conference on Cloud Computing, Data Science and Engineering-Confluence, IEEE (2017): 775-781.

4. N Kalaivani and D R Beena. "Overview of software defect prediction using machine learning algorithms". *International Journal of Pure and Applied Mathematics* 118 (2018): 3863-3871.

5. A Hammouri., *et al*. "Software bug prediction using machine learning approach". *International Journal of Advanced Computer Science and Applications* 9 (2018): 78-83.

6. Y Soe., *et al*. "Software Defect Prediction Using Random Forest Algorithm". 2018 12th South East Asian Technical University Consortium (SEATUC) 1 (2018): 1-5.

7. A Alsaeedi and M Zubair Khan. "Software defect prediction using supervised machine learning and ensemble techniques: A comparative study". *Journal of Software Engineering and Applications* 12 (2019): 85-100.

8. C Lakshmi Prabha and D N Shivakumar "Software defect prediction using machine learning techniques". in 2020 4th International Conference on Trends in Electronics and Informatics (ICOEI) (48184), IEEE (2020): 728-733.

9. M Shah and N Pujara. "A Review On Software Defects Prediction Methods". arXiv (2020): 1-5.