10.48047/jocaaa.2024.33.05.65

DESIGN AND IMPLEMENTATION OF DATA MINING TECHNIQUES TO DEVELOP SOFTWARE RELIABILITY

DR.Ch.Kishore Kumar

Assistant Professor Dept of Computer Science
VAAGDEVI COLLEGE OF ENGINEERING, INDIA.
E-MAIL: dr.kishorekumarcs@vaagdevicolleges.com

Received: 05.03.2024 **Revised: 15.04.2024** **Accepted: 31.05.2024**

Abstract

The Reliability of any software is becoming so important in our daily life need; Errors of the software unfortunately continue to be frequent to cause system failures. In software development the most time consuming and difficult task is to detect bugs and fix them. Then for solving this difficult issue it would be highly recommended if we analyze the bud and understand their behavior and their characteristics then detect them automatically. In the software that contains a huge code of data and in documents it is difficult for the developers to analyze the data and detect hem. We are approaching a data mining technique to extract a useful knowledge in the large software and contribute this knowledge for bug detecting.

1. INTRODUCTION

In modern society, computers are used for many different applications, such as nuclear reactors, aircraft, banking systems, and hospital patient monitoring systems. As the demand of the application quality becomes higher and higher, the research ofthe 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.

1.1 Software Reliability

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 [8]. 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

10.48047/jocaaa.2024.33.05.65

the likelihood that considered software exhibits its expected functions without causing any malfunctioning in defined time duration.

The prestigious organization, Institute of Electrical and Electronic Engineers (IEEE) establishes the software reliability in terms of the likelihood that the application or the software would not have to suffer any kind of system failure for certain specific time under certain very critical functional working circumstances or conditions. The likelihood is nothing else but the function of data inputs and the application of the considered system as well as a function of the presence of any intrusion or malicious entity in software. The data inputs to the developed system estimate ifthe system suffers any intrusion in function

1.2 Importance of Software Reliability

In general there exist minimum four predominant reasons due to which the software reliability has come out in the form of very significant concerns in last few years.

• Now days, majority of functional systems are being framed towards software applications. Majority of flight mechanisms are resulting into highly software centric as compared to hardware based systems. The other systems comprising of entities as teller, automated teller and other allied processing functions are software intensive.

• Software Reliable intensive requirements of customers. A number of organizations are now looking for certain secured and reliable software applications. Software systems have become significant component ofreliability allocations on varied organizations and its associated contracts.

Problems encountered by delivered software include;

1. Errors detected after delivery

2. Requirements not translated correctly

3. Product not complete

4. Difficulty in supporting software

5. Apparent lack ofa systematic process

1.3 Mining Information from Large Software

The huge amount of analysis data in large software such as source code and documents, however, renders a tedious and difficult task on developers to analyze them. In order to extract information from such large software and detect intrusions, this dissertation proposes to apply data mining techniques for analysis ofsource code and documents. This approach is the first one to apply data mining techniques in intrusion detection. Data mining, also known as knowledge

10.48047/jocaaa.2024.33.05.65

discovery and data mining (KDD), has been developed at a rapid pace in recent years due to the wide availability of voluminous data and the imminent need for extracting useful information and knowledge from them.

2. LITERATURE SURVEY

Yi Liu et al., 2010 [1] examine the trouble of software authority classification modeling utilizing the previous of metric dataset has been acquired from particular software development. The classification modeling obtain from a distinct dataset is usually not sufficient to construct strong as well as a precise model. To attempt the problem, software excellence classification modeling has been completed by use of several datasets sourced as of dissimilar software development. Earlier survey has been established on use of manifold datasets for confirmation can accomplish robust inherited programming based representation.

Song et al., 2006 [2] suggested prediction of imperfection associations with defect rectification attempt based on involvement rule mining methods. Analysis property is further efficiently owed for detecting software imperfection. The projected technique has been being relevant to additional than two hundred scheme.

Lessmann et al., 2008 [3] examined the presentation of organization algorithm. To evaluate the software deficiency prediction, experimentation is present utilizing then public province datasets from NASA Metric Data repository, employing 22 classifiers. The common impression is that the predictive correctness metric based categorization has been helpful. The consequences output also designate that the significance emotionally involved to exacting classification algorithms is not important as usually unspecified. The results demonstrated that there is no major differentiation between the best 17 classifiers.

