Int J Syst Assur Eng Manag (June 2017) 8(2):334–351
DOI 10.1007/s13198-016-0438-y

ORIGINAL ARTICLE

Prediction of defect severity by mining software project reports

Rajni Jindal[1] · **Ruchika Malhotra**[1] · **Abha Jain**[1]

Received: 20 August 2015 / Revised: 1 March 2016 / Published online: 10 March 2016

Abstract With ever increasing demands from the software organizations, the rate of the defects being introduced in the software cannot be ignored. This has now become a serious cause of concern and must be dealt with seriously. Defects which creep into the software come with varying severity levels ranging from mild to catastrophic. The severity associated with each defect is the most critical aspect of the defect. In this paper, we intend to predict the models which will be used to assign an appropriate severity level (high, medium, low and very low) to the defects present in the defect reports. We have considered the defect reports from the public domain PITS dataset (PITS A, PITS C, PITS D and PITS E) which are being popularly used by NASA's engineers. Extraction of the relevant data from the defect reports is accomplished by using text mining techniques and thereafter model prediction is carried out by using one statistical method i.e. Multi-nominal Multivariate Logistic Regression (MMLR) and two machine learning methods viz. Multi-layer Perceptron (MLP) and Decision Tree (DT). The performance of the models has been evaluated using receiver operating characteristics analysis and it was observed that the performance of DT model is the best as compared to the performance of MMLR and MLP models.

✉ Ruchika Malhotra
ruchikamalhotra2004@yahoo.com

Rajni Jindal
rajnijindal@dce.ac.in

Abha Jain
me_abha@yahoo.com

[1] Department of Computer Science Engineering, Delhi Technological University, New Delhi, India

Keywords Defect prediction · Empirical validation · Machine learning · Text mining · InfoGain · Receiver operating characteristics · Statistical methods

1 Introduction

With the advent of recent advances in software technology, the software organizations are coming up with the development of new software at an exponential rate. These software are huge in size consisting of a number of lines of code, thus resulting in the introduction of defects and faults. Today, we are at a situation where the probability of the defects being introduced in the software is increasing at a tremendous rate leading to functional failures (Myers et al. 2004). This is because of an ever increasing size of the newly developed software in order to meet the demands of the people. The introduction of defects in the software is an inevitable part which we cannot avoid. Thus, the only solution which is left to us is the detection of these defects and then devising some ways and means to remove these defects from the software as early as possible. The most critical aspect of any defect being introduced in the software is its severity level. Each defect is associated with its respective severity level which plays one of the most crucial roles in characterizing it. A severity level associated with a defect is defined as an extent to which that particular defect can be harmful for the software (Singh et al. 2010). These severity levels have a broad range which may vary from mild to catastrophic. Catastrophic defects are the most severe defects which must be identified and then dealt with as early as possible in order to prevent the damage of the software any further (Emam and Melo 1999; Aggarwal et al. 2009). A failure caused by such defects could result in

the loss of a human rated system. In other words, presence of such defects may jeopardize safety and security. On the other hand, mild defects are the defects which may not result in any kind of failure but they need to be corrected for the smooth functioning of the organization. Examples of such type of defects are standards violations, maintenance issues etc.

Identifying the defects with respect to their severity levels will be very useful for an overall benefit of the organization in terms of the resources of the organization. All the testing resources and the man power of the organization can be judiciously used for the identification and removal of the defects having a higher severity level. In other words, higher severity defects are given more priority than the lower severity defects. There are a number of studies which have been carried out in the past that talk about defect prediction in general without focusing on the severity of these defects. Runeson et al. (2007) and Wang et al. (et al. 2008) have analyzed the defect reports and developed a tool that would be used to detect duplicate reports using Natural Language Processing (NLP). These defects are reported in a bug tracking system so that the cause of the defects can be found out and their corresponding remedy can be availed (Runeson et al. 2007). The defects are encountered during testing or other development activities by developer or anyone who is involved in the development of the product. Cubranic and Murphy (2004) analyzed an incoming bug report and proposed an automated method that would assist in bug triage to predict the developer that would work on the bug based on the bug description. Canfora and Cerulo (2005) discussed the methodology which can be used by developers in handling a new change request. This change request can be either in the form of a bug or an enhancement feature. Also, a lot of empirical work has been carried out in predicting the fault proneness of classes in object-oriented (OO) software systems using a number of OO design metrics (Emam and Melo 1999; Catal and Diri 2009; Gondra 2008; Gyimothy et al. 2005; Malhotra and Singh 2011; Malhotra and Jain 2012; Ohlsson et al. 1998; Olague et al. 2007; Yu et al. 2002; Zhou et al. 2010). Although all these previous studies are focused on developing an automated tool or a method that would be helpful for defect prediction, but there are only a few studies till date which focus on the identification of the defects based on their severity levels. Assigning the correct severity level to the defects is highly beneficial as this will provide an insight about the impact of the defect on the software operation and hence will help the testers in making an effective use of the available testing resources. This would in turn be of great help to software practitioners as they can now make use of the model predicted with respect to high severity defects.

In this paper, we intend to analyze the defect reports available in the PITS dataset which being popularly used by NASA's Independent Verification and Validation program (IV & V). Initially, we have used a series of text mining techniques in order to extract the relevant data from the PITS dataset and then at a later stage we have employed different machine learning and statistical methods in order to analyze the data. A series of text mining techniques used in this paper are pre-processing, feature selection, tf-idf weighting and finally normalization. In the first step of pre-processing, we have removed all the stop words like articles, prepositions etc. from the documents. This was followed by a process of stemming which is the most crucial step of pre-processing. Thereafter, an appropriate feature selector was applied in order to further reduce the size of the feature space. In the paper, we have used Info-Gain measure as the feature selection method which ranks the given words in increasing order based on the value obtained and then selects the top N features. Now corresponding to each document, we will be having N features selected using Info-Gain measure. Thus, we can say that each document of the dataset can be represented as a N-dimensional vector consisting of n values and all the vectors together forms a vector space model or VSM. Each term in the vector is weighted using tf-idf approach. Finally, all the vectors under consideration are normalized before they are submitted to the learning algorithms for the analysis. In the paper, we have employed one statistical method namely Multi-nominal Multivariate Logistic Regression (MMLR) and two machine learning methods viz. Multi-layer Perceptron (MLP) and Decision Tree (DT) in order to predict the best model that gives the highest accuracy. The results are then analyzed using the most popular performance measures viz. Area under the curve (AUC), sensitivity value and cut-off point obtained from receiver operating characteristic (ROC) curve. The main contribution of our work is to assess the severity of the defects being reported from four NASA robotic missions based on a five-point scale used to score the severities of the defects. The scale ranges from one to five which refers to the worst to dullest defects respectively.

This paper deals with the following sections: Sect. 2 reviews the key points of available literature in the domain. Section 3 describes the research method used for this study, including data source, text mining methodology, and analysis of the dataset. Section 4 presents the result analysis. Thereafter, Sect. 5 presents the discussion which summarizes the result analysis and presents the scope for future work. Finally, we have Sect. 6 which concludes the paper.

2 Literature review

Till date, there are a number of empirical studies which talk about the relationship between OO metrics and fault-proneness of classes. But, as of now, there are only a few

studies that have been carried out to study how the OO metrics and faulty classes are related to each other at varying severity of faults. Various regression and machine learning methods have been used to develop the prediction models which will be used to empirically validate the OO metrics. Highlights of select papers have been discussed in this section. The work by Singh et al. (2010) have provided the analysis of the performance of the predicted models on the public domain NASA dataset KC1 and found that the model predicted at high severity faults has lower accuracy than the models predicted at medium and low severities and that the DT and ANN models outperformed the LR model. They also concluded that CBO, WMC, RFC and SLOC metrics are significant across all severity of faults and DIT metric is not significant across any severity of faults. LCOM and NOC are not found to be significant with respect to LSF.

