A NOVEL APPROACH TO IMPROVE SOFTWARE DEFECT PREDICTION ACCURACY USING MACHINE LEARNING

Mohd Ismail Ali Saffan[1], Md Sharoze E Akbar[2], Syed Faisal Ahmed[3], Dr Md Zainlabuddin[4]
[1,2,3]B. E Student, Department of CSE, ISL College of Engineering, India.
[4]Associate Professor, Department of CSE, ISL College of Engineering, Hyderabad, India.

ABSTRACT: Defect prediction is a prominent field within the software engineering community. In order to ensure the program's success, it is crucial to minimize the disparity between software engineering and data mining. Software defect prediction anticipates the occurrence of source code problems prior to the testing process. Various techniques, including clustering, statistical approaches, mixed algorithms, neural network-based metrics, black box testing, white box testing, and machine learning, are often used to forecast software problems and analyze their impact. This study introduces the novel use of feature selection to enhance the accuracy of machine learning classifiers in predicting faults. The aim of this project is to enhance the precision of defect prediction in five NASA datasets, namely CM1, JM1, KC2, KC1, and PC1. The NASA data sets are publicly accessible. This research employs feature selection technique in conjunction with various machine-learning techniques, namely Random Forest, Logistic Regression, Multilayer Perceptron, Bayesian Net, Rule ZeroR, J48, Lazy IBK, Support Vector Machine, Neural Networks, and Decision Stump. The objective is to enhance defect prediction accuracy significantly when compared to the scenario where feature selection is not applied (WOFS). The research workbench utilizes a machine-learning program known as WEKA (Waikato Environment for Knowledge Analysis) to enhance and preprocess data, as well as implement the specified classifiers. A tiny tab statistics tool is used for evaluating statistical studies. The research findings indicate that the accuracy of defect prediction is enhanced while using feature selection (WFS) compared to the accuracy of WOFS.

INTRODUCTION

A fault in a software system refers to an unforeseen lack of performance in meeting a client's requirements. Software testers often see this atypical behavior in software. Software testers detect flaws in the software testing process. The term "software fault" is sometimes used to refer to deviations in the software development process that often lead to software failure and do not meet user expectations. [1]. A defect refers to the absence of perfection resulting from an error, malfunction, or failure in the process or product of software development. In the paradigm, 'error' is defined as human conduct that results in unsuitable consequences, while 'defect' refers to a judgment that leads to erroneous outcomes while attempting to address a problem.

The technique of software defect prediction entails identifying faulty modules and fulfilling several testing prerequisites. Developing an effective defect prediction model in software engineering is a very challenging task. Such a model aims to anticipate software modules or flaws that may occur at the early stages of the software development life cycle. Examining the source code, doing beta testing, performing integration testing, system testing, and unit testing are all sequential stages in the conventional procedure for identifying software defects. Consequently, doing these tests becomes difficult when the program grows in size, complexity, and the amount of source code [2].

Software defect prediction has gained popularity in recent years. The ability to forecast software errors has a direct influence on the overall quality of the product. Malfunctioning software modules have a substantial effect on the product's quality, resulting in cost overruns, a prolongation of the software's completion schedule, and heightened maintenance expenses [3].

Defect detection and defect prevention are the primary approaches of software quality assurance. The objective of defect prevention is to promptly mitigate probable flaws. Defect prediction aims to identify and solve existing defects. The study conducted by Memon et al. [1] is on improving software quality by preventing defects. Our research intends to enhance software quality by predicting and anticipating problems. Defect prevention activities include tasks such as algorithm design, algorithm execution assessment, and identification of flaws in software planning [4]. Before the deployment of a software product, the fundamental objective of defect prediction is to foresee and forecast any faults, mistakes, or defects in the program in order to estimate the required maintenance work and ensure the quality of the final product [5]. The defect prevention strategy is used to enhance the quality of software [1]. Anticipating faults is an essential stage in developing high-quality software. Software deployment is done before defect prediction in order to enhance the overall performance of the system and guarantee user happiness. Timely identification of mistakes or flaws leads to efficient allocation of resources, resulting in decreased time and cost, as well as the production of a superior output. Consequently, software defect prediction models play a significant role in assisting individuals in understanding how to assess software and enhance its quality [6].

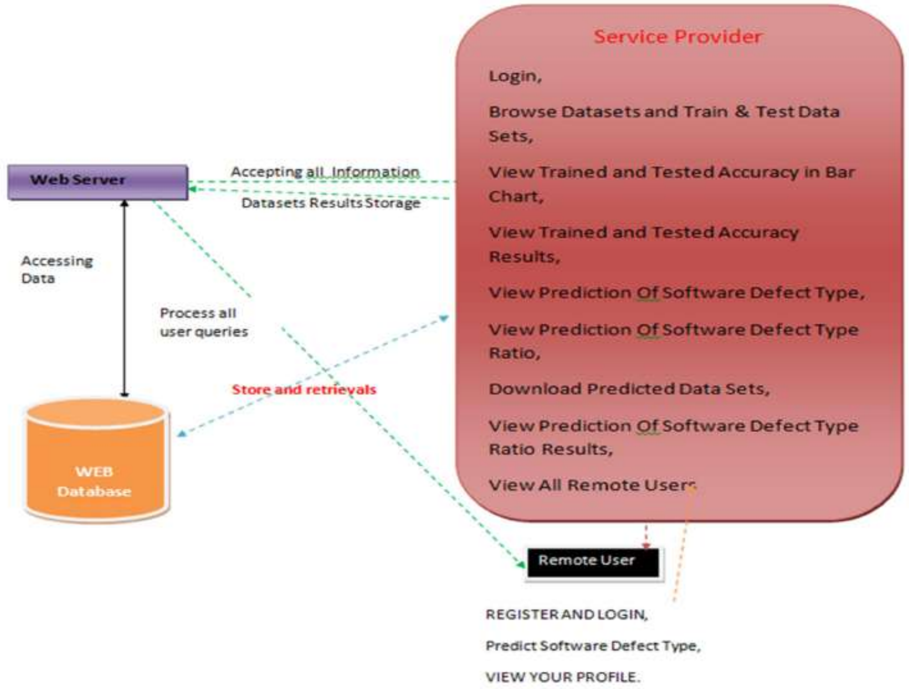
The effectiveness of the software-testing phase is enhanced by the defect-prediction method, which detects software modules that have issues. By using effective defect prediction methodologies or models, several ways and approaches have yielded exceptional outcomes. It is essential to integrate a proficient fault prediction model with a triumphant measuring system [7]. Predicting software failures enables the deployment of high-quality software that satisfies users. Code review is a commonly used software-quality assurance approach for identifying software defects [8]. Various techniques have been used to address problems or uncertainties related to software failure prediction. The literature review discusses many methodologies for defect prediction, but no one strategy is universally applicable to all datasets. The reason for this variation is contingent upon the specific attributes of the dataset. Selecting the optimal approach for fault prediction might provide a challenge. Defect prediction is most effectively accomplished with Machine Learning [9]. Defect prediction methods (DPT) are used throughout the software development life cycle (SDLC) to proactively mitigate faults in software products [10].

