PERFORMANCEOF HETEROGENEOUS ENSEMBLE APPROACH WITH TRADITIONAL METHODS BASED ON SOFTWARE DEFECT DETECTION MODEL

Dr. E.SREEDEVI[1], Mrs. D. SREE LAKSHMI[2], Mrs. A. Divya[3], Dr. P.V.S.S. GANGADHAR[4], Ms. V. PREMALATHA[5]

[1]Department of C.S.E, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Guntur, A.P, India.,

[2]Department of C.S.E, PVP Siddhartha Institute of Technology,Vijayawada, A.P,India.,

[3]Department of C.S.E, PVP Siddhartha Institute of Technology,Vijayawada, A.P,India

[4]Scientist-E, NIC, MEITY, Govt.of India, AP State Centre, Vijayawada.,A.P,India.,

[5]Department of C.S.E,KoneruLakshmaiah Education Foundation, Vaddeswaram, Guntur, A.P, India.

E-mail: [1]sridevi_fed@kluniversity.in, [2]dslakshmi@pvpsiddhartha.ac.in, [3]adivya@pvpsiddhartha.ac.in, [4]pvss.gangadhar@nic.in, [5]premaawilliams@kluniversity.in

ABSTRACT

Identifying defective modules from the developed software is very much indispensable for constructive management and control of software testing. Software defect detection models helps a lot in effective allocation of limited testing resources. In this context several software defect detection modelling has been proposed by using machine learning algorithm. The main intention of heterogeneous ensemble model is to regulate each of its specific model strengths and weakness undoubtedly leading to the finest passable decision being taken overall. In this paper, we proposed heterogeneous ensemble learning, a defect detection model in which different learners are combined to form heterogeneous ensemble learning. Performance of individual learning models is compared with our proposed heterogeneous ensemble models, and it shows that our model is giving a better accuracy then the models developed by individual learning models. The evaluation results show that our proposed model achieved up to 98% accuracy which is more than the evaluation accuracy achieved by individual learning models.

Keywords: *Software Defect Detection Models, Feature Selection, Ensemble Learning, Accuracy, Defect Detection.*

1. INTRODUCTION

Software testing is a problematic. Identifying defects is still one of the most important skill in software testing skills. Successful quality control process reduces the cost of software updates and maintenance. Defect detection is especially important duringsoftware quality control and a huge number of methods have been proposed to find defective modules in a software system.

Defect forecasting provides an efficient method of identifying vulnerabilities that arise as a result of manual or automated errors during the SDLC stages [26]. Software quality is becoming increasingly important in the modern era as software programme addiction grows. Errors and flaws in software can have an impact on the quality of the software, resulting in client dissatisfaction. Software defect detection is one of the focus areas in software development. It is specified as “the process of detecting defects in a developed software system”. Software defects are spotted by using software metrics. An accurate detection of defects leads to a successful development of a software. An inaccurate finding of defects is major familiar factors of software failure. Utilization of machine learning models in software defect detection had been giving more observation in recent years, which improves the defect detection accuracy which in turn improves the quality of the developed software. Yet, no one of the current models showed to be satisfactory under all set of conditions. The achievement of particular models is not reliable, that is, differs from dataset to dataset. Accordingly,

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

there is a requirement to raise defect detection models that are dependable and produce more accuracy. Ensembles of heterogeneous machine learning models are applicable for this purpose.

The objective of an ensemble methodology is to handle one and all its single models durability and weakness undoubtedly leading to the finest achievable opinion being taken total. In this paper, we have matured a heterogeneous ensemble of few optimized hybrid machine learning models for software defect detection. Variety of linear and non-linear combiners has been used. We have supervised an experimental examine to assess and analyse the performance of these ensembles by applying NASA defect datasets.

The rest of this paper is organises as follows. Section 2 reviews related work. Section 3 describes the heterogeneous ensemble methodology that has been developed. Section 4 reports the conducted empirical study and discusses its results. Section 5 provides concluding remarks and directions for future work.

2. RELATED WORK

Software defect detection has been under study for a very long time. Numerous software defect detection models are exists in the literature. A lot of software developers and researchers examined in this area of software defect detection. Machine learning algorithms like ANN, DT, SVM, RF etc. and their applications have been used for software defect detection.