The paper by Zhou and Leung (2006) also ended up by concluding the similar results. They have investigated the fault-proneness prediction performance of OO design metrics with respect to ungraded, high, and low severity faults by employing statistical (LR) and machine learning (Naïve Bayes, Random Forest, and NNge) methods. The authors have too used the same dataset i.e. NASA dataset KC1. From both the above papers, it was summarized that the design metrics could predict faults with low severity better than the faults with high severity in fault-prone classes. Bayesian approach was also used by the author Pai (2007) in his work to see how software product metrics and fault proneness are related to each other.

Shatnawi and Li (2008) studied the effectiveness of software metrics using three releases of the Eclipse project. They focused on recognizing the classes which are prone to error once the software has been released in the market. They observed that, the accuracy of the model in predicting the classes kept on decreasing as they moved from one release to the other. It was finally concluded that there must be some alternate methods which should be developed apart from the use of metrics in order to find error prone classes. This will lead to higher level of accuracy.

Our work is similar to the work done by Menzies and Marcus (2008). They have presented an automated method named SEVERIS (SEVERity Issue assessment) which is used to assign severity levels to the defect reports by using the data from PITS dataset which is being popularly used by the NASA's engineers. Their method is based on extracting and then analyzing the textual data from issue reports in PITS by using various text mining techniques. They have used a rule learning method as their classification method to assign the features with proper severity levels, based on the classification of the existing reports. Similar work has also been done by Sari and Siahaan (2011). They have also developed a model for the assignment of the bug severity level. They have used the same pre-processing tasks (tokenization, stop words removal and stemming) and feature selection method (InfoGain) as have been used by Menzies and Marcus (2008) in their paper for dimensionality reduction. But, Sari and Siahaan (2011) have used SVM as their classification method in contrast to the rule learning method which does classification with the help of classification rules. Lamkanfi et al. (2010) have also analyzed the textual matter of the defect documents using text mining algorithms in order to propose a technique that is used for severity prediction using three open-source projects viz. Mozilla, Eclipse and GNOME. They have used Bugzilla for tracking the bugs and Naïve Bayes as the classifier for prediction. They have evaluated the performance of the model using ROC analysis, precision and recall.

We have used different PITS datasets of NASA and employed the similar text mining techniques to extract the relevant features of each report. Then, we have used MMLR to predict the overall defect in the system. After developing a prediction model using MMLR, we have used two machine learning techniques (MLP and DT) to predict the best model that gives the highest accuracy. In other words, we can say that MLP and DT are then used to assign these features with proper severity levels, based on the classifications of the existing reports. Finally, we have evaluated the performance of these models using ROC analysis.

3 Research methodology

In this section, we present our research method. We first introduce the data source and then describe the text mining methodology that we have used in order to extract the relevant words from the defect descriptions. Thereafter, we describe our data analysis methods, specifically, logistic regression analysis and machine learning models, and our model evaluation criteria.

3.1 Data source

In this study, we have made use of the four anonymous PITS datasets which are referred to as PITS A, PITS C, PITS D and PITS E datasets. These datasets have been supplied by NASA's Software Verification and Validation (IV & V) Program. The data in all these datasets has been collected for more than 10 years and includes all the issues that have been found in the robotic satellite missions and human rated system.

The datasets comprise of the defect reports wherein each defect report contains the description of that corresponding defect, ID of the defect, and associated severity level of the

defect. According to NASA's engineers, each defect can be categorized into one of the five severity levels that are defined below in Table 1. For the purpose of empirical study, the data cannot be fed to the machine learning tool in the categorical form and therefore, we have associated a code corresponding to each severity level as can be seen from Table 1.

Defects which fall into the category of 'Very High' severity level are the defects which may jeopardize safety and security. The recovery from such defects is impossible and failures which happen due to the occurrence of such defects may result into cascading system failures. For instance, a failure caused by such defects could result in the loss of the human-rated system or may result in the loss of flight or ground personnel. Due to these reasons, such defects are extremely rare in nature. This can be observed from Table 2 which shows the number of defect reports at each severity level corresponding to all the four PITS datasets. It can be seen from the table, that there are no defects which have 'Very High' severity level corresponding to all the four datasets. Therefore, in this empirical study, we do not take into account the severity level 1 and consider only the last four severity levels viz. severity 2 (high), severity 3 (medium), severity 4 (low) and severity 5 (very low).

We went through these datasets and extracted the description of each defect from all the reports. We then analyzed these textual descriptions and applied the text mining techniques to extract the relevant words from each report. At a later stage, statistical and machine learning methods were used to assign the severity level to each defect based on the classifications of existing reports. As we know that the standard machine learning methods work well only for the data with fewer number of attributes. Therefore, before we can apply machine learning to the results of text mining, we have to reduce the number of words, referred to as the dimensions (i.e. attributes) in the data. Hence, we applied different methods of text mining for dimensionality reduction in the following order: tokenization, stop word removal, stemming, feature selection and weighting.

Table 1 Range of the severity levels

'Categories' of the severity level	'Codes' of the severity level
Very high	1
High	2
Medium	3
Low	4
Very low	5

Table 2 The data sets being used in the case study

	Severity 1	Severity 2	Severity 3	Severity 4	Severity 5
Pits A	0	311	356	208	26
Pits C	0	0	132	180	7
Pits D	0	1	167	13	1
Pits E	0	24	517	243	41

Each cell indicates the number of defect reports corresponding to each severity level

3.2 Text mining methodology

In this sub-section we will discuss the text mining methodology which has been used in our work in order to extract the relevant data from the defect reports. Text mining is done in order to associate a given document with a particular category chosen from the available set of categories (Ikonomakis et al. 2005). The defect reports present in different PITS datasets consist of the defect ID and their corresponding defect description. We extract the defect description from the reports and then apply a series of text mining steps on to the data of defect description. This is done in order to make the description in the form which can be used by the learning algorithms for performing the data analysis. The text mining steps have been applied in the following order: pre-processing, feature selection, tf-idf weighting and finally normalization as depicted below in Fig. 1. We will be explaining each of these steps one by one in the subsequent sub-sections.

3.2.1 Pre-processing

Pre-processing is the first step of text mining process, wherein the aim is to remove the irrelevant words form the document. Irrelevant words are those words which are not important for the analysis of the data and rather their inclusion in the document can degrade the performance of the learning algorithms. Removal of such words from the document leads to the reduction of feature set space, thereby making it easier for the learning algorithms to perform data analysis. Pre-processing consists of three basic steps viz. tokenization, stop-words removal and stemming. Initially, tokenization is done which replaces all the punctuations with blank spaces, removes all the non-printable escape characters and converts all the words to lowercases. Thereafter, all the stop words like prepositions, articles, conjunctions, verbs, pronouns, nouns, adjectives, adverbs etc. are removed from the document. Finally, stemming is performed which is the most crucial step. Here, all the words which have a common stem are replaced and the stem is retained as the selected feature. For example, the words like "take", "takes", "took" and "taking" can be replaced with a single word as "take". All

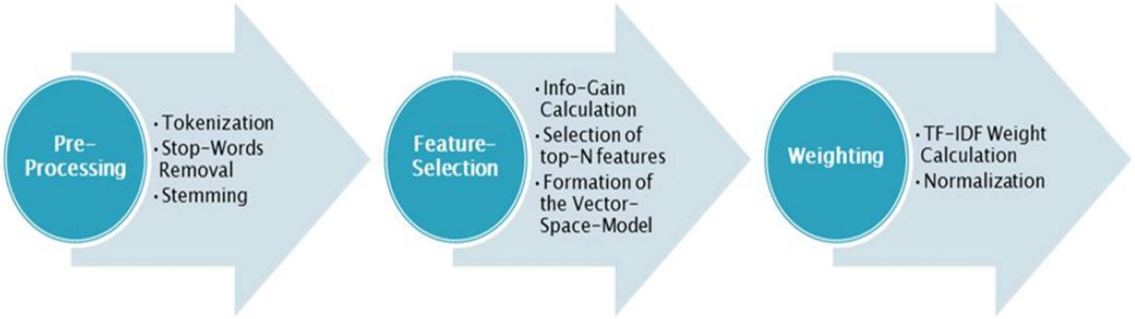

Fig. 1 Steps involved in text mining methodology

the words obtained after preprocessing were called as 'features'.