Munson et al., 1992 [4] examined arithmetical based techniques like discriminating investigation for the recognition of responsibility prone programs. During this dissertation it has been projected to put into practice principal apparatus to remove multi-collinear difficulty metrics to uncorrelated procedures on orthogonal difficulty domains.

Tom M. Mitchell 1997, [33] has been effort to categorize data utilizing Naive Bayesian algorithm. Naive Bayes is individual of the prevalently used instruction algorithms inside data mining as well as machine learning. It is admired and the reason is its well-organized and effectual inductive learning algorithms. Cataloging based scheduled NaTve Bayes algorithms provide very aggressive presentation due to its conditional independence hypothesis.

Ohlsson et al., [5] proposed consequential metrics using design articles of telecommunication software component to expect responsibility prone modules previous to testing with precise results.

10.48047/jocaaa.2024.33.05.65

Menzies et al., 2004 [6] suggest Naive Bayes apprentice for learning the imperfection detectors from static code measures. Evaluation of Naive Bayes learners as well as entropy-based conclusion tree beginner has been completed to demonstrate the efficiency of Naive Bayes learners.

Riqueitne et al., 2009 [7] employed the assure depository to get hold of the software metrics program dataset as well as projected a inherited algorithm explore for rules exemplify subgroups with a elevated possibility of being imperfect. The genetic algorithm holds the difficulty of disturbed datasets competently particularly when the unbalanced sets consist of additional nondefective example than the defective samples.

3. EXISTING SYSTEM

3.1 A Neural-Network-Based Approach for Software Reliability Modeling

In modern society, computers are used for many different applications, such as nuclear reactors, aircraft, banking systems, and hospital patient monitoring systems. As the demand of the application quality becomes higher and higher, the research ofthe computer software reliability becomes more and more essential. The software reliability is defined as the probability that the software will operate without a failure under a given environmental condition during a specified period of time [107]. To date, the software reliability assessment is one of the most important processes during the software development. Since 1970, many software reliability growth models (SRGMs) [108][109] have been proposed. In general, there are two major types of software reliability models: the deterministic and the probabilistic. The deterministic one is employed to study the number of distinct operators and operands in the program. The probabilistic one represents the failure occurrences and the fault removals as probabilistic events.

As discussed earlier in the last sections, the significant objective function of the NN can be potentially taken into consideration as its compound functions. In fact, in case it doesn't derive any form of composite functions from the conservative reliability models of the software, it conserves for constructing a NN based schematic for ensuring software reliability.

4. PROPOSED METHODOLOGY

4.1 Evolutionary Support Vector Machines

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. 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 (Vapnik, 1960) for exhibiting data

10.48047/jocaaa.2024.33.05.65

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.

4.2 Intrusion Detection using Proposed ESVM Mining Module

Normal and attack traffic are classified by the ESVM, During training ESVM learns the normal and attack patterns from the training file. In testing ESVM differentiate the attack and normal traffic using learned patterns.

4.2.1 Training ofESVM

In this research work a highly robust Evolutionary Support Vector Machine (ESVM) mechanism has been employed for training ofmining module. This training approach classifies a given data point x € Rn by assigning a label y € {Normal, ICMP, TCP, UDP, Smurf, Port scan, Land, HTTP, Session, IP}. Where x is the set of inputs to the Support Vector Machine, Rn is relation between n attributes, y is the output produced by the EMCSVM which consist often different types ofclasses including one normal class and nine attack classes. Training file for Evolutionary Support Vector Machine (ESVM) is created from Training data set. Training data set file contains 20 attributes and 10 types of classes. ESVM trained using the training data set and weight values then produce the model file; this model file Is used to classify the different types of attacks.

S.No	Kernel Name	Number of iteration	nSV	nBSV	Total nSV
	Linear	100	89	39	89
	Polynomial	170	107	45	107
	Radial basis	90	95	33	95

5. ALGORITHM FOR PROPOSED SYSTEM

5.1 Learning algorithm

The algorithm describes a process to adjust the weights. During the learning processes, the weights of network are adjusted to reduce the errors of the network outputs as compared to the standard answers. The back-propagation algorithm is the most widely employed one. In back propagation algorithm, the weights of the network are iteratively trained with the errors propagated back from the output layer.