Researchers employed a variety of categorization approaches to create these models. Researchers have used a combination of statistical and machine learning techniques to predict fault proneness models and reduce software development and maintenance costs in these strategies. The machine learning technique is the most prominent among them. [21].

It is well known reality that distinct software metrics might associate with defect-proneness in software systems[1]. Therefore remaining uncorrelated metrics could enlarge classification performance [1]. AbdullateefiO.iBalogun showed that the use of feature selection for pre-processing helps to generate better results though caution to be exercised in selecting the appropriate feature selection for a classification process[2].

Machine learning algorithms have been more prominent in the previous decade and are still one of the most widely used approaches for fault prediction. [22,23].

Mohammad Zubair Khan compared the results of different machine learning algorithms with hybrid ensemble learning for software defect detection, it is effective at reducing testing efforts, the identification of defective classes in software has been considered [3]. Abdullah Alsaeedi, MoammadZubair Khan, concentrated on different most well-known machine learning algorithms that are extensively used to predict software defects [4].

Challagulla et al.[23] evaluated the performance of several machine learning algorithms and statistical models for predicting software quality in an empirical study. Experiments using different predictor models on four different real-time software defect datasets revealed that the 1R rule-based classification learning algorithm and Instance-based learning with Consistency-based subset evaluation technique are more consistent in achieving accurate predictions than other models.

Using a hybrid of wrapper and filter techniques, Huda et al.[24] provided a system for finding significant metrics to build and assess an automated software defect prediction model.

Bowes et al. [25] recently conducted a sensitivity analysis of the prediction uncertainty generated by four distinct classifiers. Their findings revealed that classifier ensembles using non-majority voting decision-making procedures are more likely to perform well.

Dhiauddin [27] used Complexity Metrics to predict defects. When compared to other well-known past fault indicators, such as previous changes and previous errors, complexity metrics are thought to be superior predictors of potential fault. The development process or the programme can be associated with defect density by knowing which programme is prone to flaws. The bug database is a dependable source of information concerning problems. Code that changes frequently is more likely to fail than code that remains unchanged. Machine learning approaches have a greater accuracy rate and are hence significantly more stable.

This paper contrast from the above relevant works on applying ensemble models for software

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

defect detection in many directions. This paper examines and correlates the individual classifiers performance with the heterogeneous ensemble of hybrid machine learning models.

Kangtae et.al, developed a new strategy for accurate prediction of defect detection with the help of Extreme Machine Learning approach [34]. Bandini et.al, developed a new scheme in order to predict all defects of software modules efficiently [35].

Arms, et.al, investigated how the context of models, the independent variables, and the modeling techniques influence the performance of software defects detection approaches [36]. The outcomes of this approach demonstrated that, Naïve Bayes or Logistic Regression techniques can achieve better performance [36].

Arisholm developed a model for effective detection of software defects [37]. Any defect-prone software may lead to more costly fixing activities, after it is delivered to the customers. Again, detecting a non-defective module as defect-prone may unnecessarily increase the workload of testing team. The second scenario is more efficient as compared to the first one [37].

Arisholm, et.al, stated that more complex methods like SVM perform less compared to the naive Bayes (NB) or logistic regression [37]. The overall performance of SVM is completely dependent upon a specific kind of kernel. Linear kernels are very simple and generally perform better when simple datasets are considered. But these are inefficient in case of non-linear data sets. Apart from this, RBF kernels are considered as most complicated ones. But such kernels are much better during the learning of non-linear relationships. This technique is not quite efficient in case of linear and smooth datasets. Additionally, in case of skewed data, the above technique shows poor performance. In case of balanced datasets, significant performance can be noticed [37].

3. HETEROGENEOUS ENSEMBLE METHODOLOGY

Classification algorithms are used to create prediction models, and there are a variety of methods available. According to the literature there is no single "super" classification algorithm that delivers the best results in all instances (datasets). The "ensemble" classification method employs a set of classifiers to predict class labels.

There are two types of ensembles: homogeneous ensembles and heterogeneous ensembles. All of the classifiers in a homogenous ensemble are created using the same classification algorithm. The classifiers that make up the heterogeneous ensemble are created by combining different classification techniques.