Machine learning algorithms, when applied to certain data sets, enable IT systems to efficiently identify various patterns. Furthermore, the results generated by machine learning rely on pre-existing knowledge of pertinent information [11]. Systems today possess the capacity to autonomously acquire knowledge and improve their performance by analyzing prior experiences. Machine learning is a field of study that involves teaching computers to acquire knowledge or information from data or prior experience, identify patterns within the data, and then make decisions or assessments with little human involvement. The area is appealing since it allows you to expand on existing knowledge to get useful business rule logics and more. What distinguishes this from others? Nevertheless, the machine learning process is not simple. The significance of machine learning in the twenty-first century lies in its ability to facilitate ongoing learning from data and make predictions about the future. This is a robust assemblage of algorithms and models used across several sectors to optimize software operations and identify patterns and anomalies in data [12].

Machine learning operates in a manner akin to an individual's learning process. Machine learning enables people to make judgments based on acquired information [13]. It may be defined as the process of inferring a system's underlying patterns or structures using little previous information. Machine learning challenges include classification, grouping, and regression as examples [14]. By using diverse machine learning algorithms, a range of machine learning techniques may enhance the quality and efficiency of software [5]. In addition, a significant role in minimizing re-work is played by the practice of early software problem or defect forecasting, which enhances software quality [9].

Utilizing machine learning methods for software fault prediction has several benefits. It allows enterprises to prioritize their testing efforts, spend resources efficiently, and make well-informed choices on software quality. Through early identification of high-risk regions, developers may proactively resolve possible problems before they have a negative effect on end-users. This leads to enhanced customer satisfaction and decreased maintenance efforts. In this study, the authors provide a valuable contribution to the testing phase by enhancing the accuracy of a machine learning algorithm in order to improve its ability to forecast faults for the user.

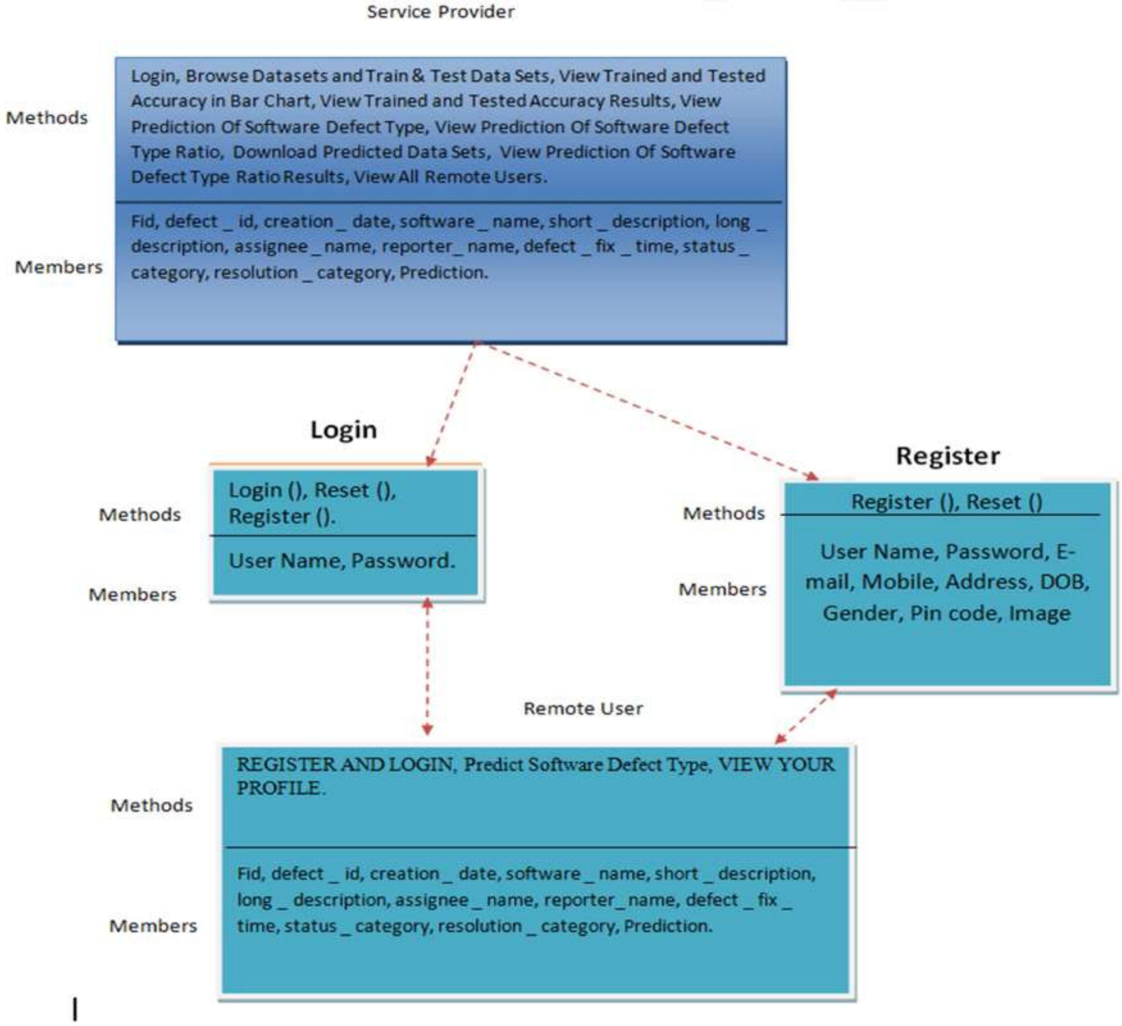
Architecture Diagram


Service Provider
Login,
Browse Datasets and Train & Test Data Sets,
View Trained and Tested Accuracy in Bar Chart,
View Trained and Tested Accuracy Results,
View Prediction Of Software Defect Type,
View Prediction Of Software Defect Type Ratio,
Download Predicted Data Sets,
View Prediction Of Software Defect Type Ratio Results,
View All Remote Users
Web Server
Accepting all Information
Datasets Results Storage
Accessing Data
Process all user queries
Store and retrievals
WEB Database
Remote User
REGISTER AND LOGIN,
Predict Software Defect Type,
VIEW YOUR PROFILE.

