Analysis and Design of Data Mining Techniques to Develop Software Reliability Ch. Kishore Kumar, Research Scholar, Department of Computer Science, (VISTAS), Chennai, India. E-mail:...

Article *in* Journal of Advanced Research in Dynamical and Control Systems · February 2023

CITATIONS
0

READS
36

3 authors, including:

Dr KISHORE Kumar
Vaagdevi College of Engineering
16 PUBLICATIONS **73** CITATIONS

SEE PROFILE

Analysis and Design of Data Mining Techniques to Develop Software Reliability

Ch. Kishore Kumar, Research Scholar, Department of Computer Science, (VISTAS), Chennai, India.
E-mail: kishore.chennuri@gmail.com

Dr.R. Durga, Assistant Professor, Dept. of Computer Science, (VISTAS), Chennai, India. E-mail: drrdurgaresearch@gmail.com

Abstract--- The main aim of software development is to develop high quality software and high quality software is developed us in vast amount of software engineering data. The software engineering data can be used to gain empirically based understanding of software development. Software is ubiquitous in our daily life. It brings us great convenience and a big headache about software reliability as well: Software is never bug-free, and software bugs keep incurring monetary loss of even catastrophes. Data mining techniques are applied in building software fault prediction models for improving the software quality. Early identification of high-risk modules can assist in quality enhancement efforts to modules that are likely to have a high number of faults. This paper presents the data mining algorithms and techniques most commonly used to produce patterns and extract interesting information from software engineering data. The techniques are organized in seven sections: classification trees, association discovery, clustering, artificial neural networks, optimized set reduction, Bayesian belief networks, and visual data mining can be used to achieve high software reliability. The Reliability of any software utility software program is becoming so crucial in our everyday existence need; mistakes of the software application sadly continue to be common place to purpose machine disasters. In software program application software development the most time ingesting and hard challenge is to discover insects and join them. Then for solving this difficult trouble it is probably substantially advocated if we take a look at the bud and apprehend their conduct and their tendencies then stumble on them robotically. Inside the software that consists of a massive code of statistics and in files. It is difficult for the builders to analyse the facts and find them. We're coming close to a facts mining approach to extract a useful understanding inside the large software program application and contribute this records for laptop virus detecting.

Keywords--- Data Mining, Reliability, Software Fault Analysis, Defect Classification, SVM Techniques, DM Software, Knowledge Discovery in Databases, Data quality.

I. Introduction

Data mining is the process of extracting useful data ior knowledge from a scattered idata and it is a process that employs various analytic tools to extract patterns and information from large datasets. Today, large numbers of datasets are collected and stored. Human are much better at storing data than extracting knowledge from it, especially the accurate and valuable information needed to create good software. Large datasets are hard to understand, and traditional techniques are infeasible for finding information from those raw data. Data mining helps scientists in hypothesis formation in physics, biology, chemistry, medicine, and engineering. There are few steps of at a mining, data integration, data cleaning, data selection, data transformation, data mining, pattern evaluation and knowledge presentation. Data mining techniques that can be applied in improving SE include generalization, characterization, classification, clustering, associative tree, decision tree or rule induction, frequent pattern mining. The purpose of this study is to explore how data mining techniques can be applied to improve Software Reliability. Objectives of this study are:

i. To review the concept of Software Reliability and data mining.
ii. To determine the problems in achieving the Software Reliability.
iii. To identify data mining techniques that can be applied to achieve high Software Reliability.

1.1. Software Reliability

Ability of a computer program to perform its intended functions and operations in a system's environment, without experiencing failure (system crash).

- Mining for reliability:-
- Making a software free from errors
- System learning automatically to give better efficiency.

DOI: 10.5373/JARDCS/V12I4/20201429
ISSN 1943-023X

Received: 14 Feb 2020/Accepted: 10 Mar 2020

- Detecting the bugs that come through the network.
- Identifying the accuracy of the software.
- Type of the failures.