3.2.2 Feature selection

Feature selection is the second step of text mining process, wherein the aim is to further reduce the size of the initial feature set obtained after pre-processing. In this step, top few words are selected from the entire list of words which is obtained after pre-processing. These top words form a feature space and are then used by the learning algorithms for performing data analysis. A number of feature selection methods are available in the literature like document frequency, term frequency, mutual information, information gain, odds ratio, χ^2 statistic and term strength. We have used Info-Gain measure as the feature selection method.

Info-Gain measure works by firstly assigning a weight (score) to each of word or feature obtained after pre-processing, then ranking all the words based on their scores, and finally selecting the top 'N' scoring features. In other words, it is based on the concept of discarding the lower weighted terms, thereby reducing the size of the initial feature space.

To understand the concept of Info-gain, let us consider an arbitrary class distribution as C_0. Now, the total number of bits required to code C_0 is $H(C_0)$. It is given by the following formula:

$$N = \sum_{c \in C} n(c) \tag{1}$$

$$p(c) = {}^{n(c)}/_{N} \tag{2}$$

$$H(C) = -\sum_{c \in C} p(c) \log_2 p(c) \tag{3}$$

Now, suppose A is a group of attributes, then the total number of bits needed to code a class once an attribute has been observed is given by the following formula:

$$H(C|A) = -\sum_{a \in A} p(a) \sum_{c \in C} p(c|a) \log_2 (p(c|a)) \tag{4}$$

Now, the attribute which obtains the highest information gain is considered to be the highest ranked attribute which is denoted by the symbol A_i.

$$\text{Infogain}(A_i) = H(C) - H(C|A_i) \tag{5}$$

Tables 3, 4, 5 and 6 show the top-100 words corresponding to PITS A, PITS C, PITS D and PITS E dataset respectively obtained on the basis of ranking done by using InfoGain measure. From these tables we can also get the top-5, top-25 and top-50 words corresponding to each of the above mentioned datasets. We have analyzed the performance of one statistical method (MMLR) and machine learning methods (DT and MLP) on each set of these features by calculating area under the ROC curve.

3.2.3 Tf-Idf weighting

Here, each document is represented in the form of a vector consisting of the weights of the 'N' words which were selected by Info-Gain measure. There are a number of methods which are used for weighting the terms, but we have used tf-idf weighting method. This method is the product of two terms viz. tf and idf. The first term is referred to as the term frequency (tf) which measures the frequency of a given term 't_i' in the document 'd_j'. The second term is known as inverse document frequency (idf) which makes the terms that are rare across a group of documents more important than the others.

Now, Tf-idf of the term 't_i' in the document 'd_j' is given by the following formula:

$$TFIDF = f_i * \log\left(\frac{n}{n_i}\right) \tag{6}$$

where, f_i = Frequency of the term 't_i' in document 'd_j', n_i = Total number of documents containing the term 't_i', n = Total number of documents.

Likewise, tf-idf weighting of all the 'N' words present in each document is computed. Finally, all the vectors are

Table 3 The top-100 terms in PITS A dataset, sorted by InfoGain

Rank	Words	Rank	Words	Rank	Words	Rank	Words
1	requir	26	rvm	51	includ	76	trace
2	command	27	obc	52	specifi	77	symbol
3	softwar	28	lsrobc	53	set	78	differ
4	srs	29	system	54	projecta	79	ground
5	lsfs	30	telemetri	55	text	80	interrupt
6	test	31	code	56	time	81	reset
7	flight	32	execut	57	task	82	design
8	line	33	number	58	column	83	document
9	referenc	34	variabl	59	access	84	question
10	engcntrl	35	initi	60	refer	85	safe
11	mode	36	issu	61	uplink	86	sourc
12	file	37	provid	62	monitor	87	attitud
13	messag	38	locat	63	perform	88	door
14	script	39	section	64	paramet	89	lead
15	error	40	control	65	fp	90	contain
16	data	41	memori	66	note	91	event
17	oper	42	verifi	67	capabl	92	appear
18	spacecraft	43	indic	68	fsw	93	process
19	link	44	address	69	ivv	94	current
20	ac	45	sequenc	70	vm	95	lsrvml
21	defin	46	lsrup	71	rate	96	checksum
22	function	47	point	72	list	97	engin
23	state	48	cdh	73	case	98	load
24	valu	49	fault	74	check	99	support
25	tabl	50	specif	75	flexelint	100	transit

normalized before they are being submitted to the learning algorithms.

3.3 Data analysis

This sub-section provides an overview of the statistical and machine learning methods which have been employed in this study to predict the model. This classification model so predicted takes the severity of defects (2, 3, 4 or 5) as the dependent variable and the relevant words (i.e. features) extracted from all the reports based on Info Gain as the independent variables. This section also elaborates on the different performance measures which are used to empirically examine the predictiveness of the classification model in order to evaluate whether or not the model is useful for predicting the defects according to their severity levels.

3.3.1 Multi-nominal multivariate logistic regression

In this paper, we have used MMLR method which is a variant of logistic regression method. Logistic regression has been one of the most popular statistical methods being used in the literature so far. In MMLR method, all the independent variables are together used to predict the different categories of the dependent variable (Hosmer and Lemeshow 1989). In other words, here the dependent variable is not binary and can have more than two values. For instance, the dependent variable is our study is the defect severity which can have four possible values of the severity levels viz. high, medium, low and very low coded as 2, 3, 4 and 5 respectively. Now, in order to predict the model using MMLR, all the independent variables are used together on each of these abovementioned categories. This is in contrast to UMR model (Univariate Multinominal Regression), wherein each independent variable is used at a time in order to predict each category of the dependent variable. Much similar to UMR and MMLR models, we have two additional models viz. Univariate Logistic Regression (ULR) and Multivariate Logistic Regression (MLR) models wherein the dependent variable is of binary type. In MLR model, all the independent variables are used together in order to predict the binary dependent variable and in ULR model, the effectiveness of a single independent variable is examined at a time.

Table 4 The top-100 terms in PITS C dataset, sorted by InfoGain

Rank	Words	Rank	Words	Rank	Words	Rank	Words
1	requir	26	mode	51	channel	76	time
2	state	27	artifact	52	includ	77	transmitt
3	fsw	28	pip	53	rate	78	ac
4	command	29	instrument	54	comm	79	accuraci
5	specif	30	verif	55	orbit	80	addit
6	trace	31	initi	56	safehold	81	sband
7	tim	32	provid	57	store	82	case
8	smrd	33	traceabl	58	test	83	design
9	perform	34	oper	59	downlink	84	mention
10	glori	35	child	60	tps	85	asec
11	ground	36	satellit	61	communic	86	cloud
12	sc	37	capabl	62	detail	87	maintain
13	document	38	rqts	63	launch	88	moc
14	spacecraft	39	s919er2342	64	power	89	revis
15	icd	40	valid	65	collect	90	safe
16	spec	41	configur	66	soh	91	attitud
17	softwar	42	scienc	67	card	92	flight
18	parent	43	list	68	point	93	maximum
19	telemetri	44	packet	69	rout	94	degre
20	matrix	45	refer	70	compon	95	electr
21	mu	46	ap	71	realtim	96	enter
22	data	47	control	72	subsystem	97	exit
23	cdh	48	mechan	73	virtual	98	normal
24	system	49	interfac	74	fpga	99	condit
25	referenc	50	adac	75	mission	100	error

The general MLR model is based on the following equation (Aggarwal et al. 2009):

$$Prob(Y1, Y2, \ldots, YN) = \frac{e^{B_0+B_1Y_1+\cdots+B_nY_n}}{1+e^{B_0+B_1Y_1+\cdots+B_nY_n}} \quad (7)$$

where, Yi's are the independent variables, Bi's are the regression coefficients corresponding to Yi's and Prob is probability that a defect was found during validation.