Proposed System

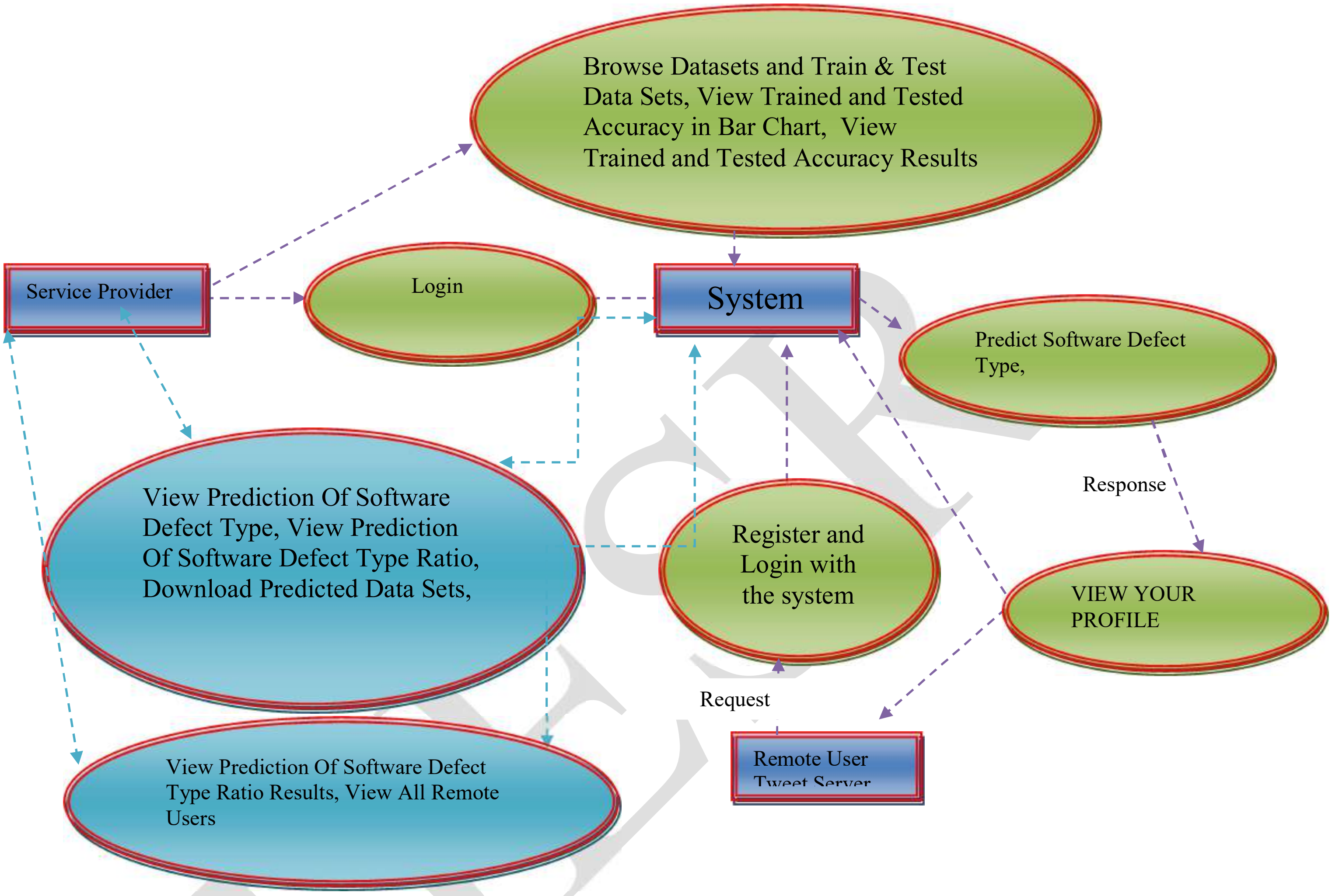
The main contribution of this research is the use of feature selection for the first time to increase the accuracy of machine learning classifiers in defects prediction. The objective of this study is to improve the defects prediction accuracy in five data sets. The machine-learning techniques used in this research are; Random Forest, Logistic Regression, Multi-layer Perceptron, Bayesian Net, Rule ZeroR, J48, Lazy IBK, Support Vector Machine, Neural Networks, and Decision Stump to achieve high defect prediction accuracy as com-pared to WOFS.

Class Diagram


Service Provider
Methods
Login, Browse Datasets and Train & Test Data Sets, View Trained and Tested Accuracy in Bar Chart, View Trained and Tested Accuracy Results, View Prediction Of Software Defect Type, View Prediction Of Software Defect Type Ratio, Download Predicted Data Sets, View Prediction Of Software Defect Type Ratio Results, View All Remote Users.
Members
Fid, defect _ id, creation _ date, software _ name, short _ description, long _ description, assignee _ name, reporter_ name, defect _ fix _ time, status _ category, resolution _ category, Prediction.
Login
Methods
Login (), Reset (), Register ().
Members
User Name, Password.
Register
Methods
Register (), Reset ()
Members
User Name, Password, E-mail, Mobile, Address, DOB, Gender, Pin code, Image
Remote User
Methods
REGISTER AND LOGIN, Predict Software Defect Type, VIEW YOUR PROFILE.
Members
Fid, defect _ id, creation _ date, software _ name, short _ description, long _ description, assignee _ name, reporter_ name, defect _ fix _ time, status _ category, resolution _ category, Prediction.