Data mining is a process that employs various analytic tools to extract patterns and information from large datasets. Today, large numbers of datasets are collected and stored. Human are much better at storing data than extracting knowledge from it, especially the accurate and valuable information needed to create good software. Large datasets are hard to understand, and traditional techniques are infeasible for finding information from those raw data. Data mining helps scientists in hypothesis formation in biology, physics, chemistry, medicine, and engineering. The data mining process is shown in Figure.1.

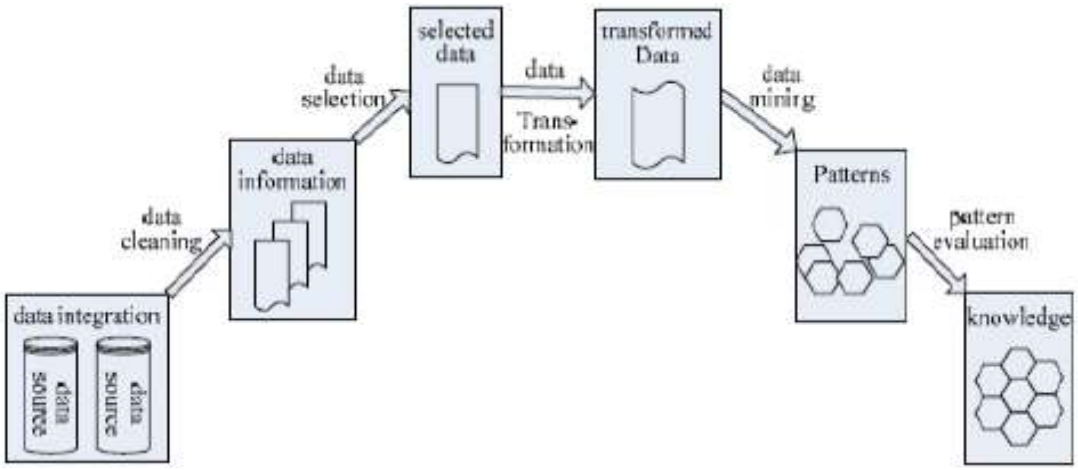

Figure 1: Data Mining Process

Software security is software reliability enlist hacker expertise, but stay with academic fault naming conventions, when defending against the risk of exploitation of vulnerabilities and intrusions Software reliability states for the probability that software will not cause the failure of a system for a specified time under specified conditions in specific environment. The probability is a function of the inputs to and use of the system as well as the associated functions of the subsistence of any malicious events in software under consideration. The data input estimate the probability of existence of any malicious events. The software reliability states for the likelihood that considered software exhibits its expected functions without causing any malfunctioning in defined time duration.

The purpose of this study is to explore how data mining techniques can be applied to improve Software Reliability. Objectives of this study are:

a) To review the concept of Software Reliability and data mining.
b) To determine the problems in achieving the Software Reliability.
c) To identify data mining techniques that can be applied to achieve high Software Reliability.

In modern society, computers are used for many different applications, such as nuclear reactors, aircraft, banking systems, and hospital patent monitoring systems. As the demand of the application quality becomes higher and higher, the research of the computer software reliability becomes more and more essential. Size, complexity, and human dependency on software-based products have grown dramatically during past decades. Software developers are struggling to deliver reliable software with acceptable level of quality, within given budget and schedule. Reliability of software has become increasingly important, especially for server applications that require high availability. System failures can degrade system performance, crash systems and corrupt important data, which would significantly reduce system availability and lead to huge loss in productivity and business.

1.2 Problems Encountered by Delivered Software Include

a. Errors detected after delivery.
b. Requirements not translated correctly.
c. Product not complete.
d. Difficulty in supporting software.
e. Apparent lack of a systematic process.

DOI: 10.5373/JARDCS/V12I4/20201429
ISSN 1943-023X

Received: 14 Feb 2020/Accepted: 10 Mar 2020

II. Literature Review

Magazinius & Feldt (2011) have verified intentional distortions of estimates reported in earlier study. This study was based on questionnaire responses from 48 software practitioners from eight dissimilar companies. The effects of the questionnaire propose that, presence of intentional distortions was affected by the organizational type and the software development process in use. The effects also propose that presence of distortions differs depending on the organizational factors, such as company type and development process (agile or plan-driven). These differences were further investigated by including more responses symmetrically divided between the comprised companies. Moreover, a future questionnaire should comprise information on age, experience and role as it might cause the high standard variations in the effects of this study.