Heterogeneous ensemble methodology consolidates at least two individual machine learning algorithms. For example neural network with support vector machine system in a hybrid neurovector system. Heterogeneous models are defines as any adequate consolidation of machine learning approach in sequential or parallel manner that works and produces more accuracy compare to the simple machine learning technique.

The main challenges of heterogeneous ensemble model are efficiency of each algorithm, acceleration of process and the time require in developing a hypothesized high-performance decision model. In this paper we have used Pearson Correlation Feature selection technique along with the different machine learning algorithm results in Heterogeneous ensemble model. The objective of using PCF is to perform feature reduction while protecting the randomness in the high-dimensional space.

3.1 Feature Selection

The number of input variables should be reduced to lower the computational cost of modelling and, in some situations, to increase the model's performance.. There are two basic strategies for measuring software in current research: extracting and selecting the characteristics. Feature extraction is a technique for generating new characteristics from an original set of attributes by altering or combining them. And choice of features is a strategy that uses repeated selection and search tests to select a subset of the most important software quality characteristics from an initial set of characteristics. Filtering and wrapping are two often used selection strategies.

Some scholars believe that a single approach of selecting attributes can lead to an optimal location. As a result, strategies like ensemble technology, which combines multiple selection techniques rather than a single method, and an iteration strategy that repeatedly re-examples the

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

characteristics might be pushed. Other techniques, such as correlation assessment, logistic regression, and mutual information analysis, are also used to calculate software metrics[28].

During a statistical analysis reduction of number of features can possibly leads to many benefits like:

- Improvement in accuracy
- Reduction of over fitting problem
- Less time for training process
- Improvement in data visualization
- Increase in process of conveying of our model

Analysing a prototype group of features from that a classification model is constructed for a precise work is a main problem in machine learning. The main hypothesis is that best feature sets contains features that are extremely associated with the class [5]

PCF promptly determines and screams irrelevant, redundant and noisy features, and finds relevant features as long as their relevance does not shortly build up on other features [6]. For feature X with values x and classes Y with y, where X,Y are treated a random variables, Pearson Correlation Coefficient is defined as

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}} \quad (1)$$

r= ±1 if x and y are linearly dependent and zero if they are absolutely uncorrelated.

3.2 Heterogeneous Ensemble Model

An ensemble methodology, in which huddle of machine learning algorithms are combined and their results acting as an individual decision maker. Every machine learning algorithm has its own constraints and various learning algorithms are applicable for distinct problems[7]. The major assumption of ensemble model is, the result of individual learners are combined to produce better results which in turn improves the performance of the overall system [7]. Numerous studies were conducted and reported that ensemble learning models accuracy importantly surpass the single learning models. The main agenda of ensemble model is managing every single models strengths and weakness automatically, leading to the best achievable decision being taken overall [8].

Ensemble methods use a combination of models to increase accuracy [20]. Combine a series of k learned models M_1, M_2, . . . M_n with the aim of creating an improved model M^* [20]. Popular ensemble methods are :

- Bagging: average the prediction over a collection of classifiers
- Boosting: weighted vote with a collection of classifiers
- Ensemble: combining a set of heterogeneous classifiers.

Heterogeneous ensemble model contains of members having distinct learning algorithms such as SVM, ANN and Random Forest are of not same type. Such classifier models are also known as hybrid ensemble classifiers. These heterogeneous ensemble classifiers are associated with challenges like:

- Selection
- Performance
- Combining

Selections of base classifier need to be consider, and selecting the collection of classifier which will best classify dataset is a difficult task [9]. Performance of these classifiers differs from one data set to another dataset. The performance of the ensemble classifier to be better than that of the single classifier and tis can be done by effective combining the base classifier [10].

In this work we have combined each of three different machine learning models. Then finally, the majority voting method is used to finalize the results of the combined models. The three different base learners which are combined in our heterogeneous model are SVM, ANN and Random Forest[11].

3.2.1 Support vector machine (SVM)

The SVM is selected as one among the learning algorithms to train the model, as it is best applicable for binary classification and so far shown best results in the Optical character recognition, fore casting, electric load, medical diagnostics and other fields [12][13].

