Bug Triage Using Data Reduction with Priority and Security

Vijay N. Kukre[1], Shyam Gupta[2]

[1, 2]Department of Computer Engineering, Siddhant College of Engineering, Sudumbare Pune - 412109, India

Abstract: ***Many Open Source Software Development organizations pay quite 45 % of expense in solving bugs. Bug triaging is significant phase in procedure of bug solving. The purpose of bug triaging is to allocate coming bugs to appropriate developer. The current bug triaging methodologies are initiated on algorithms, which form classifiers from the training data sets of bug report into training, these methods are suffering from huge scale and low quality training set. Here, the training sets reduction with feature selection method Chi Square Statistic (CH) and instance selections method*** *Iterative Case Filter (ICF)* ***for bug triaging are suggested. Feature selection and instance selections methods are used to get better the correctness of CHI , instance selections algorithm Iterative Case Filters (ICF) are premeditated. The training sets reduction by the bug records is calculated. For training sets, 70% words and 50% bugs report are separated after training sets lessening. The outcome illustrates that novel and minor training data sets deliver improved correctness than unique one. The next advantage is, it provides priority according to severity of bug so that bug can be solved on the priority basis and security using AES algorithm so that no another developer can access it.***

Keywords: Bug data reduction, feature selection, instance selection, bug triage

1. Introduction

Most of the Open Source Software Development Companies pays lot of the money for fixing the bugs. They have bug repository that collects all the knowledge associated with bugs. In bug depository, every software package bug incorporates a bug report. The bug report includes matter info concerning the bug and updates related with standing of bug fixing. Once a bug report is made, a person who is called as bug triager assigns this bug to a developer and developer fix this bug. The assigned bug may have allotted to a different developer if the current assigned developer cannot fix this bug. The method of assigning an accurate developer for fixing the bug is termed as a bug triage. Bug sorting is one among the foremost time consuming step in handling of bugs in software package comes. Manual bug sorting by a person called as triager is time consuming process as the quantity of daily bugs is massive and lack of information in developers regarding all bugs. That is why, bug fixing becomes up in costly time loss, high price and low accuracy. The information keep in bug reports has 2 main challenges. First, the massive scale information and second low quality of knowledge because of sizable amount of daily reported bugs, the number of bug reports is scaling up within the repository. Noisy and redundant bug’s is degrading the standard of bug reports. In this paper an efficient automated bug fixing system is projected which will save lots of the labor cost of developers and time by creating a top quality set of bug data by removing the redundant and non-informative bug reports. Beside this, it provides priority according to severity of bug so that bug can be solved on the priority basis and security using AES algorithm so that no another developer can access it.

2. Related Work

A lengthy step of handling software bugs is called as bug triage, which is used to allocate a accurate developer to repair a new bug. In usual software development, new bugs are manually triaged by an specialist developer, i.e., a human being triager. Due to the huge number of every day bugs and the not have of expertise of all the bugs, manual bug triage is costly by time and low down the accuracy. In manual bug triage in Eclipse, 44% of bugs are allocated by mistake whereas the time cost involving opening one bug and its first triaging is 19.3 days on standard. To avoid the high-priced cost of manual bug triage, existing work [1] has projected an automatic bug triage technique, which uses text classification techniques to expect developers for bug reports. In this technique, a bug report is plotted to a document and a associated developer is plotted to the label of the document. Then, bug triage is transformed into a problem of text classification and is automatically solved with mature text classification techniques, e.g., Naive Bayes. Depends on the results of text classification, a human triager allocates a new bugs to expertise. But, large-scale and low-quality bug data in bug repositories obstruct the techniques of automatic bug triage. So,, data reduction for bug triage is introduced, i.e., how to decrease the bug data to save the labour cost of developers and get better the quality to make easy the method of bug triage. Data reduction for bug triage plans to construct a small-scale and high-quality set of bug data by eliminating bug reports and words, which are redundant or non-informative.

Bug triaging is imperfect, tiresome and time consuming task so going with Revisiting Bug fixing and determination Practices [2]. Bug triaging and fixing exercise as well as bug reassignments and re-openings are considered, within the perspective of the Mozilla and Firefox, which is measured to be representative example of a large-scale open source software development. As well they require consider to perform qualitative and qualitative testing of the bug assignment exercises. They have an interest in providing insights into a number of areas: categorization exercises, reconsider and endorsement processes; source reason investigation of bug reassignments and reopens in open source software code projects; and suggestion for enhancement/reform of bug tracking systems.