I

Data Flow Diagram :


Browse Datasets and Train & Test Data Sets, View Trained and Tested Accuracy in Bar Chart, View Trained and Tested Accuracy Results
Service Provider
Login
System
Predict Software Defect Type,
View Prediction Of Software Defect Type, View Prediction Of Software Defect Type Ratio, Download Predicted Data Sets,
Register and Login with the system
Response
VIEW YOUR PROFILE
Request
View Prediction Of Software Defect Type Ratio Results, View All Remote Users
Remote User
Tweet Server

Decision tree classifiers

Decision tree classifiers are used successfully in many diverse areas. Their most important feature is the capability of capturing descriptive decision making knowledge from the supplied data. Decision tree can be generated from training sets. The procedure for such generation based on the set of objects (S), each belonging to one of the classes C1, C2, …, Ck is as follows:

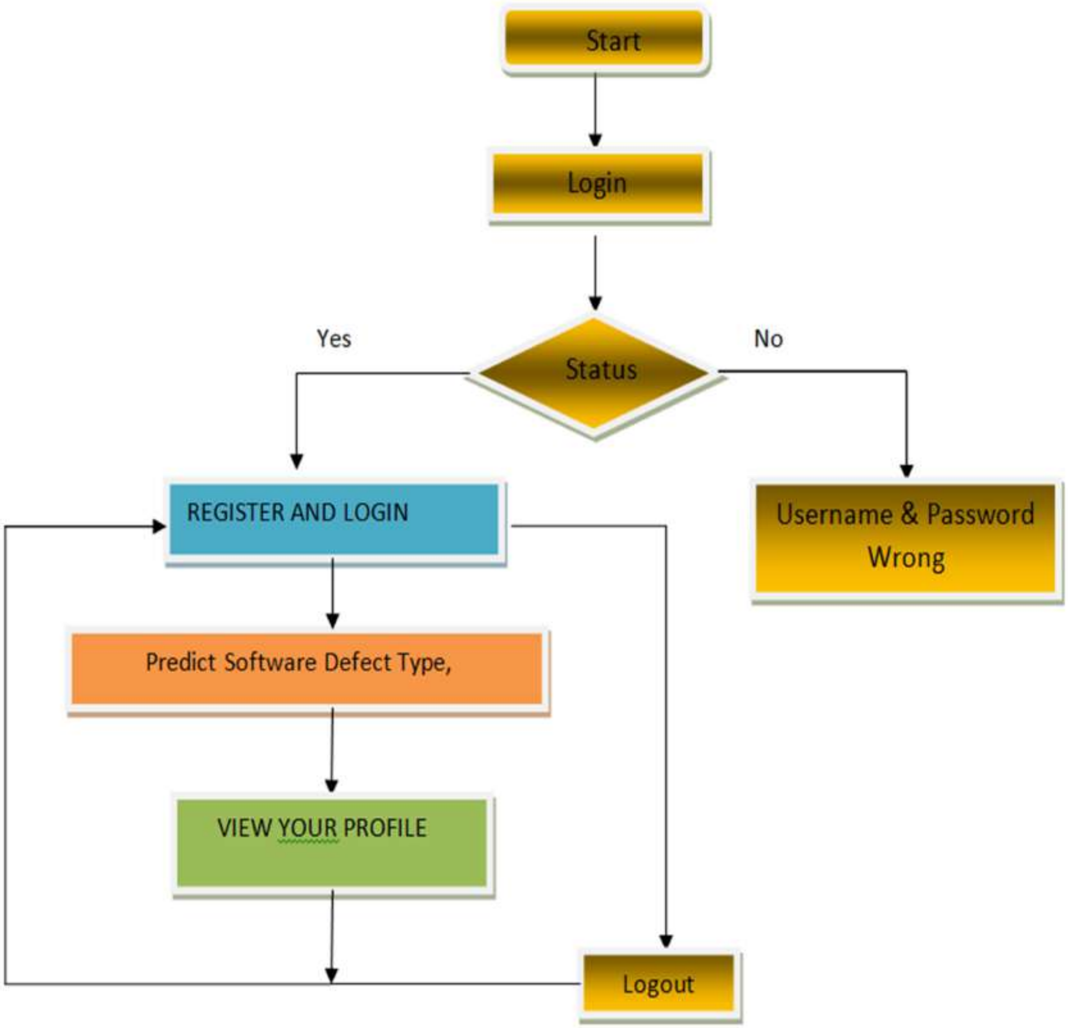
Step 1. If all the objects in S belong to the same class, for example Ci, the decision tree for S consists of a leaf labeled with this class