Lee et al (2010) have performed the calibration of the COCOMO II model by means of data gathered from the defense software development projects by employing Calico calibration software. The effect has pointed out that calibration could improve the precision of the original estimation model. They offer the effect of the empirical study on improving the precision of software cost estimation model for defense software development project applications. The results showed that estimation precision has improved after calibration. In addition they found that the performance of calibrating the multipliers and exponent as both was much better than calibrating multipliers only. The cause of the lower calibration performance and future works to improve the estimation precision is also discussed.

Tsunoda et al (2013) have spotlighted on an effort estimation method based on early phase activities and its ratio to the effort of the whole improvement process, and it assesses estimation precision of the model by means of early phase effort. To attain the goal, they built effort estimation models by means of early phase effort as a descriptive variable, and compared the estimation accuracies of these models to the effort estimation models based on software size. In their test, they employed International Software benchmarking Standards Group (ISBSG) dataset, which was gathered from software development companies, and considered planning phase effort ‟and,, requirement study effort ‟as early phase effort. The effect of the experiment demonstrated that when both software size and sum of planning and requirement study phase effort were employed as descriptive variables, the estimation precision was most enhanced.

In all aspects of project management, Global software project development needs new approaches. Azzeh (2013) has explored the applicability of Use Case Point estimation model to global software project development. This kind of projects was appropriate when local cost rate was high and the demand to release software to the market rapidly. This study has also demonstrated and emphasized the feasible significance role of UCP software cost estimation model in multi-site development. It examines the potential of Use Case estimation model for global projects and employs this as a basis to converse three suggested factors Global team trust, Global team composition and Culture value that will assist in managing the global software project development. Future extension of the suggested model was designed to reflect on the result of the different complexity factors.

Keung et al (2008) have suggested a means of software cost estimation as a substitute to other data-intensive methods such as linear regression. Disappointingly, there were disadvantages to the method. There was no mechanism to evaluate its suitability for a particular data set. Additionally, heuristic algorithms were essential to choose the best set of variables and recognize unusual project cases. They bring solution to these problems based on the use of the Mantel's correlation randomization test called Analogy-X. They employ the strength of correlation among the distance matrix of project features and the distance matrix of famous effort values of the data set.

III. Methodologies for Data Mining Software Engineering

a. Association rule: Association rule learning is a popular and well researched method for discovering interesting relations between variables in large databases. Association Rule mining techniques is applied to the records in order to discover the patterns that are likely to cause high severity defects [6]. The discovered patterns are then helpful to predict the subsequent actions that may result in high severity defects. The concept of association in mining software engineering data is based on set of strong rules. The computational cost of association rules mining in software development can be reduced by reducing the number of passes over the database, sampling the software database, adding extra constraints on the structure of patterns, through parallelization [7]. Association mining searches for strong and interesting relationships among a large set of data items. After discovery of interesting association relationships among huge amounts of software and cost records, it can be helpful in a many decision making process, such as catalog design, cross- marketing, and loose-leader analysis [8]. An example of association rule mining is market based analysis and requirement

DOI: 10.5373/JARDCS/V12I4/20201429
ISSN 1943-023X

Received: 14 Feb 2020/Accepted: 10 Mar 2020

analysis for developing new software. This process analyzes customer requirements and interests by association from past interactions with customers and products of competitors. Association rule based algorithms are developing swiftly. Algorithms such as apriority algorithm, FP-growth algorithm and OPUS search are quite helpful in generating association rules that are utilized in software engineering.