Markov chains which supports a graph model detain bug moving record [3] is introduced and has many interesting merits. First, it exposes developer network and to investigate appropriate specialist for a substitute assignment. Secondly, it assists to higher allocate developers to bug reports. A experiment with 445,000 bug reports, this method compacts bug moving records up to seventy two additionally; the model amplifies the prediction accurateness up to twenty three percentages as compared to earliest bug triaging approaches.

Latest study shows that optimizing proposal accuracy negative aspect and proposes a solution of an instance of content-based recommendation (CBR) [4]. However, CBR is popular to source overspecialization, recommending exclusively the group of bugs that every developer have resolved earlier. This problem is ruthless in apply, as some experienced developers may be over loaded which will slow down the bug fixing speed. The two ways are specified to finger out this problem, first authors have a propensity to develop the substance as an optimization problem of every accuracy and outlay. Secondly, they accept a content-boosted cooperative filtering (CBCF), merges an existing CBR with a cooperative filtering (CF), which improves the advice quality of either approach alone.

Techniques given in [5] uses either data repossession and machine learning to look for the principal same bugs previously predetermined and propose knowledgeable developers, or examine the change in data stemming from ASCII text file to suggest professional bug solvers. This scheme does not merge matter likeness with modification set analysis and not make a use of the probable of the difficult between bug reports and alter words in bug data within the designed system, the combination of instance choice and have selection is employed. The designed systems are imposed in java language thus, it will be platform freelance. As there is no constraint on the scope of bug’s info, a tester will insert large number of bugs within the system which will be the biggest benefits of the planned system. While the entire bugs info is accessible all the developers, which will, takes a smaller amount time for the developer to need the choice. Developer will rapidly choose. For the bug to repair. As bug categorization mean to expect the developers who will fix the bugs, hence there is an affinity to track the nearby work to get clear knowledge of unsettled bug reports, e.g., the new bug reports or not fixed bug reports. Furthermore, in bug repositories, various developers have exclusively settled only a few bugs. Such dormant developers might not offer enough info for predicting accurate developers. Hence here have a tendency to take away the developers, who have mounted but ten bugs.

A semi-supervised text classification method used for bug triaging [6] is recommended to avoid the absence of labeled bug reports in previous text supervised methods which merges Naive Bayes classifier and probability maximization to obtain benefit of labeled and unlabeled bug reports. Such approach prepares a classifier with a component of labeled bug reports and then iteratively labels a number of unlabeled bug reports and set up a new classifier with labels of all the bug reports. A subjective opinion record is also used to increase the performance by daunting the burden of several developers in training the classifier. Investigational outcome on bug reports of Eclipse demonstrate that new method is superior to the existing supervised methods compare to classification accuracy of bug triage increases up to 6% but does not offer fully automatic bug triage with a bug repository.

The five term selection methods are used for the correctness of bug task to decrease time and expenditure of bug triaging and also rebalance the workload between developers based on their knowledge [7] so, they carry out testing on four factual datasets. The first term selection method, Log Odds Ratio (LOR) counts the odds of the word available in the positive set stabilized by the negative set. The second term selection method, Chi-Square (X2) test is employ to monitor dependence of two measures. The third term selection method, Term Frequency Relevance Frequency (TFRF) is used to choose further tall frequency for instances in the positive group than in the negative group. The fourth term selection method, Mutual Information (MI) is used to counts the common reliance of two random variables. The fifth term selection method, Distinguishing Feature Selector (DFS) provides discriminatory control of the features above the complete text set quite than being set particular. The result of investigation shows the F-score can be considerably enhanced by selecting a little number of selective terms.

The both feature and instance selection methods are used in proposed to get better the accurateness of bug triage to estimate the training bug data set reduction on the bug data of Eclipse [8]. As a result, 70% words and 50% bug reports are detached following the training bug data set reduction. The investigational results prove that the novel and little training sets be able to give enhanced correctness than the unique one. The drawbacks of their approach are low precision rate and cannot be directly used in other projects because results are based on the bug data from the Eclipse only.