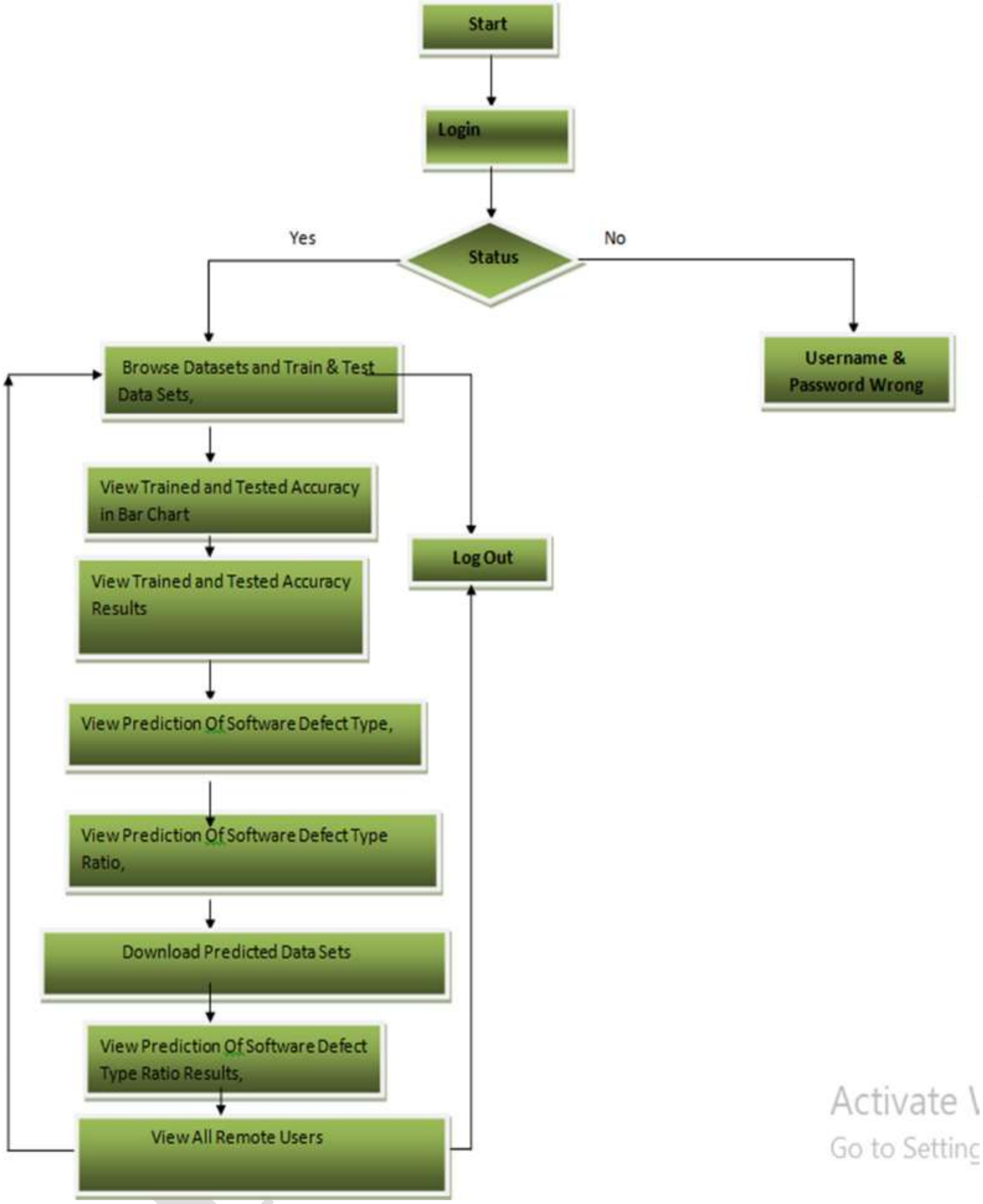
Step 2. Otherwise, let T be some test with possible outcomes O1, O2,…, On. Each object in S has one outcome for T so the test partitions S into subsets S1, S2,… Sn where each object in Si has outcome Oi for T. T becomes the root of the decision tree and for each outcome Oi we build a subsidiary decision tree by invoking the same procedure recursively on the set Si.

Gradient boosting

Gradient boosting is a machine learning technique used in regression and classification tasks, among others. It gives a prediction model in the form of an ensemble of weak prediction models, which are typically decision trees.[1][2] When a decision tree is the weak learner, the resulting algorithm is called gradient-boosted trees; it usually outperforms random forest.A gradient-boosted trees model is built in a stage-wise fashion as in other boosting methods, but it generalizes the other methods by allowing optimization of an arbitrary differentiable loss function.

Flow Chart : Remote User




Start
Login
Status
Yes
No
Username & Password Wrong
Browse Datasets and Train & Test Data Sets,
View Trained and Tested Accuracy in Bar Chart
View Trained and Tested Accuracy Results
View Prediction Of Software Defect Type,
View Prediction Of Software Defect Type Ratio,
Download Predicted Data Sets
View Prediction Of Software Defect Type Ratio Results,
View All Remote Users
Log Out
Activate \
Go to Settinc

SYSTEM TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

OTHER TESTING METHODOLOGIES

User Acceptance Testing

User Acceptance of a system is the key factor for the success of any system. The system under consideration is tested for user acceptance by constantly keeping in touch with the prospective system users at the time of developing and making changes wherever required. The system developed provides a friendly user interface that can easily be understood even by a person who is new to the system.

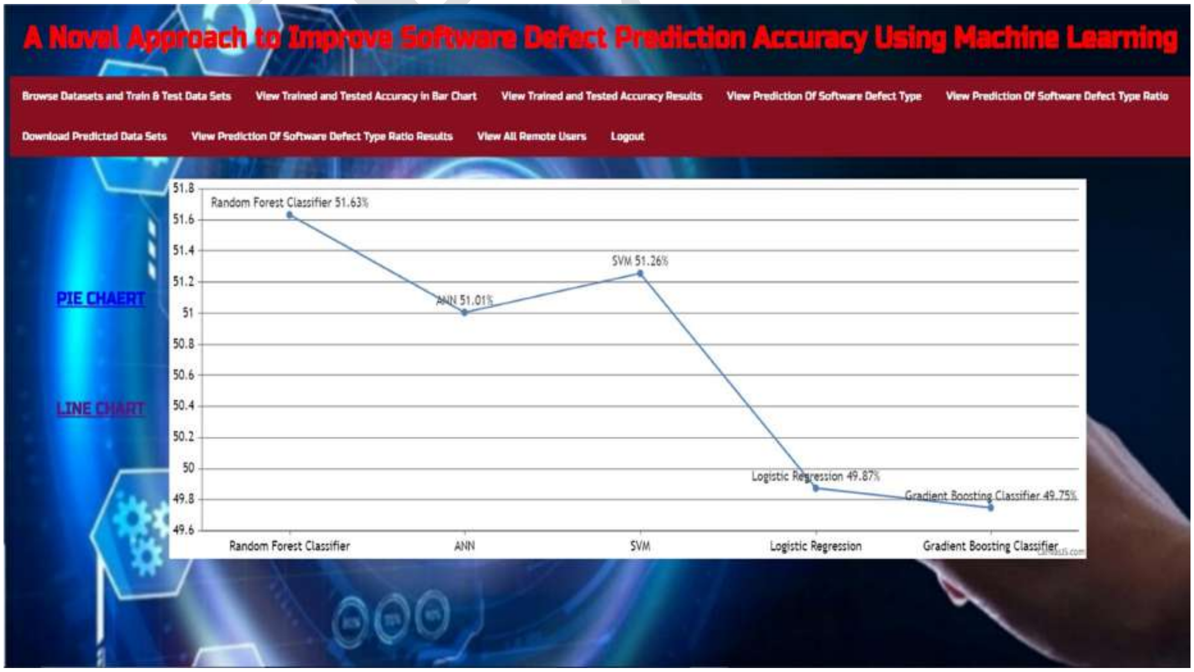
Output Testing

After performing the validation testing, the next step is output testing of the proposed system, since no system could be useful if it does not produce the required output in the specified format. Asking the users about the format required by them tests the outputs generated or displayed by the system under consideration. Hence the output format is considered in 2 ways – one is on screen and another in printed format.