b. Classification: Classification is one of the main tasks in data mining for assigning a data item to a predefined set of classes. Classification can be described as a function that maps (classifies) a data item into one of several predefined classes. Here the goal is to induce a model that can be used to classify future data items with unknown classification into unique classes. In software development process the performance of depends upon the type and class of data. There are different forms of data available in software engineering. It is imported to work with relevant data items and classify them into sub classes and keep on adding new data items into pre existing classes. We implement different classification algorithms in software engineering to solve various problems in different phases. Classification can be used to identify the types of bugs and thus helps in building bug detector. Decision trees are important tools in classification technique that helps in identifying the risky modules in software depending upon the attributes of system and modules. Classification and assignment can sometimes be automated, but are often done by humans, especially when a bug is incorrectly filed by the reporter or the bug database.

c. Clustering: Cluster analysis is a group of multivariate techniques whose primary purpose is to group entities based on their attributes. Similar objects are placed in the same cluster according to predetermined selection criteria. The objective of any clustering algorithm is to ort entities into groups, so that the variation between lusters is maximized relative to variation within clusters [10]. The set of entities to cluster needs to be identified, before applying clustering to a software system. The next phase is attribute selection. Most software clustering methods initially transform a fact base to a data table, where each row describes one entity to be clustered. Each column contains the value for a specific attribute. After completion of all preparation steps the clustering algorithm can start to execute. Clustering algorithms used in software engineering are: graph-theoretical algorithms, construction algorithms, optimization algorithms, hierarchical algorithms. For high dimensional data, many of the existing methods fail due to the curse of dimensionality, which renders particular distance functions problematic in high-dimensional spaces which led to new era of clustering algorithms for high-dimensional data that focus on subspace clustering and correlation clustering that also looks for arbitrary rotated subspace clusters that can be modelled by giving a correlation of their attributes. Examples for such clustering algorithms are CLIQUE. Several different clustering systems based on mutual information have been proposed. One is Marina Meilă's variation of information metric another provides hierarchical clustering.

d. Text mining: Approximately 80% of information is stored in computers is in form text. Example of software engineering text data includes project and bug reports, e-mails and code comments Text mining is an area of data mining with extremely broad capability. Rather than requiring data in a very specific format such as numerical data, database entries, text mining can discover previously unknown information from textual data. As many artefacts in software engineering are based on text, there are abounded sources of data from which information may be extracted. There are several applications of text mining and their implications for software development processes. Code duplication is one of the biggest problems which complicates maintenance and evolution of software systems. Several drawbacks of all existing code duplication techniques can be overcome by using visual approach which is language-independent. Although this approach requires no language-specific parsing, it is able to detect significant amounts of code duplication. Also duplication of bug reports is common in software development, it can be overcome using neural language processing with data mining [14]. Text data mining refers to the discovery of hidden information and potentially useful knowledge from a collection of texts which is done by automatic extraction and by analyzing information. A key factor is to extract the appropriate information and link it together to form new facts to be explored further. The Natural Language Description Technique combines computer science and linguistics to enhance the interactions between computers and natural languages.

Support vector machines**:-** Although SVMs have been used in various fields of study, the use of SVMs for gene expression analysis was described in detail by Brown et al. SVM is a supervised learning classification technique. The algorithm uses supplied information about existing relationships between members of a subset of the elements to be classified. The supplied information, an initial presumed between a set of elements, coupled with the expression pattern data leads to a binary classification of each element. Each element is considered either in or out of the initial presumptive classification. The algorithm proceeds through two main phases. The first of these phases, training, uses the presumptive classification (supplied knowledge) and the expression data as inputs to produce a set of weights which will be used during the next phase. The second phase, classification, uses the weights created during training

DOI: 10.5373/JARDCS/V12I4/20201429
ISSN 1943-023X

Received: 14 Feb 2020/Accepted: 10 Mar 2020

and the expression data to assign a discriminant or score to each element. Based on this score each element is placed into or out of the class.