3.3.2 Multilayer perceptron

It is one of the most popular algorithm which is used for supervised classification. It is responsible for mapping a set of input values onto a set of appropriate output values. The most elementary block used in its architecture is the 'Adaptive Linear Element' or Adaline which refers to an adaptive threshold logic element (Widro and Lehr 1990). MLP technique is based on the concept of back-propagation. Back propagation is a type of learning procedure which is used to train the network. It comprises of two passes viz. forward pass and a backward pass. In the forward pass, the inputs are fed to the network and then the effect is propagated layer by layer by maintaining all the weights of the network as fixed. In the backward pass, the updation of weights takes place according to the error computed. An error is defined as the difference between the expected output and the actual output. This error is then fed back to the system, thereby making the parameters of the system to adjust in an orderly fashion. This process is repeated over and over again until the desired performance is achieved.

3.3.3 Decision tree

The entire decision tree is traversed during classification beginning from the root of the tree until a leaf node is reached (Porter and Selby 1990). Each non-terminal node is associated with an independent variable and each terminal node (leaf node) is associated with a classification value. Chi squared Automatic Interaction Detection (CHAID) algorithm is used in the DT method. CHAID works at each step by choosing the independent variables which are most effecting in determining the dependent variable. The categories corresponding to each predictor are combined if they are not majorly different with respect to the dependent variable.

Table 5 The top-100 terms in PITS D dataset, sorted by InfoGain

Rank	Words	Rank	Words	Rank	Words	Rank	Words
1	requir	26	field	51	switch	76	default
2	trace	27	note	52	autonom	77	perform
3	avion	28	point	53	detail	78	photomet
4	data	29	fault	54	logic	79	provid
5	spacecraft	30	kav	55	comment	80	column
6	appear	31	code	56	exist	81	dfm
7	downward	32	design	57	store	82	enabl
8	flight	33	ffi	58	line	83	util
9	cadenc	34	sc	59	period	84	acquir
10	projectd	35	protect	60	apertur	85	collater
11	softwar	36	set	61	ground	86	detect
12	command	37	fpa	62	segment	87	function
13	fsw	38	initi	63	status	88	number
14	mode	39	engin	64	violat	89	singl
15	process	40	fgs	65	attitud	90	array
16	long	41	pixel	66	control	91	mask
17	implic	42	csc	67	defin	92	messag
18	target	43	hardwar	68	enter	93	packet
19	andor	44	tabl	69	support	94	test
20	implement	45	method	70	error	95	time
21	spec	46	safe	71	respect	96	bin
22	capabl	47	short	72	addit	97	ccd
23	cpp	48	oper	73	call	98	channel
24	manag	49	file	74	check	99	class
25	case	50	statement	75	collect	100	document

3.3.4 Model evaluation

The defect prediction results can be categorized into four different types as defined below (Jiang et al. 2008):

- True Positive (TP)—Refers to the number of correctly predicted positive instances
- False Negative (FN)—Refers to the number of incorrectly predicted positive instances
- False Positive (FP)—Refers to number of incorrectly predicted negative instances
- True Negative (TN)—Refers to number of correctly predicted negative instances

To measure the performance of the predicted model, we have used the following performance evaluation measures:

a. *Sensitivity*: Sensitivity is defined as the ratio of correctly classified positive instances to the total number of actual positive instances. It is also referred to as Recall. If we get a sensitivity value of 1.0 for a particular class C, then this means that all the instances which belong to class C were correctly classified as belonging to class C. Sensitivity is given by the following formula:

$$\text{Sensitivity or Recall(Rec)} = \frac{\text{TP}}{\text{TP} + \text{FN}} * 100 \quad (8)$$

b. *Receiver Operating Characteristics (ROC) Curve*: One of the most important characteristic of the ROC analysis is the 'ROC curve'. ROC curve is the visual representation which is used to picture the overall accuracy of the prediction model. It is represented by a graph together with a diagonal line. The diagonal line represents a random model that has no predictive power. The x-axis of the graph represents the false-positive rate with values between 0 (no false positives) and 1 (100 % false positives). It is often referred to as 1- specificity. The y-axis of the graph represents the true -positive rate with values between 0 (no true positives) and 1 (100 % true positives). It is referred to as sensitivity. So, we can say that ROC curve is a plot of the true positive rate against the false positive rate at different possible cut-off points (Emam et al. 1999b). We can also interpret the curve by saying that there is a trade-off between sensitivity and specificity in the sense that any increase in the value of sensitivity will lead to a decrease in the value of specificity. The main objective of constructing ROC curves is to obtain the

Table 6 The top-100 terms in PITS E dataset, sorted by InfoGain

Rank	Words	Rank	Words	Rank	Words	Rank	Words
1	requir	26	defin	51	check	76	posit
2	file	27	case	52	return	77	spacecraft
3	function	28	l4	53	enabl	78	start
4	inst5	29	verifi	54	rev	79	actuat
5	softwar	30	vm	55	implement	80	mission
6	inst6	31	design	56	unsign	81	int
7	test	32	l5	57	list	82	chart
8	inst4	33	ra	58	provid	83	evr
9	fsw	34	time	59	releas	84	line
10	project	35	capabl	60	status	85	convent
11	srs	36	interfac	61	condit	86	oper
12	command	37	trace	62	assign	87	rqmts
13	data	38	flight	63	tabl	88	switch
14	state	39	level	64	flow	89	clear
15	set	40	icd	65	word	90	inst6a
16	document	41	jpl	66	sw	91	hardwar
17	variabl	42	issu	67	ccd	92	fail
18	artifact	43	read	68	mode	93	lander
19	ep	44	power	69	messag	94	compon
20	code	45	number	70	descript	95	init
21	control	46	cmd	71	septemb	96	version
22	initi	47	inst8	72	dl	97	call
23	error	48	specif	73	refer	98	tefsw
24	inst9	49	tc	74	sensor	99	num
25	paramet	50	fault	75	type	100	current

required optimal cut-off point that maximizes both sensitivity and specificity. An overall indication of a ROC curve is the area under the curve (AUC). Values of AUC range from 0 to 1 and higher values indicate better prediction results.

c. *Validation method used*: The validation method used in our study is Hold-out validation (70-30 ratio) in which the entire dataset is divided into 70 % training data and remaining 30 % as test data. There are two methods by which we can assign the cases. Either we can randomly assign the cases based on relative number of cases or else we can use a partitioning variable. We have not randomly assigned the cases, but have rather used a partitioning variable to assign the cases. In other words, we have used a variable that splits the given dataset into training and testing samples in 70–30 ratio. This variable can have the value either 1 or 0. All the cases with the value of 1 for the variable are assigned to the training samples and all the other cases are assigned to the testing samples. To get more generalized and accurate results, we have done validation using 10 separate partitioning variables. A single variable is used at a time on the basis of which the given dataset will be divided randomly in the ratio of 70–30. The corresponding training samples will be used by two machine learning methods (MLP and DT) and one statistical method (MMLR) to predict the model and the remaining testing samples will be used to validate the model. The same process is repeated for 10 runs corresponding to each partitioning variable. The procedure of Hold-out validation corresponding to a single partitioning variable has been depicted below in Fig. 2.