TESTING STRATEGY :

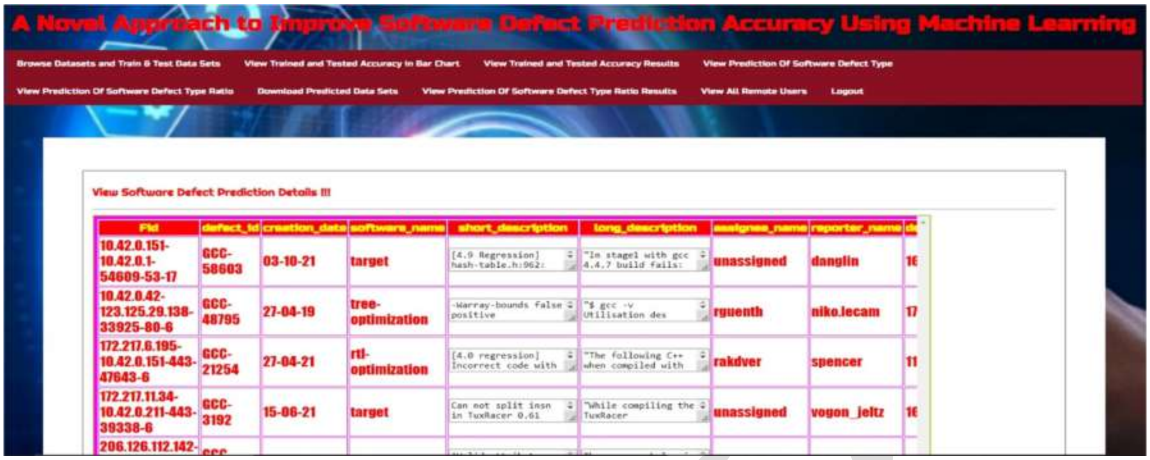
System testing strategies combine system test cases and design methodologies to build software successfully. The testing strategy must coordinate test planning, case design, execution, data collecting, and assessment.Software testing must include low-level tests to check a tiny source code segment and high-level tests to validate significant system functionalities versus user requirements. Software quality assurance relies on software testing to examine specification design and code. Testing is an intriguing software quirk. Thus, the proposed system undergoes many tests before user acceptability testing.

RESULTS



A Novel Approach to Improve Software Defect Prediction Accuracy Using Machine Learning

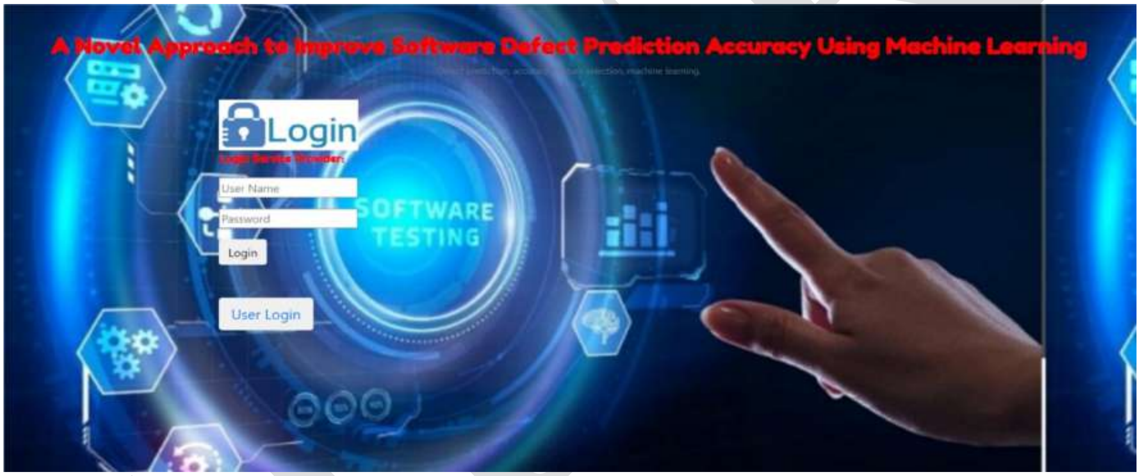
Browse Datasets and Train & Test Data Sets
View Trained and Tested Accuracy in Bar Chart
View Trained and Tested Accuracy Results
View Prediction Of Software Defect Type
View Prediction Of Software Defect Type Ratio
Download Predicted Data Sets
View Prediction Of Software Defect Type Ratio Results
View All Remote Users
Logout
PIE CHAERT
LINE CHART
51.8
51.6
51.4
51.2
51
50.8
50.6
50.4
50.2
50
49.8
49.6
Random Forest Classifier 51.63%
ANN 51.01%
SVM 51.26%
Logistic Regression 49.87%
Gradient Boosting Classifier 49.75%
Random Forest Classifier
ANN
SVM
Logistic Regression
Gradient Boosting Classifier


A Novel Approach to Improve Software Defect Prediction Accuracy Using Machine Learning

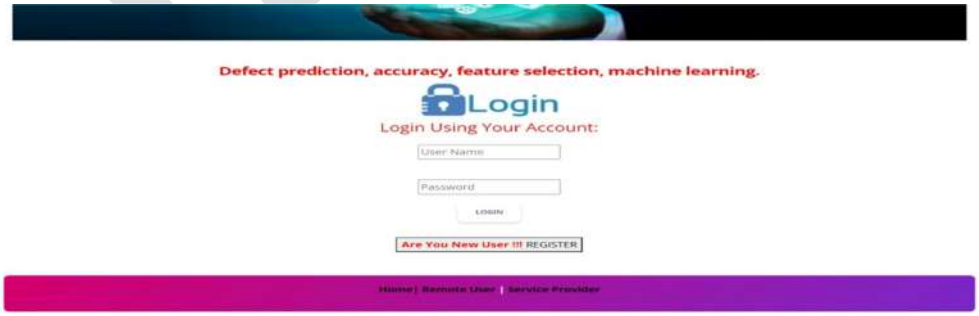
Browse Datasets and Train & Test Data Sets
View Trained and Tested Accuracy in Bar Chart
View Trained and Tested Accuracy Results
View Prediction Of Software Defect Type
View Prediction Of Software Defect Type Ratio
Download Predicted Data Sets
View Prediction Of Software Defect Type Ratio Results
View All Remote Users
Logout

View Software Defect Prediction Details !!!