The initial dialog is used to define the basic SVM mode. One can select to classify genes or experiments and can select to perform one or both phases of the algorithm. The Train and Classify option allows one to run both phases of the algorithm. Starting with a presumptive classification and expression data the result is a final classification of each element. The Train only option produces a list of weights which can be stored as an 'SVM' file along with training parameters so that they can be applied to data to classify at a later time. The Classify only option prompts the user for an SVM file of weights and parameters and results in final classification. The user also has an option to produce hierarchical trees on the two resulting sets of elements. The second dialog is used during either the Train and Classify mode or the Train Only mode. The upper portion is used to indicate whether the initial presumptive classification will be defined using the SVM Classification Editor or supplied as an SVC file.

Block Diagram

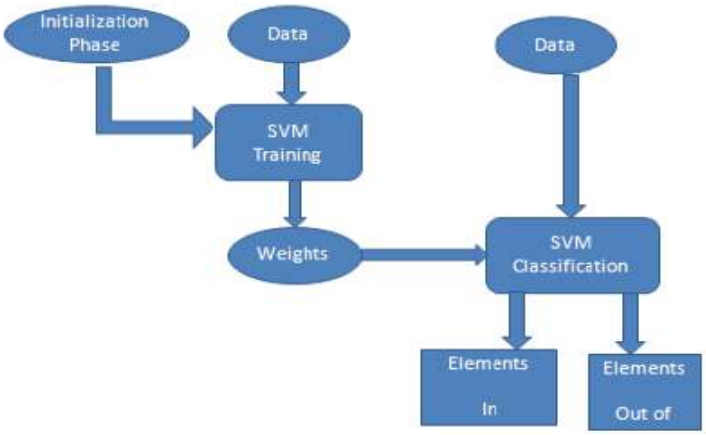

Fig. 2: SVM Process

IV. Proposed System

The evolutionary support vector machine (ESVM) is an enhanced or optimized form of generic support vector machine approach and it represents optimized algorithms for training to leam various classification as well as regression rules from datasets under consideration.