3. Proposed System Architecture

The main purpose of proposed system is to keep watch the all the bugs in the project and construct the project user friendly to all users and bugs free system. The developed products, Bugs and Bug Triage (bug history) are preserved by the system. It has benefit of maintaining bug history it stores all the particulars from bug source to bug resolution i.e. from origin of bug to solution of bug. Every product may have versions for trouble-free maintenance of the product, for easy difference making and all the client of the product is accumulated in the database. It provides the benefit of maintaining users to the bugs and decree provided by them. Proposed System offers the search based on status (idle or active), priority (by providing numbering) and operating system. It offers user and bug ladder, which is useful to know the relation between bugs and users selected to the bug. It provides a fully secured system with password encryption using AES encryption algorithm with priority given to bug to be fixed and facilitate to store attachment files for a bug Track of the bug in a product can be monitored with lower cost and with very less efforts. Maintaining log reports which are help full to know any errors, bugs or mistreat of the system by another users is an additional advantage of this system.

The new bug reports acts as the input to system for fixing the bug from appropriate developer as shown in Figure 1 of the proposed system architecture. Each bug for fixing sends to developer in authority and details of one that makes who have worked on that separate bug for fixing. Bugs for fixing go to developer in authority is mainly separated in two parts, manager and admin.

Basically actual data normally consists of noise and redundancy where noisy data may provide the incorrect meaning about the data investigation techniques and the redundant data may increase the expenditure of data processing. In bug data repositories, the bug reports are entered by developers in normal languages. The low class bugs reside in bug data repositories with the increase in scale. Such extensive and low-class bug data may decline the usefulness of fixing up bugs.

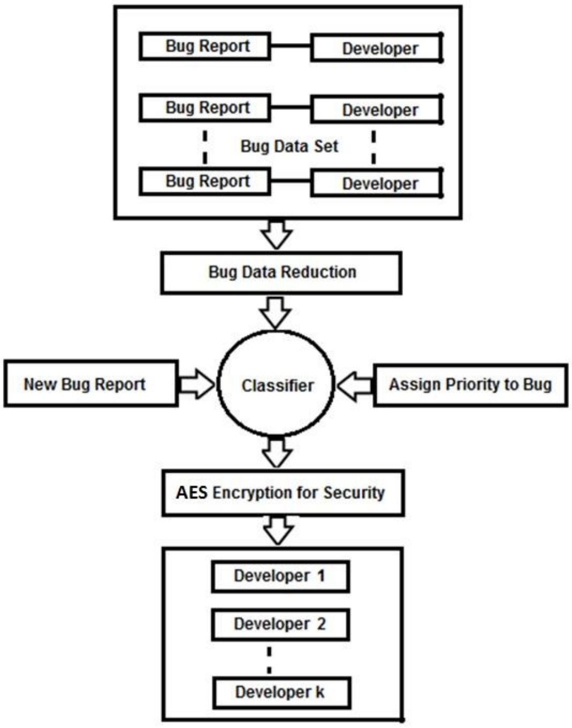

Figure 1: Proposed System Architecture

Figure 2 shows, how the bug data is reduced which is used to prepare training data set for bug triaging hence combination of instance and feature selection is used to eliminate definite bug reports and words. The reduced training set which is used to replace the original training set of bug triage. To differentiate the groupings, we two phases i.e. *FS→IS means* first applying *FS* and then *IS*. Then next, *IS→FS* means first applying *IS* and then *FS*.

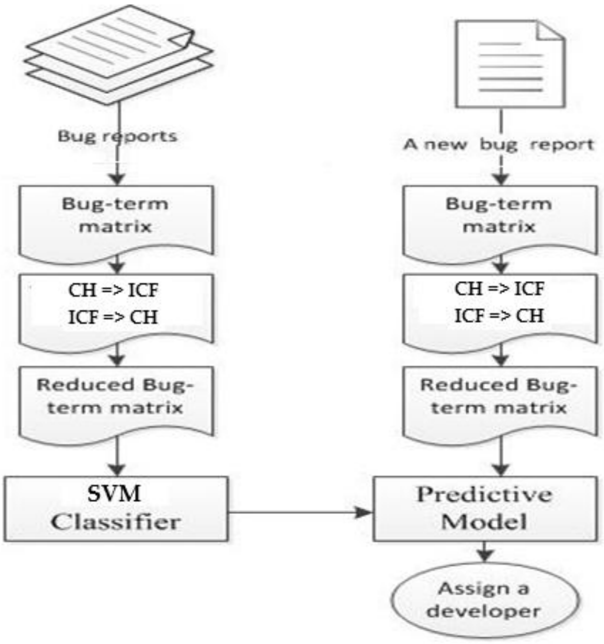

Figure 2: Bug Data Reduction

Proposed system provides predicted results in form of output. Basically, there are two types of users in proposed system. First is developer and second is tester. Developer will get software bugs for fixing which is given to him. Another that can work on only one software bugs for fixing at a time. Tester can link new bugs for fixing to system.