4 Result analysis

In this section, we will analyze the results obtained when the three most popular methods viz. MMLR, MLP and DT have been applied to different datasets of PITS namely PITS A, PITS C, PITS D and PITS E. We have divided this section into the following Sects. 4.1, 4.2 and 4.3. Section 4.1 presents the results of MMLR method when applied on each of these PITS dataset corresponding to top-5, top-25, top-50 and top-100 words. On a similar basis, Sects. 4.2 and 4.3 present the results of MLP and DT

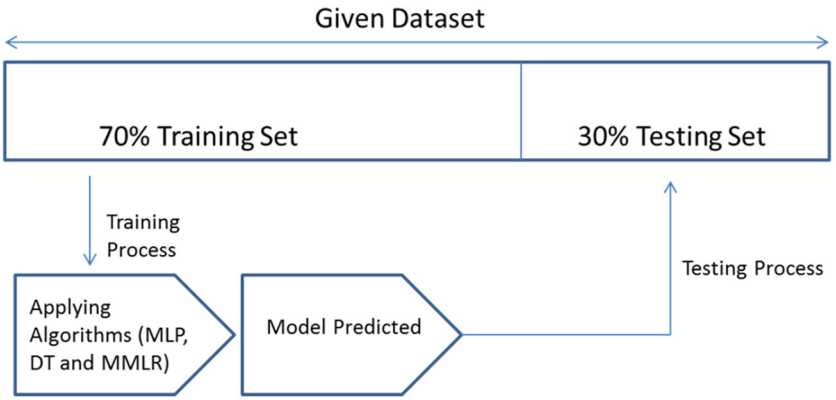

Fig. 2 The procedure of Hold-out validation corresponding to a single partitioning variable

method respectively for each of the PITS dataset. In our study, we have considered Area Under the Curve (AUC), sensitivity and the cut-off point as the performance measures which are used to evaluate the performance of the predicted model. The results shown below depict the average values of Area Under the Curve (AUC), sensitivity and the cut-off point. As already mentioned, we have performed holdout validation for 10 separate partitioning variables in order to get more generalized results. Corresponding to each partitioning variable, we have been noting down the value of AUC, sensitivity and cut-off point. As a result, we will be having 10 different values of these performance measures corresponding to each of the 10 consecutive runs. Then, the average of these 10 values was computed and reported as the final result. These results are being depicted below in the following subsections.

4.1 Multi-nominal multivariate logistic regression

This section presents the results obtained when MMLR method is being applied to different PITS datasets with respect to varying severity levels viz. high, medium, low and very low. Tables 7 and 8 depict the MMLR results with respect to PITS A dataset, Tables 9 and 10 depicts the MMLR results with respect to PITS C and PITS D dataset respectively and finally Tables 11 and 12 depict the MMLR results with respect to PITS E dataset. We will be explaining each of these tables below.

As can be seen from Tables 7 and 8 that the model has performed very well in predicting the defects at high severity levels irrespective of the number of words considered for classification. This observation is indicative from the value of AUC. As can be seen, the value of AUC for top-5 words is 0.84 corresponding to high severity defects. Similarly, the value of AUC is as high as 0.85, 0.80 and 0.71 for top-25, 50 and 100 words respectively. Also, AUC value lies in the range of 0.62–0.76 for medium severity defects for each set of words. This means that the performance of the model is good even for medium severity defects. In contrast to this, model has not performed nicely in predicting the defects at low and very low severity levels in the case of top-5 and top-100 words. This is so because the value of AUC is 0.63 with the sensitivity value as 58.4 % for low severity defects and is 0.68 with the sensitivity value being 68.4 % for very low severity defects when top-5 words were considered for classification. Also, much similar result can be observed while considering top-100 words. The value of AUC is as low as 0.53 with 54.1 % sensitivity value corresponding to defects at low severity levels and is 0.55 with 57.1 % sensitivity value for very low severity defects. On the other hand, performance of the model in predicting low and very low severity defects corresponding to top-25 and 50 words is similar to its performance in predicting medium severity defects. So we can conclude the analysis by saying that the performance of MMLR model for PITS A dataset is very good in predicting high severity defects and good in predicting medium severity defects irrespective of the number of words considered. Also, the performance is good in predicting low and very low severity defects for top-25 and 50 words and is poor for top-5 and 100 words.

Now, we would be elaborating on Table 9 which shows the results corresponding to PITS C dataset corresponding to medium, low and very low severity levels. There are no results corresponding to high severity level. This is so as there are no reports in PITS C dataset which have high severity level. This can be seen from Table 2. Table 9 shows that the performance of MMLR model for PITS C dataset is somewhat consistent in predicting the defects at medium and low severity defects corresponding to each set of words. This is evident from the maximum average value of AUC being 0.96 and 91.9 % sensitivity value for medium severity defects and 0.94 as the maximum average value of AUC and 90.4 % sensitivity value for low severity defects. On the contrast, performance of the model is average in predicting very low severity defects as the values of AUC lies in the range of 0.69 and 0.80. Thus, we can conclude by saying that the model performed very well in terms of medium and low severity defects and showed an average performance for very low severity defects.

Table 7 Results of MMLR for top-5 and top-25 words of PITS A dataset

	Top-5 words				Top-25 words			
	High	Medium	Low	Very low	High	Medium	Low	Very low
AUC	0.84	0.73	0.63	0.68	0.85	0.76	0.72	0.71
Sens	77.3	73.7	58.4	68.4	76.9	71.0	64.0	62.1
Cut-off	0.25	0.41	0.29	0.05	0.27	0.42	0.26	0.02

Table 8 Results of MMLR for top-50 and top-100 words of PITS A dataset

	Top-50 words				Top-100 words			
	High	Medium	Low	Very low	High	Medium	Low	Very low
AUC	0.80	0.76	0.65	0.76	0.71	0.62	0.53	0.55
Sens	74.3	71.8	61.4	73.1	67.0	60.6	54.1	57.1
Cut-off	0.23	0.30	0.16	0.00	0.14	0.10	0.18	0.00

Table 9 Results of MMLR for top-5, top-25, top-50 and top-100 words of PITS C dataset

	Top-5 words			Top-25 words			Top-50 words			Top-100 words		
	Medium	Low	Very low	Medium	Low	Very low	Medium	Low	Very low	Medium	Low	Very low
AUC	0.92	0.87	0.75	0.96	0.94	0.79	0.92	0.92	0.80	0.84	0.84	0.69
Sens	81.2	78.5	83.5	91.9	90.4	72.3	87.3	88.6	77.8	80.9	82.3	72.4
Cut-off	0.41	0.63	0.05	0.50	0.41	0.10	0.20	0.39	0.10	0.31	0.20	0.02

Table 10 Results of MMLR for top-5, top-25, top-50 and top-100 words of PITS D dataset

	Top-5 words		Top-25 words		Top-50 words		Top-100 words	
	Medium	Low	Medium	Low	Medium	Low	Medium	Low
AUC	0.90	0.90	0.80	0.70	0.80	0.80	0.70	0.80
Sens	79.9	100.0	75.5	72.7	74.8	86.0	68.5	80.7
Cut-off	0.78	0.13	0.43	0.36	0.51	0.28	0.49	0.18

Table 11 Results of MMLR for top-5 and top-25 words of PITS E dataset

	Top-5 words				Top-25 words			
	High	Medium	Low	Very low	High	Medium	Low	Very low
AUC	0.57	0.61	0.57	0.62	0.68	0.66	0.67	0.59
Sens	57.1	53.0	55.5	56.4	62.4	59.4	61.3	56.3
Cut-off	0.03	0.64	0.28	0.04	0.01	0.65	0.29	0.03

Table 12 Results of MMLR for top-50 and top-100 words of PITS E dataset

	Top-50 words				Top-100 words			
	High	Medium	Low	Very low	High	Medium	Low	Very low
AUC	0.63	0.66	0.69	0.63	0.61	0.67	0.59	0.60
Sens	61.8	61.3	64.8	56.4	62.0	63.6	57.0	55.2
Cut-off	0.00	0.63	0.29	0.01	0.11	0.53	0.14	0.01

Table 10 depicts the result for PITS D dataset. This table also shows the result corresponding to only medium and low severity defects. There are no results corresponding to high and very low severity levels. This is so as for each high and very low severity level, there is only 1 report present in PITS D dataset as can be seen from

Table 2. So, after hold out validation is being conducted, there are no reports left corresponding to these severity levels for which results could be computed. Now, it is very clear from Table 10 that the model has performed consistently well in predicting the defects for both the medium and low severity levels. The average value of AUC for medium severity level lie in the range of 0.70 and 0.90 and lie in the range of 0.70 and 0.90 for low severity level. Thus, consistent behavior of the model is indicated.

Tables 11 and 12 depict the results for PITS E dataset. Performance of the model is poor for PITS E dataset irrespective of the number of words considered for classification and irrespective of the degree of the severity level. This consistent behavior of the model can be seen with the values of AUC lying in the range of 0.57 and 0.67 when taken into consideration all the severity levels for all sets of words.