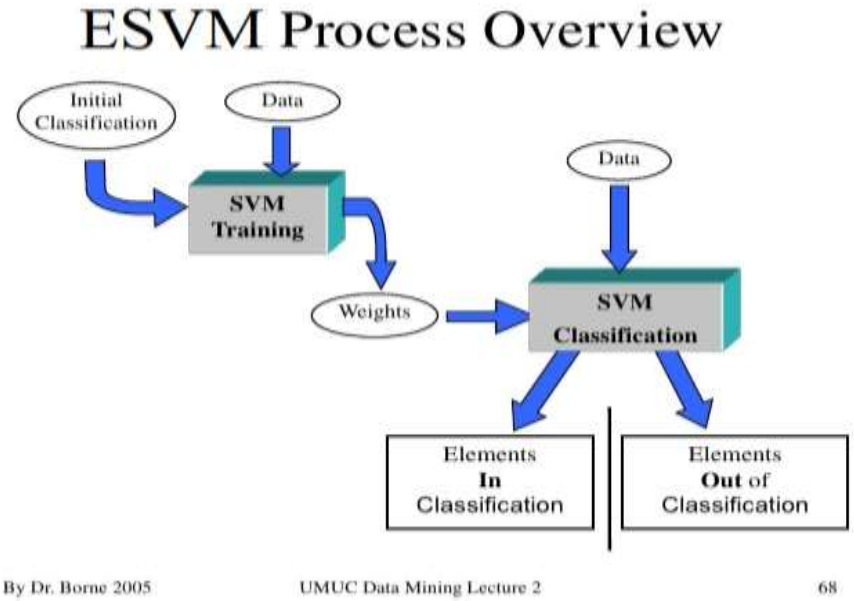

Figure 3: Block Diagram for ESVM

DOI: 10.5373/JARDCS/V12I4/20201429
ISSN 1943-023X

Received: 14 Feb 2020/Accepted: 10 Mar 2020

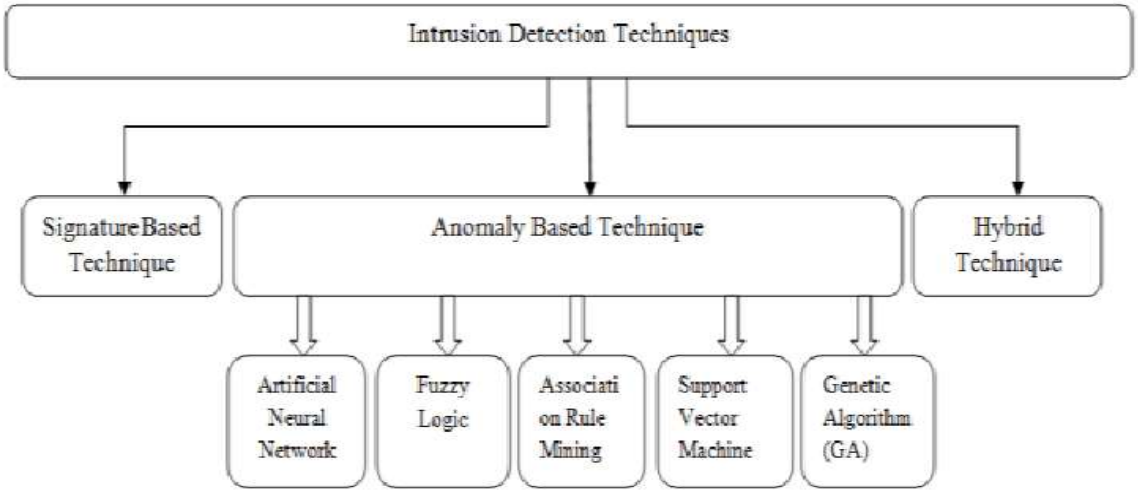

Figure 4: Block Diagram of IDS

As for illustration, the Evolutionary Support Vector Machine (ESVM) can be potentially employed for learning various classifier approaches such as polynomial classifier, radial basis function (RBF) based classifiers and multi-layer perceptrons (MLP) kinds of classifiers. Inceptionally,

The evolutionary SVM (ESVMs) were first recommended by for exhibiting data classification and possess lately become a vicinity of strenuous investigate in the red for enhancements in the approaches and speculation together with conservatories to exhibit regression and estimation of density.

As mentioned earlier that has been mentioned and shown the proposed approach is achievable. In this section, the implementation of the proposed neural-network-based models will be described. The following steps are required to apply approach for software reliability prediction. Analyze the assumptions of the software project and if there is a suitable model, select the model. If there is no suitable model, select those models which partially meet the assumptions of project. Construct the neural network of selected models by designing the activation functions and bias.

V. Conclusion

Software reliability is one of the most predominant factors now days which is being considered seriously to ensure quality services and faultless function by certain software applications. There are a number of applications which run on huge data sets and in such application the presence of malicious data is common.

In order to eliminate the system failure the detection of such malicious nodes or datasets becomes compulsory. The intrusion detection scheme (IDS) with certain robust data mining scheme can be a potential approach to ensure software reliability and undisrupted operations.

In this research work and hence developed thesis work, an effective evolutionary approach based support vector machine algorithm for data mining has been proposed and developed that not only ensures efficient intrusion detection but also makes the system reliable to functions with huge datasets.

The proposed system called Evolutionary Support Vector Machine (ESVM) has exhibited better results as compared to Neural network (NN) based data mining scheme. The enhancement of generic support vector machine has made this system more robust and efficient by means of incorporation with evolutionary computing schemes.

The proposed ESVM approach has been enhanced with an automatic internal relevant feature detection facility and parametric tuning employing evolutionary computing paradigm which has been enriched with estimator of generalization capability.

VI. Future Work

The system enhancement can be further explored with optimized evolutionary computing approaches and enhanced artificial intelligence based systems. The compatibility of the reliability model with varied attacks types and applications scenario can be explored.

DOI: 10.5373/JARDCS/V12I4/20201429
ISSN 1943-023X

Received: 14 Feb 2020/Accepted: 10 Mar 2020

Considering the present needs and applications the proposed ESVM system has performed better for reliability assurance and thus the proposed research work has been accomplished successfully with its optimum efficiency.