3.2.2 Artificial neural networks (ANN)

ANN can be defined as an especially simplified model of the brain cells that coordinate with each other to perform the required function. ANN can be used for different task, such as classification, noise reduction and prediction [14]. One of the main advantages of the ANN is the chance to recover hidden information that allows solving complex problems [14]. Another advantage of the neural

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

networks is the ability to generalise and produce both linear and non-linear outputs [14].

3.2.3 Random Forest (RF)

There are various types of RF's that are distinguished by the way every single tree is constructed, the strategy applied to produce the modified data sets, and the way the predictions of every single tree are accumulated to produce a distinctive prediction [15]. The RF becomes a considerable analysis tool in dissimilar fields, exceptionally in bio-informatics and different experiments shows that conventionally results of Random Forests are absolutely good [15].

4. METHODOLOGY

In this paper open source different types of projects developed by using different programming languages and with different domain have been selected and are collected form PROMISE Software Engineering repository. The dataset considered for the experimental word contains Object-Oriented software metrics and traditional software metrics.

4.1 DataSets

One possible concern is that a classifier that makes use of poorly defined attributes will be difficult to interpret and reproduce in another setting [16]. As a result, the datasets selected for this study are limited to those that contain clearly specified attribute measures.

Specifically, we look at datasets that make use of Halstead [20] and McCabe [21] metrics. Both McCabe and Halstead metrics consist of well defined measurements and calculations that are easily reproducible [25]. These specific metrics are a natural choice due to the fact that they have previously been used for software defect prediction [26].

In diverse domain applications, sophisticated databases are implemented in heterogeneous databases or homogeneous databases. With the rise of the software industry, more and more software companies are concerned about software quality control and process optimization [29].

Both dependent and independent variables are available in the defective dataset what we selected for our work. Each datasets contains set of instances and each instance contains set of features and dependent class label which indicates whether instances defective or non defective [17]. The object oriented defect data set what we selected for our work contains the software metrics which are shown in the Table 1.

Table 1: Object-OrientedSoftwareMetrics

wmc	rfc	Npm	mfa
dit	locm	icom3	cam
Noc	ca	Loc	ic
Cbo	ce	Dam	cbm
max_cc	avg_cc	Moa	amc

Other datasets what we selected for our work contains the traditional software metrics which are shown in Table 2.

Table 2: Traditional Software Metrics

LOC_BLANK	CYCLOMATIC_COMPLEXITY	HALSTEAD_CONTENT	HALSTED_LEVEL
BRANCH_COUNT	DESIGN_COMPLEXITY	HALSTED_DIFFICULTY	HALSTED_PROG_TIME
LOC_CODE_AND_COMMENT	ESSENTIAL_COMPLEXITY	HALSTED_EFFORT	HALSTED_VOLUME
LOC_COMMENTS	LOC_EXECUTABLE	HALSTED_ERROR	NUM_OPERANDS
NUM_UNIQUE_OPERATOR	NUM_UNIQUE_OPERAND	HALSTED_LENGTH	NUM_OPERATORS

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

4.1 Proposed Model

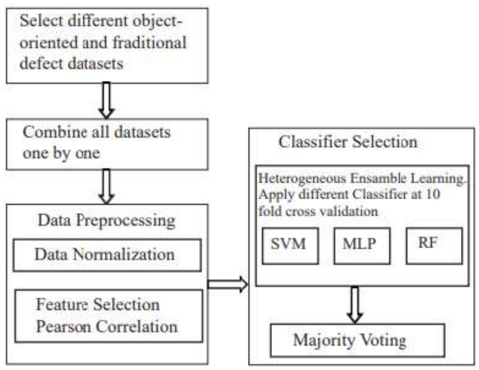

Figure 1: Proposed Model

Figure1 depicts our proposed model. In our proposed model we selected two types of data sets. The data set contains object-oriented defect data and traditional defect data. Data sets are combined one by one and are applied to pre-processing technique. The pre-processing step contains data normalization and feature selection technique. To scale the values normalization is applied on the attributes.

The accuracy of the model improved if the model is trained with the best features. In order to select best features from the data set Person Correlation Feature selection technique is applied. Different classifiers at 10-corss validation are applied to classify the instances are defective or non-defective. Heterogeneous ensemble method is used. The results from the different classifiers are decided by using majority voting method.

iTo forecast the software detection process, an ensemble learning model is created from a number of base classifiers. Ensemble models are created in our model utilising a variety of base classifiers, a process known as heterogeneous ensemble modelling. The suggested ensemble learning model is used to analyse multiple software flaws with a high number of features.