Fid	defect_id	creation_date	software_name	short_description	long_description	assignee_name	reporter_name	de
10.42.0.151-10.42.0.1-54609-53-17	GCC-58603	03-10-21	target	[4.9 Regression] hash-table.h:962:	"In stage1 with gcc 4.4.7 build fails:	unassigned	danglin	16
10.42.0.42-123.125.29.138-33925-80-6	GCC-48795	27-04-19	tree-optimization	-Warray-bounds false positive	"$ gcc -v Utilisation des	rguenth	niko.lecam	17
172.217.6.195-10.42.0.151-443-47643-6	GCC-21254	27-04-21	rtl-optimization	[4.0 regression] Incorrect code with	"The following C++ when compiled with	rakdver	spencer	11
172.217.11.34-10.42.0.211-443-39338-6	GCC-3192	15-06-21	target	Can not split insn in TuxRacer 0.61	"While compiling the TuxRacer	unassigned	vogon_jeltz	16
206.126.112.142-	GCC							


A Novel Approach to Improve Software Defect Prediction Accuracy Using Machine Learning

Defect prediction, accuracy, feature selection, machine learning.


Defect prediction, accuracy, feature selection, machine learning.

Login
Login Using Your Account:
User Name
Password
LOGIN
Are You New User !!! REGISTER
Home | Remote User | Service Provider

A Novel Approach to Improve Software Defect Prediction Accuracy Using Machine Learning

Defect prediction, accuracy, feature selection, machine learning.

CONCLUSION

Software engineering research includes fault prediction. Software defect prediction anticipates source code defects before testing. Traditional defect detection approaches include box, system, and unit testing. When a project extends to test software faults, classification, clustering, mixed algorithms, data mining statistical approaches, neural networks, and machine learning are commonly used, making these tests difficult. Research methods have been planned to solve software fault prediction issues. Many software defect prediction algorithms exist, but none work for all datasets. This makes it reliant on the dataset's core data. Software prediction methods might be tough to choose. Predicting software defects involves finding source code flaws. Source code review, beta testing, black box testing, integrated testing, white size and complexity. Detecting and resolving defects becomes harder. These challenges are addressed by software defect models. Software systems are becoming more complicated in the technology age. Therefore, defects must be found. Without the right software defect detection mechanism, a product may be inferior. Software quality and dependability are most important, and defect prediction is a vital indication of both. This research analyzes five NASA data sets to enhance software fault predictions:

REFERENCES

1. [1] M. A. Memon, M.-U.-R. Magsi, M. Memon, and S. Hyder, ‘‘Defects prediction and prevention approaches for quality software development,’’ Int. J. Adv. Comput. Sci. Appl., vol. 9, no. 8, pp. 451–457, 2018.
2. [2] M. Gayathri and A. Sudha, ‘‘Software defect prediction system using multilayer perceptron neural network with data mining,’’ Int. J. Recent Technol. Eng., vol. 3, no. 2, pp. 2277–3878, 2014.

3. [3] R. Malhotra, L. Bahl, S. Sehgal, and P. Priya, ‘‘Empirical comparison of machine learning algorithms for bug prediction in open source software,’’ in Proc. Int. Conf. Big Data Anal. Comput. Intell. (ICBDAC), Andhra
4. Pradesh, India, 2017, pp. 40–45, doi: 10.1109/ICBDACI.2017.8070806.
5. [4] M. S. Rawat and S. K. Dubey, ‘‘Software defect prediction models for quality improvement: A literature study,’’ Int. J. Comput. Sci. Issues, vol. 9, pp. 288–296, Jan. 2012.
6. [5] I. Singh, ‘‘A survey? Data mining techniques in software engineering,’’ Int. J. Res. IT, Manage. Eng., vol. 6, no. 3, pp. 30–34, Mar. 2016.
7. [6] N. Kalaivani and R. Beena, ‘‘Overview of software defect prediction using machine learning algorithms,’’ Int. J. Pure Appl. Math., vol. 118, pp. 3863–3873, Feb. 2018.
8. [7] M. Dhiauddin and S. Ibrahim, ‘‘A prediction model for system testing defects using regression analysis,’’ Int. J. Soft Comput. Softw. Eng., vol. 2, no. 7, pp. 55–68, Jul. 2012.
9. [8] R. Malhotra and A. Jain, ‘‘Fault prediction using statistical and machine learning methods for improving software quality,’’ J. Inf. Process. Syst., vol. 8, no. 2, pp. 241–262, Jun. 2012.
10. [9] A. Hammouri, M. Hammad, M. Alnabhan, and F. Alsarayrah, ‘‘Software bug prediction using machine learning approach,’’ Int. J. Adv. Comput. Sci. Appl., vol. 9, no. 2, pp. 78–83, 2018.
11. Ijteba Sultana, Dr. Mohd Abdul Bari ,Dr. Sanjay,” *Routing Performance Analysis of Infrastructure-less Wireless Networks with Intermediate Bottleneck Nodes*”, International Journal of Intelligent Systems and Applications in Engineering, ISSN no: 2147-6799 IJISAE,Vol 12 issue 3, 2024, Nov 2023
12. Md. Zainlabuddin, "*Wearable sensor-based edge computing framework for cardiac arrhythmia detection and acute stroke prediction*”, Journal of Sensor, Volume2023.
13. Md. Zainlabuddin, "*Security Enhancement in Data Propagation for Wireless Network*”, Journal of Sensor, ISSN: 2237-0722 Vol. 11 No. 4 (2021).
14. Dr MD Zainlabuddin, "*CLUSTER BASED MOBILITY MANAGEMENT ALGORITHMS FOR WIRELESS MESH NETWORKS*”, Journal of Research Administration, ISSN:1539-1590 | E-ISSN:2573-7104 , Vol. 5 No. 2, (2023)
15. Vaishnavi Lakadaram, " Content Management of Website Using Full Stack Technologies”, Industrial Engineering Journal, ISSN: 0970-2555 Volume 15 Issue 11 October 2022
16. Dr. Mohammed Abdul Bari,Arul Raj Natraj Rajgopal, Dr.P. Swetha ,” *Analysing AWSDevOps CI/CD Serverless Pipeline Lambda Function's Throughput in Relation to Other Solution*”, International Journal of Intelligent Systems and Applications in Engineering , JISAE, ISSN:2147-6799, Nov 2023, 12(4s), 519–526
17. Ijteba Sultana, Mohd Abdul Bari and Sanjay,” *Impact of Intermediate per Nodes on the QoS Provision in Wireless Infrastructure less Networks*”, Journal of Physics: Conference Series, Conf. Ser. 1998 012029 , CONSILIO Aug 2021
18. M.A.Bari, Sunjay Kalkal, Shahanawaj Ahamad," *A Comparative Study and Performance Analysis of Routing Algorithms*”, in 3rd International Conference ICCIDM, Springer - 978- 981-10-3874-7_3 Dec (2016)