4.2 Multi-layer perceptron

This section presents the results obtained when MLP method is being applied to different PITS datasets with respect to varying severity levels viz. high, medium, low and very low. Tables 13 and 14 depict the MLP results with respect to PITS A dataset, Tables 15 and 16 depicts the MLP results with respect to PITS C and PITS D dataset respectively and finally Tables 17 and 18 depict the MLP results with respect to PITS E dataset. We will be explaining each of these tables below.

Tables 13 and 14 show that the performance of MLP model is exceptionally well in predicting the defects for PITS A dataset at high severity level for each set of words. This is evident from the values of AUC being 0.86, 0.88, 0.91 and 0.91 for top-5, top-25, top-50 and top-100 words respectively. Also, model has performed quite well for the remaining severity levels corresponding to top-50 words as the value of AUC is 0.82, 0.80 and 0.81 for medium, low and very low severity defects. On the other hand, it can be seen that the model has shown a consistent performance in predicting the defects corresponding to top-5, top-25 and top-100 words at medium, low and very low severity levels. So, it can be concluded that the high severity defects of PITS A dataset should be predicted using MLP model.

Table 15 shows that the model has performed very well in predicting the defects of PITS C dataset at medium and low severity levels as the AUC value is lying in the range of 0.93 and 0.99 with the corresponding sensitivity value as 86.6 and 94.9 %. Performance of the model is also good for very low severity defects and thus we can say that the MLP model has shown an overall very good performance in predicting the defects of PITS C dataset.

Similar results can also be seen from Table 16 wherein the model has performed very well in predicting the defects of PITS D dataset at both medium and low severity levels. This is so because the average value of AUC is as high as 0.91 with 85.5 % sensitivity value for top-100 words corresponding to medium severity defects. Not only this, average value of AUC is even much higher being 0.99 for low severity defects with 100 % sensitivity value. So, from this analysis, it can be concluded that MLP model should be widely used in predicting the type of defects present in PITS D dataset.

As is very evident from Tables 17 and 18, performance of the MLP model is poor for the defects of PITS E dataset. The average value of AUC is lying in the range of 0.57 and 0.77 for high severity defects when taken into consideration all the set of words. Similarly, the maximum value of AUC is 0.73, 0.74 and 0.71 for medium, low and very low severity defects for all the set of words. These values are very less as compared to the values of AUC being obtained for the defect prediction corresponding to the other datasets of PITS. So, the conclusion which can be drawn is that the MLP model is not suitable to predict the type of defects present in PITS E dataset.

4.3 Decision tree

This section presents the results obtained when DT method is being applied to different PITS datasets with respect to varying severity levels viz. high, medium, low and very low. Tables 19 and 20 depict the DT results with respect to PITS A dataset, Tables 21 and 22 depicts the DT results with respect to PITS C and PITS D dataset respectively and finally Tables 23 and 24 depict the DT results with respect to PITS E dataset. We will be explaining each of these tables below.

It is quite evident from Tables 19 and 20 that the performance of DT model is very good in predicting the defects of high severity level for each of the top-5, top-25, top-50 and top-100 words as their respective average

Table 13 Results of MLP for top-5 and top-25 words of PITS A dataset

	Top-5 words				Top-25 words			
	High	Medium	Low	Very low	High	Medium	Low	Very low
AUC	0.86	0.78	0.72	0.74	0.88	0.80	0.77	0.76
Sens	70.8	60.0	66.2	71.4	80.2	72.3	67.9	69.0
Cut-off	0.17	0.44	0.32	0.05	0.25	0.44	0.26	0.03

Table 14 Results of MLP for top-50 and top-100 words of PITS A dataset

	Top-50 words				Top-100 words			
	High	Medium	Low	Very low	High	Medium	Low	Very low
AUC	0.91	0.82	0.80	0.81	0.91	0.80	0.80	0.75
Sens	83.8	73.7	70.9	70.7	83.0	71.9	72.2	64.9
Cut-off	0.25	0.44	0.23	0.03	0.26	0.42	0.27	0.03

Table 15 Results of MLP for top-5, top-25, top-50 and top-100 words of PITS C dataset

	Top-5 words			Top-25 words			Top-50 words			Top-100 words		
	Medium	Low	Very low	Medium	Low	Very low	Medium	Low	Very low	Medium	Low	Very low
AUC	0.93	0.89	0.74	0.99	0.98	0.86	0.98	0.98	0.86	0.96	0.93	0.68
Sens	86.6	81.7	86.9	94.9	93.7	80.4	94.2	93.5	81.0	92.7	89.8	73.1
Cut-off	0.37	0.65	0.04	0.56	0.59	0.04	0.54	0.57	0.02	0.54	0.67	0.02

Table 16 Results of MLP for top-5, top-25, top-50 and top-100 words of PITS D dataset

	Top-5 words		Top-25 words		Top-50 words		Top-100 words	
	Medium	Low	Medium	Low	Medium	Low	Medium	Low
AUC	0.89	0.88	0.90	0.99	0.85	0.99	0.91	0.98
Sens	77.9	100.0	83.4	100.0	91.1	100.0	85.5	100.0
Cut-off	0.86	0.13	0.57	0.46	0.64	0.56	0.74	0.45

Table 17 Results of MLP for top-5 and top-25 words of PITS E dataset

	Top-5 words				Top-25 words			
	High	Medium	Low	Very low	High	Medium	Low	Very low
AUC	0.57	0.63	0.60	0.65	0.77	0.70	0.70	0.67
Sens	55.7	58.7	58.9	59.9	69.5	62.3	63.5	60.9
Cut-off	0.03	0.64	0.29	0.04	0.04	0.64	0.29	0.05

Table 18 Results of MLP for top-50 and top-100 words of PITS E dataset

	Top-50 words				Top-100 words			
	High	Medium	Low	Very low	High	Medium	Low	Very low
AUC	0.64	0.72	0.74	0.64	0.66	0.73	0.71	0.71
Sens	63.6	66.4	66.1	58.5	65.6	66.8	65.4	66.3
Cut-off	0.03	0.64	0.30	0.04	0.29	0.65	0.29	0.09

values of AUC are 0.85, 0.83, 0.84 and 0.84. On the other hand, DT model shows a consistent behavior for the defects having medium, low and very low severity levels. This is evident from the range of AUC values for these severity levels. AUC values lies in the range of 0.68–0.78, 0.69–0.72, 0.69–0.71 for medium, low and very low severity level defects respectively. Thus, we can conclude that the performance of DT model for PITS A dataset is very good for the defects of high severity level and is average for the defects of remaining severity levels i.e. medium, low and very low severity levels.

Table 21 depicts the results of DT for PITS C dataset. The performance of the model is exceptionally good in predicting the defects at medium and low severity levels as can be seen from the values of AUC and sensitivity at both these severity levels. The AUC values lie in the range of 0.92–0.97 for the defects at medium severity level with 90.8–96.0 % range of sensitivity values. Similarly, the range of AUC values for low severity defects is 0.87–0.96 with 78.5–89.6 % sensitivity range. In contrast to this, the model has shown an average performance in predicting the defects at very low severity levels as the range of AUC

Table 19 Results of DT for top-5 and top-25 words of PITS A dataset

	Top-5 words				Top-25 words			
	High	Medium	Low	Very low	High	Medium	Low	Very low
AUC	0.85	0.78	0.72	0.71	0.83	0.74	0.71	0.69
Sens	73.6	93.2	72.0	82.5	70.2	89.6	79.1	83.8
Cut-off	0.42	0.33	0.30	0.04	0.23	0.42	0.30	0.02

Table 20 Results of DT for top-50 and top-100 words of PITS A dataset

	Top-50 words				Top-100 words			
	High	Medium	Low	Very low	High	Medium	Low	Very low
AUC	0.84	0.72	0.69	0.70	0.84	0.68	0.69	0.68
Sens	71.7	91.4	86.8	93.0	70.8	87.8	97.9	94.3
Cut-off	0.21	0.44	0.27	0.02	0.33	0.45	0.19	0.02

Table 21 Results of DT for top-5, top-25, top-50 and top-100 words of PITS C dataset