Each software measure is filtered using the feature selection technique, and several types of base classifiers like Support Vector Machines, Artificial Neural Networks, and Random Forest are employed in our model to compare the performance of the proposed model to traditional models. Models become more accurate and faster when the best feature from a feature set is chosen.

4.2 Proposed Algorithm

Algorithm of the proposed heterogeneous software defect detection model is as follows:

Input: A set of datasets (D1, D2….Dn), a set of metrics (M1, M2, ..., Mm), set of classifiers (C1,C2,…Ck)

Output: Final averages accuracy of the proposed model

Algorithm Steps:

Step1: Select different defect datasets (D1, D2,….Dn)

Step2: Combine all the datasets one by one $D=\sum_{i=1}^{n} Di$

Step3: Pre-processing step

3.1: apply data normalization on the datasets to scale the values of an attribute so that it falls in a smaller range. Minimum and Maximum values from data is calculated and each value is replace according to the following formula.

$$v' = \frac{v - min_A}{max_A - min_A}(new\_max_A - new\_min_A) + new\_min_A \quad (3)$$

3.2 apply Pearson Correlation Feature selection technique to select the best features from the data set

Step4:iselectitheisetiofimetricsi$M_s$ifromistep3i

Step5:iapplyidifferenticlassifieriC=(C_1,C_2,…C_k)iusingi$M_s$ioni$D_v$iati10-Crossivalidationi

Step6:iCalculateiaccuracyiA_{ii}ofiC_{ii}oni$D_v$iwhereii=1, i2,i3,….k

Step7:icalculateifinalidecisionibyiusingimajorityivotingitechnique

In our ensemble machine learning model multiple models are combined and the predictions from these multiple models are finalized by using majority voting method [19]. Majority voting technique that can be used to boost-up the model performance, compare with the any single model used in the ensemble model. This technique can be used in both classification and also in regression [17]. In classification, the predictions are collected from each label are summed and the label with the majority vote is predicted. It results in a lower variance than any model used in the ensemble [13][18].

$$c(x): \underset{i}{arg\ max} \sum_{j=1}^{B} w_j \hat{p_{ij}} \qquad (2)$$

p_{ij} is the probability estimate from the j^{th} classification rule for the i^{th} class

4.3 Experimental Results

The model is trained and tested on validated data set. Table 3 depicts the results of our model with object oriented defect dataset. Accuracy of the single learning model with heterogeneous learning model is compared and it is observed that our model is giving improved results.

The evaluation results show that our proposed model achieved up to 98% accuracy which is more than the evaluation accuracy achieved by individual learning models. This is achieved by using heterogeneous ensemble model methodology, in which three individual learners are combined which strengthened the overall proposed model. Before training the model the best features are selected by using feature selection technique like Pearson Correlation Feature selection technique.

In this part we tested our suggested model on NASA and Promise software defect datasets and compared the results to traditional defect prediction models. The NASA Metric Data Program is open to the public and can be used to evaluate, test, and improve software engineering predictive models. The dataset consists of the McCabe and Halstead features extractions of the code. The measures are module based.

The probability of detection is related to the effort; consequently, a higher rate of detection necessitates a greater amount of effort. Each module has a defective or non-defective output mark that indicates whether errors in the respective modules have been detected [30].

In comparison to traditional software defect prediction models, experimental data reveal that the suggested model has a high defect detection rate [31].

Table3: Performance analysis of proposed approach with traditional methods using defect prediction rate

Algorithm	Ant	Camel	Jedit	Integrated	lucene	Poi
ProposediModel	0.925	0.965	0.984	1.001	0.991	1.001
SVM	0.891	0.823	0.831	0.842	0.91	0.871
ANN	0.824	0.832	0.871	0.892	0.824	0.891
RandomiForest	0.841	0.871	0.863	0.852	0.824	0.823

IIn Fig 2 it can be seen that our proposed heterogeneous ensemble model producing better accuracy when compare with the single machine learning model.

With the proposed ensemble heterogeneous ensemble model, the accuracy rate improves by an average of 15%. In order to estimate the defect rate in the training data, the suggested algorithm
uses the majority voting approach. As a result, the Integrated and POI datasets outperformed other defect datasets in terms of accuracy.