References

[1] Yi (Cathy) Liu, Taghi M. Khoshgoftaar, and Naeem Seliya: Evolutionary Optimization of Software Quality Modeling with Multiple Repositories, *IEEE Transactions on Software Engineering*, Vol, 36, No. 6, November/December 2010, pp. 852-864.

[2] Qinbao Song, Martin Shepperd, Michelle Cartwright, and Carolyn Mair: Software Defect Association Mining and Defect Correction Effort Prediction, *IEEE Transactions on Software Engineering*, Vol. 32, No. 2, February 2006, pp. 69-82.

[3] Lessmann, S. Baesens, B. Mues, C. Pietsch, S.: Benchmarking Classification Models for Software Defect Prediction; A Proposed Framework and Novel Findings, *IEEE Transactions on Software Engineering*, Vol. 34, No. 4, July 2008, pp. 485-496.

[4] J.C. Munson, T.M. Khoshgoftaar: The Detection of Fault-Prone Programs, *IEEE transactions on Software engineering*, May 1992, Vol. 18, No. 5, pp. 423-433.

[5] Ohlsson, N., Helander, M., and Wohlin, C.: Quality improvement by identification of faultprone modules using software design metrics, Sixth International Conference of Software Quality, Ottawa, Canada, 1996, pp. 1-13. PhD Thesis Page 131 Data mining techniques for software reliability

[6] Benjamin Livshits, Thomas Zimmermann. "Dynamine: Finding Common Error Patterns by mining Software Revision Histories".

[7] Zhenmin Li and Yuanyuan Zhou. "PR- Miner: Automatically Extracting Implicit Programming Rules and Detecting Violations in Large software Code", Dept. of Computer Science, University of Illinois at Urbana-Champaign, Urbana, USA.

[8] Ray –Yaung Chang, Andy Podgurski and Jiong Yang. "Finding what's not there: A New Approach to Revealing Neglected conditions in Software."

[9] R. Durga, Dr.P. Sudhakar. "Design of a Wireless Transfer for a Secure Signal of Sender and Receiver System through the Network*". International Journal of Applied Engineering Research*, Issn 0973-4562 Vol. 10 No.82 (2015), © Research India Publications Httpwww.Ripublication.Comijaer.Htm.

[10] R. Durga, Dr.P. Sudhakar, "Cryptographic Approach for Data Transfer Using Protocols". *International Journal of Advances in Engineering Research* Http://Www.Ijaer.Com (Ijaer) 2015, Vol. No. 10, Issue No. Vi, December E-Issn: 2231-5152/ P-Issn: 2454-1796, 208.

[11] R. Durga, Dr.P. Sudhakar. "Recent Developments in Progress on Network Security Using Cryptography And Wireless Security". *Jour of Adv Research in Dynamical & Control Systems*, Vol. 9, No. 4, 2017

[12] R. Durga, M. Prem Kumar. "Analysis and Research on Integrated Multi Model Wireless Sensor Adhoc Network in Embedded Tracking Technology", Jasc: *Journal of Applied Science and Computations* Volume V, Issue Xii, December/2018, Issn No: 1076-5131.

[13] Dr.R. Durga, B. Vinothini, "A Survey of Enhancing and Developing Multi Block Proxy Re-Encryption Methodology In Network Security". *International Journal of Information and Computing Science*, Volume 6, Issue 5, May 2019 222, Issn No: 0972-1347.

[14] Dr.R. Durga, B. Vinothini, "Enhancing and Developing Multi Block Proxy Re-Encryption Methodology in Network Security". *International Journal of Recent Technology and Engineering close*, Vol. No. 8, Issue 2s11 Sep 2019, Issn: 2277-3878.

[15] Munson, T.M. Khoshgoftaar: The Detection of Fault-Prone Programs, *IEEE transactions on Software engineering*, May 1992, Vol. 18, No. 5, pp. 423-433.

DOI: 10.5373/JARDCS/V12I4/20201429
ISSN 1943-023X

Received: 14 Feb 2020/Accepted: 10 Mar 2020