	Top-5 words			Top-25 words			Top-50 words			Top-100 words		
	Medium	Low	Very low	Medium	Low	Very low	Medium	Low	Very low	Medium	Low	Very low
AUC	0.92	0.87	0.70	0.97	0.96	0.76	0.97	0.96	0.76	0.97	0.96	0.76
Sens	90.8	78.5	86.9	96.0	89.6	65.2	96.0	89.6	65.2	96.0	89.6	65.2
Cut-off	0.32	0.66	0.03	0.53	0.39	0.02	0.53	0.39	0.02	0.53	0.39	0.02

Table 22 Results of DT for top-5, top-25, top-50 and top-100 words of PITS D dataset

	Top-5 words		Top-25 words		Top-50 words		Top-100 words	
	Medium	Low	Medium	Low	Medium	Low	Medium	Low
AUC	0.66	0.65	0.85	0.93	0.83	0.86	0.88	0.95
Sens	31.0	50.0	92.5	98.0	90.9	90.0	96.9	90.0
Cut-off	0.93	0.57	0.83	0.20	0.74	0.27	0.60	0.54

Table 23 Results of DT for top-5 and top-25 words of PITS E dataset

	Top-5 words				Top-25 words			
	High	Medium	Low	Very low	High	Medium	Low	Very low
AUC	0.57	0.60	0.57	0.59	0.56	0.66	0.65	0.60
Sens	55.0	83.2	65.2	62.5	74.5	86.9	85.2	59.0
Cut-off	0.03	0.59	0.27	0.03	0.02	0.62	0.25	0.05

Table 24 Results of DT for top-50 and top-100 words of PITS E dataset

	Top-50 words				Top-100 words			
	High	Medium	Low	Very low	High	Medium	Low	Very low
AUC	0.56	0.62	0.61	0.60	0.60	0.62	0.60	0.59
Sens	84.0	87.1	88.7	63.1	96.7	90.4	92.9	57.9
Cut-off	0.01	0.62	0.23	0.04	0.01	0.62	0.23	0.04

values is 0.70–0.76 with 65.2–86.9 % range of sensitivity values.

Table 22 shows that the performance of DT model for PITS D dataset is not good when top-5 words are being considered for classification. This is clear from the average

values of AUC being 0.66 for medium severity defects and 0.65 for low severity defects. In contrast to this, the performance of the model is very good for top-25, top-50 and top-100 words for both the levels of severity i.e. medium and low. This is evident from the values of AUC and the values of sensitivity. Thus, we can conclude that DT model with respect to PITS D dataset is not preferable for top-5 words when considered for classification and is suitable when top-25, top-50 and top-100 words are being considered for classification.

Tables 23 and 24 depict the results of DT model on PITS E dataset. As is very clear from the results, the performance of the model is very poor. In general, it has been observed from the previous analysis also that none of the models are giving good results with respect to the defects of PITS E dataset. But, in contrast to the previous results, the most noticeable values which can be seen from Tables 23 and 24 are the values of sensitivity. The values of sensitivity are as high as 96.7, 90.4, 92.9 % for high, medium and low severity defects when taken into consideration each set of words despite the low values of AUC corresponding to these sensitivity values. But, for the defects of very low severity levels, the values of both AUC and sensitivity are quite low. The results for the defects of very low severity levels are very much similar to the results obtained for the previous models on PITS E defects. Thus, we can conclude that the performance of DT model is good for the prediction of the PITS E defects having high, medium and low severity levels for each set of words. But, the model is not well suited for the defects having very low severity levels.

5 Discussion

The defect prediction results obtained when applying different machine learning and statistical methods have been discussed in this section for further analysis of the data. The data which is being used in the work refers to the defect data contained in different PITS datasets being used by the NASA's engineers. These PITS datasets are named as PITS A, PITS C, PITS D and PITS E. Hold-out validation has been done for these datasets and then ROC analysis is carried out to evaluate the performance of the models. Here, we would be discussing the performance of three models viz. MMLR model, MLP model and DT model on each of these PITS datasets so as to gain insight regarding which model has performed better on which dataset.

The results obtained for the different models indicate that all the three models viz. MMLR model, MLP model and DT model have performed very well in predicting the defects present in PITS A dataset corresponding to high severity level for each set of words. This is indicative from the values of AUC which are 0.84, 0.85, 0.80 and 0.71 for the MMLR model corresponding to top-5, top-25, top-50 and top-100 words respectively. Very much similar to this, the values of AUC for MLP model are 0.86, 0.88, 0.91 and 0.91 for each set of these words at high severity level. Last but not the least, the AUC values for DT model corresponding to each set of words at high severity level are 0.85, 0.83, 0.84 and 0.84. In contrast to this, it has been observed that the two models viz. MMLR model and MLP model have shown an average performance for the remaining severity levels viz. medium, low and very low severity levels corresponding to the type of defects present in PITS A dataset. The range of AUC values for the defects with medium severity level lies between 0.62 and 0.82 with sensitivity values lying between 60.0 and 73.7 % when considering both MMLR and MLP prediction models together. Likewise, low severity defects have their AUC values lying between 0.53 and 0.80 along with the maximum value of sensitivity as 72.7 % for these two models. Similarly, defects with very low severity levels also have an AUC range between 0.55 and 0.81 along with sensitivity range between 57.1 and 71.4 %. But the values of sensitivity for DT model are very good lying in range of 87.8–93.2 % for medium severity defects, in the range of 72.0–97.9 % for low severity defects and in the range of 82.4–94.3 % for very low severity defects despite the low values of AUC falling in the range of 0.68–0.78 corresponding to all these three severity levels. So, from the above analysis, it can be concluded that any of the three models can be used to predict high severity defects present in PITS A dataset. But, in order to predict the defects at medium, low and very low severity levels, only DT model is suitable.

Now, if we consider PITS C dataset, we can see that the performance of all the three models has been reported with respect to only three severity levels viz. medium, low and very low. This is so because, there are no defects present in PITS C dataset which have high severity level. Now, the performance of all the three models (MMLR, MLP and DT model) is exceptionally good for the defects having medium and low severity levels when taken into consideration each set of words. This is clear from the values of AUC which are lying in the range of 0.84–0.99 corresponding to the medium severity defects corresponding to all the three models. Similarly, AUC values are falling in the range of 0.84 to 0.98 for the defects having low severity level. Also, the models have performed quite well in predicting the defects at very low severity level as the AUC values fall in the range of 0.68–0.86. Thus, we can summarize by saying that prediction capability of all the three models have been very good in predicting the type of defects present in PITS C dataset.

Similar to PITS C dataset, the analysis of PITS D dataset have also been considered with respect to the defects having medium and low severity levels. This is so as for each high and very low severity level, there is only one report present in PITS D dataset. Therefore, once hold-out validation is conducted, there are no reports left corresponding to these severity levels. Now, as far as performance of the models is concerned, it has been observed that MMLR and MLP models have shown a consistent performance in predicting the defects at both the severity levels viz. medium and low severity levels. With respect to the medium severity level, the values of AUC are lying between 0.70 and 0.90 corresponding to the MMLR and MLP models and for low severity level the AUC values are falling in the range of 0.70–0.99 corresponding to these two models. With respect to DT model, it was observed that the performance of the model got better as the number of words considered for classification increased. In other words, DT model performed very well corresponding to top-25, top-50 and top-100 words as is evident from the values of AUC. The AUC values lie between 0.83 and 0.88 for medium severity defects and lie between 0.86 and 0.95 for low severity defects. Thus, we can conclude that MMLR and MLP models performed very well with respect to PITS D dataset irrespective of the number of words considered for classification, but DT model is preferable only when more number of words are considered for classification.

In contrast to the above analysis, the performance of all the three models for PITS E dataset is very poor. In general, it has been observed from the results that none of the models are giving good results with respect to the defects of PITS E dataset. This is so as the values of AUC for the MMLR model are lying in the range of 0.57–0.69 when taken into consideration each set of words and all the severity levels. Also, AUC values lie between 0.57 and 0.77 for the MLP model and lie between 0.57 and 0.66 corresponding to the DT model for each set of words. Despite the low values of AUC for all the three models, the values of sensitivity for the DT model corresponding to the defects having high, medium and low severity defects are 96.7, 90.4 and 92.9 %. Thus, we can say that DT model can be used for the prediction of the type of defects which are present in PITS E dataset.