Table 4 depicts the results of our model with traditional defect dataset. For this we selected some defect data sets which contain McCabe's cyclomatic complexity metrics, Lines of Code (LOC) metrics and Halsted metrics

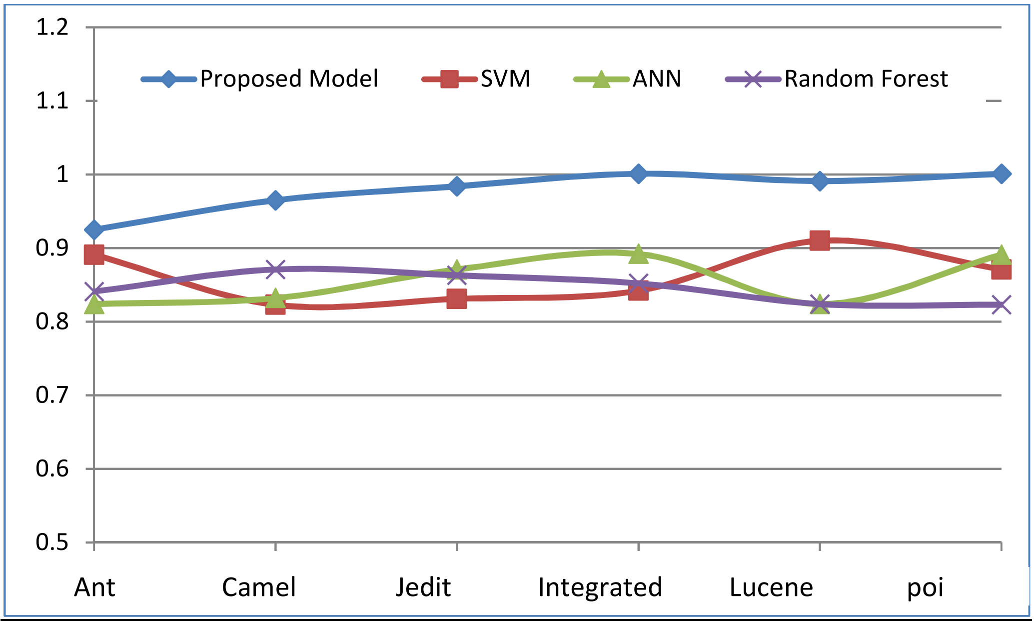

Figure 2: Performance of the proposed model with existing models

Table 4: Accuracy comparison of our model with traditional single learning models

Algorithm	PC1	JM1	KC1	MC1	MW1	PC30
ProposediModel	1.023	0.962	0.975	1.023	0.981	0.972
SVM	0.914	0.822	0.847	0.977	0.897	0.882
ANN	0.925	0.819	0.854	0.975	0.871	0.87
RandomiForest	0.913	0.926	0.86	0.976	0.897	0.881

Classification Accuracy is what we usually mean, when we use the term accuracy [12]. It is the ratio of number of correct predictions to the total number of input samples [13].

Accuracy= Number of correct defect predictions/ Total number of defects

We compared accuracy what we achieved from our proposed heterogeneous ensemble model with single machine learning model. In figure 3 it can be seen that our model is producing better accuracy

5. CONCLUSION AND FUTURE SCOPE

Software defect prediction is a necessary step before creating high-quality software defect classification models [32]. In both static and dynamic software metrics, ensemble classification models are more effective in detecting software defects. The main goal of this paper is to experiment and asses the heterogeneous ensemble classifier model by calculating the proposed model accuracy on various datasets.