4. Training Set Reduction for bug triage

Now, we will see how to use combination of feature selection algorithm Chi Square Statistic (CH and an instance selection algorithm Iterative Case Filter (ICF) for data reduction for bug triage to minimize noisy or duplicate data in the training data set for bug triage.

4.1 CHI Feature Selection Algorithm (CF)

CHI (χ^2statistic) feature selection algorithm is used to quantity of the dependence between bug report words and software developers. In this paper we are using following mathematical model to find dependence between bug report words and software developers.

$$\mathrm{CHI}(w) = .\max_{dn \in D}(\chi^2(w,d_n)) = \max_{dn \in D}\left(\frac{m\,x\,(A\,xD - C\,x\,B)}{(A+C)\,x(B+D)\,x(A+B)x(C+D)}\right)$$

Where,

$\mathbb{D}$ = Set of Developer
dn = n^{th} Developer in set of Developer
w = Bug word
A = No. of times co-occures of w and d_n
B = No. of times occure of w without d_n
C = No. of times occure of d_n without w
D = No. of times co-occures neither w and d_n

4.2 ICF Instance Selection Algorithm

The ICF (Iterative Case Filter) instance selection algorithm based on the k-Nearest Neighbor algorithm (k-NN) used reduces the number of instances and to improve the quality of training data set in the term of data space required and less response time

4.3 Support Vector Machine (SVM) Classifier

Support Vector Machine (SVM), which is a supervised machine-learning algorithm and mostly used in classification. The advantages of support vector machines are competent in high dimensional spaces, competent where number of dimensions is more than the number of samples, uses a subset of training points in the decision making function called support vectors, so it is also memory competent. Here SVM is used to predict appropriate developer.

4.4 Advanced Encryption Standard (AES) Algorithm for Security

The universally adopted and largely used symmetric encryption algorithm now a days is the Advanced Encryption Standard (AES), which is at least six times quicker than triple DES. AES completes all its calculations on bytes rather than bits. Hence, AES gives the 128 bits of a plaintext block as 16 bytes and arranged in four columns and four rows for processing as a matrix. The number of rounds in AES is variable and determined by on the length of the key, hence AES practices 10 rounds for 128-bit keys.

A set of specially derived keys called round keys is used in the AES encryption process and applied on an array of data that holds exactly one block of data to be encrypted along with other operations, which is called as the state array.

AES steps of encryption for a 128-bit block are given below:
1) Obtain the set of round keys from the cipher key.
2) Initialize the state array with the block data i.e. plaintext.
3) Add the primary round key to the starting state array.
4) Complete nine rounds of state manipulation.
5) Complete the tenth and last round of state manipulation.
6) Copy the last state array out as the encrypted data i.e ciphertext.

5. Algorithms for Bug triage with AES for security

The algorithm as given below is used to shrink the bug data using FS→IS. In this bug data set, the output of bug data reduction is a recently reduced bug data set. To achieve this, FS and IS algorithms are applied sequentially. During feature selection in step 2, some of bug reports may be blank, due to all the words in a bug report are removed and in the feature selection, such blank bug reports are removed. FS→IS and IS→FS are analyzed as two orders of bug data reduction.
Instance selection method is used to reduce the number of instances by deleting noisy and redundant instances. Hence by deleting non-representative instances; it can offer a reduced data set. Feature selection is a pre-processing for selecting a condensed set of features for large-scale data sets. The reduced bug data set is considered as the representative features of the novel feature set. Based on feature selection, words in bug reports are arranged as per said feature values and a given quantity of words with large values are selected as representative features.

```
Algorithm 1   Data Reduction using Feature and Instance
              Selection
Input:        T - Bug Training Set having N words and
              M bug reports,
              FS→IS -Data Reduction flow
              $N_F$ - Total count of words,
              $M_I$ -Total count of bug reports
              $T_{FI}$ - Reduced bug data set for bug triaging
              Dn – Appropriate Developer
Output:
Step 1:       Use FS to N words of Bug Training Set T
              and estimate objective values for all the
              words;
Step 2:       Choose the top $N_F$ words of T and create a
              training set $T_F$ ;
Step 3:       Use IS to $M_I$ bug reports of $T_F$ ;
Step 4:       Stop IS if the number of bug reports is $\leq M_I$
              and create the concluding training set $T_{FI}$
Step 5:       Use SVM to predict appropriate developer
              $D_n$ using training set $T_{FI}$
Step 6:       Assign priority depending on severity of bug
              $M_I$
Step 7:       Apply AES encryption to bug $M_I$
Step 8:       Assign encrypted bug $M_I$ to appropriate
              developer Dn
```