Thus, we can conclude the discussion by saying that the machine learning methods as well as the statistical method employed in our study for development of the defect prediction models have shown an overall very good performance with respect to PITS C and PITS D datasets. In other words, the three models viz. MMLR model, MLP model and DT model can predict all the type of defects present in PITS C and PITS D datasets corresponding to each severity level. But regarding PITS A dataset, these models are only suitable to predict the defects having high severity level. For the prediction of defects having medium, low and very low severity levels, only DT model is preferable. Not only this, DT model has even shown a very good performance in predicting the defects present in PITS E dataset. In other words, MMLR model and MLP model have shown a poor performance for the prediction of the PITS E defects. Another very important point to note here is that the results being reported in our study are the average value of the AUC, sensitivity and cut-off point. As already mentioned, we have performed holdout validation for 10 separate partitioning variables in order to get more generalized results. Corresponding to each partitioning variable, we have been noting down the value of AUC, sensitivity and cut-off point. So, we will be having 10 different values of these performance measures corresponding to each of the 10 consecutive runs. Finally, the average of these 10 values was computed and reported as the final result.

In near future, we intend to explore more datasets available in open source software repository and then would be replicating this study across different datasets by applying more set of machine learning and statistical methods. Through this, we will be able to get more generalized and accurate results, which will in turn help us to draw more meaningful observations and thus, ending up with better conclusion.

6 Conclusion

In real-time applications, the defects are being introduced in the software at an exponential rate. This defect rate has been increasing tremendously over the past few years which is hampering the overall progress of an organization development in terms of its infrastructure, man-power and money. Now, the time has come when there is an urgent need to look into this situation and come out with an appropriate solution. One of the most viable solutions is to have a defect tracking system in various software and IT organizations which will be used to store the defects. But, the defects introduced in the software come with varying severity levels which is the most emerging issue in present days which cannot be tracked by these defect tracking systems. As a result, in this paper we have developed defect prediction models which will be used to predict the severity of the defects which are present in the software. We have employed two machine learning methods viz. Multi-Layer Perceptron (MLP) and Decision Tree (DT) and one statistical method i.e. Multi-nominal Multivariate Logistic Regression (MMLR) to predict the models. These methods are used over the defect data which we have taken from PITS datasets. PITS datasets are being most popularly used by NASA's engineers and are named as PITS A, PITS

C, PITS D and PITS E. The defect data present in these datasets is of raw form which cannot be applied to the machine learning and statistical methods. Therefore, we have used a series of text mining techniques in order to extract the relevant information from these datasets. Different measures have been used in order to evaluate the performance of these models such as Area Under the ROC curve, sensitivity value and cut-off point.

From the observed results, it was seen that all the three models viz. MMLR model, MLP model and DT model have shown a fairly good performance in predicting the defects at varying severity levels (high, medium, low and very low). With respect to PITS A dataset, it was observed that all these three models have shown a very good performance in predicting the defects having high severity level. But, in order to predict the defects having medium, low and very low severity levels, only DT model is suitable. Similarly, with respect to PITS C and PITS D dataset, it can be summarized that prediction capability of all the three models have been very good irrespective of the number of words considered for classification. But, the performance of the DT model was much better in predicting the defects when more number of words were considered for classification. Also, it was only DT model which was suitable to predict the defects present in PITS E dataset. So, it can be concluded that DT model has shown an overall best performance for all the PITS datasets followed by the performance of remaining two models.

References

Aggarwal KK, Singh Y, Kaur A, Malhotra R (2009) Empirical analysis for investigating the effect of object-oriented metrics on fault proneness: a replicated case study. Softw Process Improve Practice 16(1):39–62

Canfora G, Cerulo L (2005) How software repositories can help in resolving a new change request. In: Workshop on empirical studies in reverse engineering

Catal C, Diri B (2009) A systematic review of software fault prediction studies. Expert Syst Appl 36:7346–7354

Cubranic D, Murphy GC (2004) Automatic bug triage using text categorization. In: Proceedings of the sixteenth international conference on software engineering and knowledge engineering

Emam KE, Melo W (1999) The prediction of faulty classes using object-oriented design metrics. Technical Report NRC 43609

Emam KE, Benlarbi S, Goel N, Rai S (1999b) A validation of object-oriented metrics. NRC Technical report ERB-1063

Gondra I (2008) Applying machine learning to software fault-proneness prediction. J Syst Softw 81:186–195

Gyimothy T, Ferenc R, Siket I (2005) Empirical validation of object-oriented metrics on open source software for fault prediction. IEEE Trans Softw Eng 31(10):897–910

Hosmer D, Lemeshow S (1989) Applied logistic regression. Wiley, New York

Ikonomakis M, Kotsiantis S, Tampakas V (2005) Text classification using machine learning techniques. WSEAS Trans Comput 4(8):966–974

Jiang Y, Cukic B, Ma Y (2008) Techniques for evaluating fault prediction models. Empir Softw Eng 13(15):561–595

Lamkanfi A, Serge D, Giger E, Goethals B (2010) Predicting the severity of a reported bug. In: 7th IEEE working conference on mining software repositories (MSR), pp 1–10

Malhotra R, Jain A (2012) Fault prediction using statistical and machine learning methods for improving software quality. J Inf Process Syst 8(2):241–262

Malhotra R, Singh Y (2011) On the applicability of machine learning techniques for object-oriented software fault prediction. Softw Eng Int J 1(1):24–37

Menzies T, Marcus A (2008) Automated severity assessment of software defect reports. In: IEEE international conference on software maintenance (ICSM)

Myers G, Badgett T, Thomas T, Sandler C (2004) The art of software testing, 2nd edn. John Wiley & Sons Inc, Hoboken

Ohlsson N, Zhao M, Helander M (1998) Application of multivariate analysis for software fault prediction. Softw Qual J 7:51–66

Olague H, Etzkorn L, Gholston S, Quattlebaum S (2007) Empirical validation of three software metrics suites to predict fault-proneness of object-oriented classes developed using highly iterative or agile software development processes. IEEE Trans Softw Eng 33(8):402–419

Pai G (2007) Empirical analysis of software fault content and fault proneness using Bayesian methods. IEEE Trans Softw Eng 33(10):675–686. doi:10.1109/TSE.2007.70722

Porter A, Selby R (1990) Empirically guided software development using metric-based classification trees. IEEE Softw 7(2):46–54. doi:10.1109/52.50773

Runeson P, Alexandersson M, Nyholm O (2007) Detection of duplicate defect reports using natural language processing of 29th IEEE international conference on software engineering (ICSE), pp 499–508

Sari GIP, Siahaan DO (2011) An attribute selection for severity level determination according to the support vector machine classification result. In: Proceedings of the 1st international conference on information systems for business competitiveness (ICISBC)

Shatnawi R, Li W (2008) The effectiveness of software metrics in identifying error-prone classes in post-release software evolution process. J Syst Softw 81:1868–1882

Singh Y, Kaur A, Malhotra R (2010) Empirical validation of object-oriented metrics for predicting fault proneness models. Softw Qual J 18:3–35

Wang X, Zhang L, Xie T, Anvik J, Sun J (2008) An approach to detecting duplicate bug reports using natural language and execution information. Association for Computing Machinery

Widro B, Lehr MA (1990) 30 years of adaptive neural networks: perceptron, madaline, and backpropagation. Proc IEEE 78(9):1415–1442

Yu P, Systa T, Muller H (2002) Predicting fault-proneness using OO metrics: an industrial case study. In: Proceedings of sixth European conference on software maintenance and reengineering, Budapest, pp 99–107

Zhou Y, Leung H (2006) Empirical analysis of object-oriented design metrics for predicting high and low severity faults. IEEE Trans Softwre Eng 32(10):771–789

Zhou Y, Xu B, Leung H (2010) On the ability of complexity metrics to predict fault-prone classes in object-oriented systems. J Syst Softw 83:660–674

Rajni Jindal

Ruchika Malhotra

Abha Jain