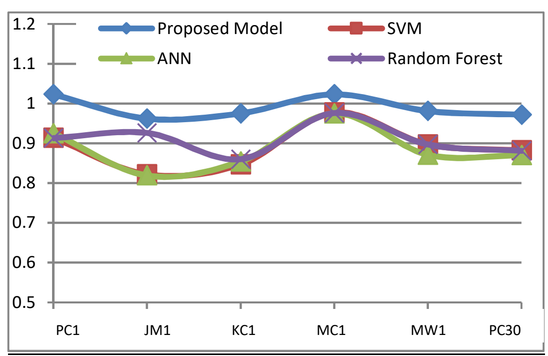

Figure 3: Accuracy comparison our proposed model with single learning models

It is finalized from all the discussed in the paper that at prediction stage some datasets producing similar prediction accuracy results some are giving different results in terms of accuracy [33]. This type of different in the

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

accuracy is due to the different software (datasets) that have been selected for our study. The software which are selected have been prepared in various environments, by different team members in terms of their expertise, and skills by different organization. The conclusions made in our investigation are encouraging and more experiments are indispensable to sketch any specific pattern. This task can be done in the future in software modules that are implemented from web apps created by a large number of users.

REFERENCES

[1] IssamH.laradji, Mohammad Alshyeb, LahouariGhouti, "Software defect prediction using ensemble learning on selected features", *Inform.Sfotw.Technol*, 2014

[2] Abdullateef O. Balogun, Amos O. Bajeh,Victor A. Orie and Ayisat W. Yusuf-Asaju, "Software Defect Prediction Using Ensemble Learning: An ANP Based Evaluation Method",*FUOYE Journal of Engineering and Technology*, Vol. 3, No. 2, September 2018.

[3] Mohammad Zubair Khan, "Hybrid Ensemble Learning Technique for Software Defect Prediction", *I. J. Modern Education and Computer Science*, 2020.

[4] Abdullah Alsaeedi, Mohammad Zubair Khan, "Software Defect Prediction Using Supervised Machine Learning and Ensemble Techniques: A Comparative Study", *Journal of Software Engineering and Applications*, Vol. 12, 2019, pp. 85-100

[5] Prasanth Y., Sreedevi E., Gayathri N. and Rahul A.S., "Analysis and implementation of ensemble feature selection to improve accuracy of software defect detection model", *Joiurnal of advanced research in Dynamical and Control Systems*, Vol 9, No. 18, 2017, pp.601-613

[6] E. Chandra Blessie, E. Karthikeyan, "Sigmis: A Feature Selection Algorithm Using Correlation Based Method", *Journal of Algorithms & Computational Technology* Vol. 6, No. 3,2012

[7] Ran Li, Lijun Zhou, Shudong Zhang, Hui Liu,Xianhgyang Huang, Zhong Sun, "Software defect prediction based on Ensemble learning", *Proceedings of the 2019 2nd International Conference on Data Science and Information Technology*, July 1-6, 2019

[8] Sivakumar. S, SoumyaRanjanNayak, Ashok Kumar, S. Vidyanandini, GopinathPalai, "An empirical study of supervised learning methods for Breast Cancer Diseases", *International Journal for Light and Electron Optics* , Vol. 175, 2018, pp. 105-114

[9] Sivakumar S, Sreedevi E, Premalatha V, Haritha D, "Parallel Defect Detection Model on Uncertain Data for GPUs Computing by a Novel Ensemble Learning", *Applications of Artificial Intelligence for Smart Technology*,2020, pp. 146-163

[10] ShaikRazia, "A Comparative study of machine learning algorithms on thyroid disease prediction", *International Journal of Engineering and Technology (UAE)*, Vol. 7(2.8), 2008, pp. 315-319

[11] Sivakumar. S, Ganesan. P,andSundar. S. "A MMDBM Classifier with CPU and CUDA GPU computing in various sorting procedures", *International Arab Journal of Information Technology*, Vol. 14, No.7, 2017,pp. 897-906

[12] XiaotaoRong, Feixiang Li, Zhihua Cui, "A model for software defect prediction using support vector machine based on CBA", *Int. J. Intelligent Systems Technologies and Applications*, Vol. 15, No. 1, 2016

[13] Rajesh Kumar. E and K.V.S.N. Rama Rao. "Suicide Prediction in Twitter Data using Mining Techniques: A Survey", *International Conference on Intelligent Sustainable Systems (ICISS 2019)*, 2020,pp.122-131

[14] SaiqaAleem , Luiz Fernando Capretz andFaheem Ahmed, "Benchmarking machine learning techniques for software defect detection", *International Journal of Software Engineering & Applications (IJSEA)*, Vol.6, No.3, May 2015

[15] Yan Naungsoe, Paulus InsapSantosa, Rudy Hartanto, "Software defect prediction using Random Forest algorithm", *12th South East Asian Technical University Consortium (SEATUC)*,2018