6. Results and Discussion

The outcome of data reduction for bug triage can be calculated in two aspects, i.e. the scales of data sets and the quality of bug triage. Based on Algorithm 1, the scales of data sets (including the number of bug reports and the number of words) are configured as input parameters. The superiority of bug triage calculated with the accurateness, precision and recall of bug triage, which is defined as

$$Accuracy_k = \frac{\text{\# correct relevant developers}}{\text{\# all bug reports in test set}}$$

$$Precision_k = \frac{\text{\# correct relevant developers}}{\text{\# relevant developrers} \times k}$$

$$Recall_k = \frac{\text{\# correct relevant developers}}{\text{\# correct developers}}$$

To examine the accuracy decrease by instance selection,we define the loss from origin to ICF as

$$Lossk = \frac{Accuracykbyorigin - AccuracykbyICF}{Accuracykbyorigin}$$

where the recommendation list size is k. We arrange developers by the number of their fixed bugs in descending order in a data set. That is, we sort classes by the number of instances in classes. Then a new data set with s developers is built by selecting the top-s developers. For one bug data set, we construct new data sets by varying s from 2 to 30. To balance the precision and recall, the F1-measure is defined as

$$F1 = \frac{2*Recall*Precision}{Recall+Precision}$$

The outcome of data reduction for bug triage considered in two parts, firstly the size of data sets and secondly the quality of bug triage. The size of data sets includes the number of bug reports and words are treated as input given in Algorithm 1. The accuracy of instance and feature selection for a bug triage algorithm 1 is shown in Table 1 and Figure 3 shows graph which shows accuracy.

Table 1: Accuracy of training set of eclipse

List size	Origin	CHI	ICF	CHI→ICF	ICF→CHI
1	25.83	30.91	21.18	25.22	**27.23**
2	35.71	43.21	31.85	38.18	**40.15**
3	41.76	50.80	38.09	46.73	**48.57**
4	45.02	55.16	42.17	51.61	**53.45**
5	47.68	58.66	45.29	55.82	**57.29**
6	50.15	61.58	47.96	58.93	**60.09**
7	52.58	63.54	50.22	60.75	**62.00**
8	54.58	65.22	52.31	62.88	**63.92**
9	56.53	67.35	53.91	64.85	**65.64**
10	57.92	68.66	55.23	65.90	**66.95**

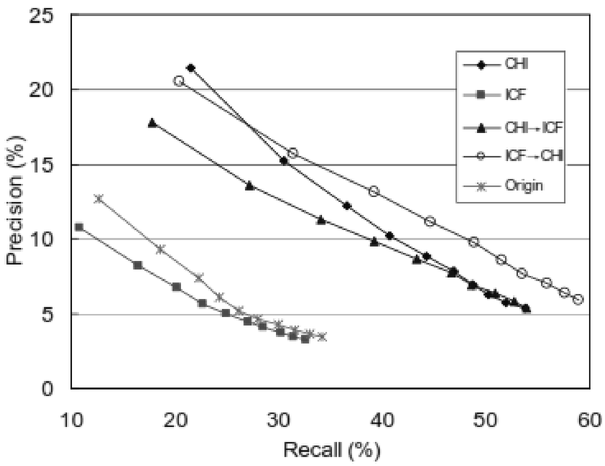

Figure 3: Precision and recall after combining CHI and ICF

Here, Whole System taken many more attribute for the input purpose but here we mainly focuses on the Time and performance of system as given in Table 2 and shown in Figure 4 graphically.