19. Mohammed Rahmat Ali,: BIOMETRIC: AN e-AUTHENTICATION SYSTEM TRENDS AND FUTURE APLLICATION", International Journal of Scientific Research in Engineering (IJSRE), Volume1, Issue 7, July 2017
20. Mohammed Rahmat Ali,: BYOD.... A systematic approach for analyzing and visualizing the type of data and information breaches with cyber security", NEUROQUANTOLOGY, Volume20, Issue 15, November 2022
21. Mohammed Rahmat Ali, Computer Forensics -An Introduction of New Face to the Digital World, International Journal on Recent and Innovation Trends in Computing and Communication, ISSN: 2321-8169-453 – 456, Volume: 5 Issue: 7
22. Mohammed Rahmat Ali, Digital Forensics and Artificial Intelligence ...A Study, International Journal of Innovative Science and Research Technology, ISSN:2456-2165, Volume: 5 Issue:12.
23. Mohammed Rahmat Ali, Usage of Technology in Small and Medium Scale Business, International Journal of Advanced Research in Science & Technology (IJARST), ISSN:2581-9429, Volume: 7 Issue:1, July 2020.
24. Mohammed Rahmat Ali, Internet of Things (IOT) Basics - An Introduction to the New Digital World, International Journal on Recent and Innovation Trends in Computing and Communication, ISSN: 2321-8169-32-36, Volume: 5 Issue: 10
25. Mohammed Rahmat Ali, Internet of things (IOT) and information retrieval: an introduction, International Journal of Engineering and Innovative Technology (IJEIT), ISSN: 2277-3754, Volume: 7 Issue: 4, October 2017.
26. Mohammed Rahmat Ali, How Internet of Things (IOT) Will Affect the Future - A Study, International Journal on Future Revolution in Computer Science & Communication Engineering, ISSN: 2454-424874 – 77, Volume: 3 Issue: 10, October 2017.
27. Mohammed Rahmat Ali, ECO Friendly Advancements in computer Science Engineering and Technology, International Journal on Scientific Research in Engineering(IJSRE), Volume: 1 Issue: 1, January 2017
28. Ijteba Sultana, Dr. Mohd Abdul Bari ,Dr. Sanjay, "*Routing Quality of Service for Multipath Manets, International Journal of Intelligent Systems and Applications in Engineering"*, JISAE, ISSN:2147-6799, 2024, 12(5s), 08–16;
29. Mr. Pathan Ahmed Khan, Dr. M.A Bari,: Impact Of Emergence With Robotics At Educational Institution And Emerging Challenges", International Journal of Multidisciplinary Engineering in Current Research(IJMEC), ISSN: 2456-4265, Volume 6, Issue 12, December 2021,Page 43-46
30. Shahanawaj Ahamad, Mohammed Abdul Bari, Big Data Processing Model for Smart City Design: A Systematic Review ", VOL 2021: ISSUE 08 IS SN : 0011-9342 ;Design Engineering (Toronto) Elsevier SCI Oct : 021
31. Syed Shehriyar Ali, Mohammed Sarfaraz Shaikh, Syed Safi Uddin, Dr. Mohammed Abdul Bari, "Saas Product Comparison and Reviews Using Nlp", Journal of Engineering Science (JES), ISSN NO:0377-9254, Vol 13, Issue 05, MAY/2022

32. Mohammed Abdul Bari, Shahanawaj Ahamad, Mohammed Rahmat Ali," Smartphone Security and Protection Practices", International Journal of Engineering and Applied Computer Science (IJEACS) ; ISBN: 9798799755577 Volume: 03, Issue: 01, December 2021 (International Journal,U K) Pages 1-6
33. .A.Bari& Shahanawaj Ahamad, "Managing Knowledge in Development of Agile Software", in International Journal of Advanced Computer Science & Applications (IJACSA), ISSN: 2156-5570, Vol: 2, No: 4, pp: 72-76, New York, U.S.A., April 2011
34. Imreena Ali (Ph.D), Naila Fathima, Prof. P.V.Sudha ,"Deep Learning for Large-Scale Traffic-Sign Detection and Recognition", Journal of Chemical Health Risks, ISSN:2251-6727/ JCHR (2023) 13(3), 1238-1253
35. Imreena, Mohammed Ahmed Hussain, Mohammed Waseem Akram" An Automatic Advisor for Refactoring Software Clones Based on Machine Learning", Mathematical Statistician and Engineering ApplicationsVol. 72 No. 1 (2023)
36. Mrs Imreena Ali Rubeena,Qudsiya Fatima Fatimunisa "Pay as You Decrypt Using FEPOD Scheme and Blockchain", Mathematical Statistician and Engineering Applications: https://doi.org/10.17762/msea.v72i1.2369 Vol. 72 No. 1 (2023)
37. Imreena Ali , Vishnuvardhan, B.Sudhakar," Proficient Caching Intended For Virtual Machines In Cloud Computing", International Journal Of Reviews On Recent Electronics And Computer Science , ISSN 2321-5461,IJRRECS/October 2013/Volume-1/Issue-6/1481-1486
38. Heena Yasmin, A Systematic Approach for Authentic and Integrity of Dissemination Data in Networks by Using Secure DiDrip, INTERNATIONAL JOURNAL OF PROFESSIONAL ENGINEERING STUDIES, Volume VI /Issue 5 / SEP 2016
39. Heena Yasmin, Cyber-Attack Detection in a Network, Mathematical Statistician and Engineering Applications, ISSN:2094-0343, Vol.72 No.1(2023)
40. Heena Yasmin, Emerging Continuous Integration Continuous Delivery (CI/CD) For Small Teams, Mathematical Statistician and Engineering Applications, ISSN:2094-0343, Vol.72 No.1(2023)