[16] Sreedevi, E, PremaLatha V, Sivakumar. S. "A Comparative Study on New Classification Algorithm using NASA MDP Datasets for Software Defect Detection", *2nd International Conference on Intelligent Sustainable Systems (ICISS 2019)*, 2019, pp.312-317

[17] David Bowes, Tracy Hall, Jean Petric," Software defect prediction: do different

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

classifiers find the same defects?", *Software Quality Journal,* Vol. 26, 2018, pp. 525–552

[18] ShaikRazia, "Machine Learning Techniques for Thyroid Disease Diagnosis: A Systematic Review, Modern Approaches in Machine Learning and Cognitive Science: A Walkthrough". *Studies in Computational Intelligence*,2020, pp. 885

[19] Sreedevi E, Premalath V, Prasanth Y, Sivakumar S, "A Novel Ensemble Learning for Defect Detection Method With Uncertain Data", *Applications of Artificial Intelligence for Smart Technology*, 2020, pp.67-79

[20] Raja KhurramShahzad, NiklasLavesson, "Comparative Analysis of Voting Schemes for Ensemble-based Malware Detection", *Journal of Wireless Mobile Networks, Ubiquitous Computing and Dependable Applications*, Vol. 4, No. 1, 2012, pp. 98-117

[21] Rawat M.S., Dubey S.K.,"Software defect prediction models for quality improvement: A literature study". *IJCSI Int. J. Comput. Sci*. No. 9, 2012, pp. 288–296.

[22] Catal C., Sevim U.,Diri, B. "Practical development of an Eclipse-based software fault prediction tool using Naive Bayes algorithm". *Expert Syst*. *Appl.*Vol. 38, 2011, pp. 2347–2353.

[23] Son, L.H., Pritam, N, Khari M., Kumar R., Phuong P.T.M., Thong P.H. "Empirical Study of Software Defect Prediction: A Systematic Mapping". *Symmetry 2019*, 11, 212.

[24] Challagulla V.B., Bastani F.B., Yen I.L., Paul, R.A. "Empirical assessment of machine learning based software defect prediction techniques". *Int. J. Artif. Intell. Tools,*Vol. 17, 2008, pp.389–400.

[25] Huda S.,Alyahya S., Ali M.M., Ahmad S., Abawajy J., Al-Dossari H., Yearwood J. "A Framework for Software Defect Prediction and Metric Selection". *IEEE Access 2018*, Vol. 6, pp. 2844–2858,2018.

[26] 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.

[27] M. Dhiauddin, M. Suffian, and S. Ibrahim, "A Prediction Model for System Testing Defects using Regression Analysis," Int. J. Soft Comput. Softw. Eng., vol. 2, no. 7, pp. 55-68, 2012.

[28] Cong Jin and Shu-Wei Jin, "Prediction approach for fault-proneness based on hybrid artificial network and quantum particle swarm optimization", Applied Soft Computing, Vol. 35, pp. 717-725, 2015.

[29] G. Fung, S. Sandilya, And R.B. Rao , "Rule Extraction Form Linear Support Vector Machines", Proceedings Of The Eleventh International Conference O Knowledge Discovery In Data Mining, pp. 32-40,2005.

[30] 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.

[31] 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.

[32] 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.

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

[34] Kangtae Kim,Hyungrok Kim,Woomok Kim: Building Software Product Line from the Legacy Systems "Experience in the Digital Audio and Video Domain", Software Product Line Conference 2007, SPLC 2007, IEEE Conferences, Kyoto, 2007, pp.171 - 180.

[35] Bandini, R. K., Vaishnavi, V. K. and Turk, D. E. 2003. Predicting Maintenance Performance Using Object Oriented Design Complexity Metrics, IEEE Transactions on Software Engineering,Vol. 29, Issue 1, pp.77-87.

[36] Ayanam, V. S., 2009. So ftware security Vulnerability Vs. Software Coupling, A Study with Empirical Evidence. Masters Thesis. Southern Polytechnique State University.

[37] Arisholm, E., Briand, L. and Johanvessen, E., 2010. A systematic and comprehensive investigation of methods to build and evaluate fault prediction model, Journal of system and software,Vol. 83,