Table 2: Proposed System

Algorithms	Accuracy	Searching time in ms
Instance Selection	3.6	2.4
Feature Selection	2.5	3.4
SVM	3.5	2.8
AES	4.5	2.8

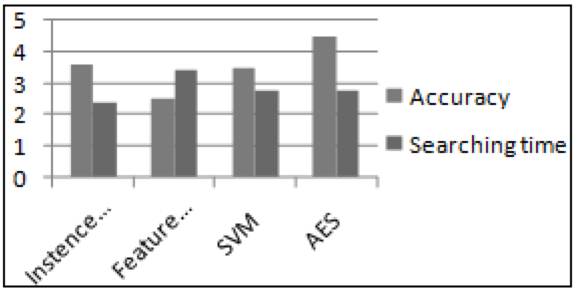

Figure 3: Accuracy and Searching Time

7. Conclusion

Bug triaging is step of software maintaining projected systems that aims to perform reduction and high superiority bugs information in software development and maintenance. Bug triage is expensive for software maintenance in both labor and time. The feature selection with instance selection has been merged to reduce the quantity of bug data sets as well as get enhanced the data quality. The projected system is used for any open supply comes that generate immense bug knowledge. Various software corporations engaged on comes like banking, food chain management will use the applying of the projected system. The advantage of proposed system is, it uses the combination of feature and instance selection to minimize the level of bug data sets and improve the data quality. The next advantage is, it provide priority according to severity of bug and security so that no another developer can access

Acknowledgement

I would like to articulate my profound thankfulness and deep stare to my guide Prof. Shyam Gupta for his excellent direction, valuable advice and regular support all the way through the period of the project. His precious advices were of enormous help all the way through my project effort. His sensitive analysis kept me operational to build this project in a much enhanced way. Working with him was an tremendously knowledgeable experience for me. Also I am very much thankful to our HOD Computer Engineering, SCOE, Sudumbare, Pune.

References

[1] J. Xuan, H. Jiang, Y Hu, Z Ren, W. Zou, Z. Luo, and X. Wu, "Towards Effective Bug Triage with Software Data Reduction Techniques", IEEE transactions on Knowledge and Data Engineering,, 2015

[2] Holmes, R., Godfrey, M. W. & Baysal, O., "Revisiting bug triage and resolution practice" In *User Evaluation for Software Engineering Researchers (USER), 2012* IEEE.

[3] Zimmermann T., Jeong, G. & Kim, S., "Improving bug triage with bug tossing graphs" in p*roceedings of the the 7th joint meeting of the European software engineering conference and the ACM SIGSOFT symposium on The foundations of software engineering* ACM.

[4] S. Christian Muller, T. Fritz, K. Katja, and Harald C. Gall. "Collaborative bug triaging using textual similarities and change set analysis", In *Cooperative and Human Aspects of Software Engineering (CHASE), 6th International Workshop on*, IEEE, 2013

[5] H. Jiang, Z. Ren, J. Yan, and Z. Luo, Xuan, Jifeng, "Automatic Bug Triage using Semi- Supervised Text Classification" in *SEKE*, , 2010..

[6] Z. Ren, J. Yan, and Z. Luo, Xuan, Jifeng, He Jiang, "Automatic Bug Triage using Semi- Supervised Text Classification" in *SEKE*, 2010.

[7] K. Magel, and S. Banitaan, Alenezi, Mamdouh, "Efficient bug triaging using text mining." *Journal of Software* (2013): 2185-2190.

[8] Y. Hu, J. Xuan, and H. Jiang, Zou, Weiqin, "Towards training set reduction for bug triage." In *Computer Software and Applications Conference (COMPSAC), IEEE 35th Annual*, IEEE, 2011.

[9] E. Giger, A. Lamkanfi, S. Demeyer, and B. Goethals, “Predicting the severity of a reported bug,” in Proc. 7th IEEE Working Conf. Mining Softw. Repositories, May 2010.

[10] T. Xie, J. Anvik, X. Wang, L. Zhang, and J. Sun, “An approach to detecting duplicate bug reports using natural language and execution information,” in Proc. 30th Int. Conf. Softw. Eng., May 2008.

Author Profile

Vijay N Kukre received the BE in Computer Technology from Nagpur University in 1993 and pursuing ME degree in Computer Engineering from Siddhant College of Engineering Pune

Prof. Shyam S. Gupta received the BE, ME and pursuing PhD Degree in Computer Science and Engineering. He is Associate Professor in Computer Engineering Department of Siddhant College of Engineering, Sudumbare, Pune affiliated to Savitribai Phule Pune University India