10.48047/jocaaa.2024.33.05.65

Subsequently, as it is describe the learning algorithm of the neural network in a mathematical form. The objective of the neural networks is to approximate a non-linear function that can receive the vector x1,……Xn in RN Thus the network can be denoted as;

$$y = F(x),$$

Where $x = (x_1, \cdots, x_n)$ and $y = (y_1, \cdots, y_m)$. The values of y_k are given by

$$y_k = g\left(b_k + \sum_{j=1}^{H} w_{jk}^{0} h_j\right), k = 1, M$$

5.2 The Derivations of Proposed Approach in Software Reliability Modeling

In inceptional case the logistic growth curve model as discussed in [120] is taken into consideration. This system model would efficiently adapt itself for the average function in the form ofthe logistic function, as depicted in the following expression:

$$m(t) = \frac{a}{1 + ke^{-bt}}, a > 0, b > 0, k > 0$$

Substitute k with e^{-c}:

$$m(t) = \frac{a}{1 + ke^{-bt}}, \frac{a}{1 + e^{-c}e^{-bt}} = \frac{a}{1 + e^{-(ct+c)}}$$

Cnsider that

$$g(x) = bx + c$$

$$f(x) = \frac{1}{1 + e^{-x}}$$

And

$$m(x) = ax$$

Hence, it would come out as

$$m\left(f(g(x))\right) = m(f(bx + c))$$

$$= m\left(\frac{1}{1 + e^{-(bx+c)}}\right) = \left(\frac{1}{1 + e^{-(bx+c)}}\right)$$

10.48047/jocaaa.2024.33.05.65

5.3 An Implementation Of Proposed Approach

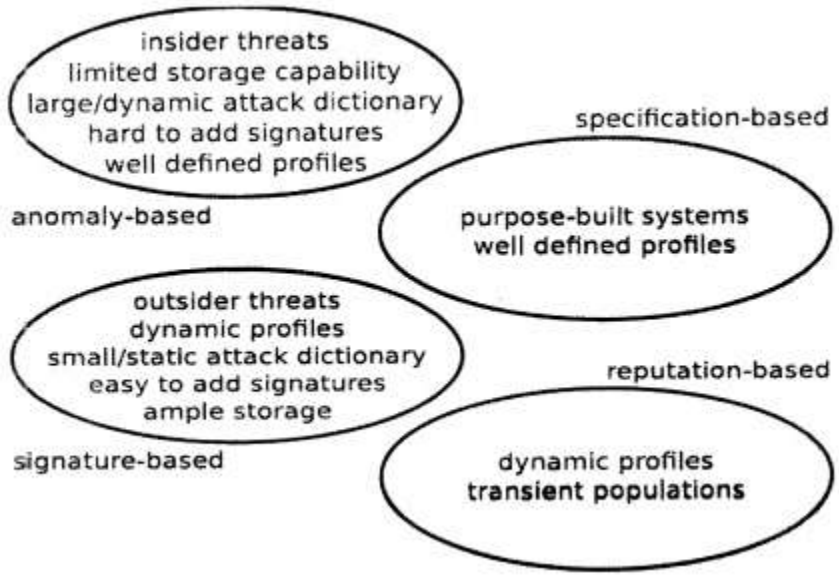
As mentioned earlier that has been mentioned and shown the proposed approach is achievable. In this section, the implementation of the proposed neural-network-based models will be described. The following steps are required to apply approach for software reliability prediction.

1. 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.
2. Construct the neural network of selected models by designing the activation functions and bias

6. TOOLS AND TECHNIQUES

6.1Detection Technique

In this section first described existing IDS detection techniques including signature-based, anomaly-based, specification-based and reputation-based techniques. Subsequently discussed, the effectiveness of IDS detection techniques which is applying to CPSs. Below figure shows the Detection Technique dimension.



6.2 Data Mining Tools

- Traditional Data Mining Tools
- Dashboards
- Text-mining Tools
- WEKA TOOL

10.48047/jocaaa.2024.33.05.65

- DBMINER TOOL

7. RESULTS AND ANALYSIS

7.1 Background

This research has been developed and carried out with a goal to provide software reliabiity using data mining approach and intrusion detection system. In this thesis Evolutionay Support Vector Machine (ESVM) based system model has been developed for data mining and intrusion detection scheme has been developed. In this research the attacks of DDoS was experimented over available data sets so as to ensure effectiveness of the proposed system model. In order to exhibit the robusteness of the proposed system, a parallel system based on neural network (NN) for software reliability essurance.

The neural network (NN) approach has been employed for data mining and intrusion detection that ensures reliability of certain software application to be employed with huge data sets. The overall system has been developed with C# programming language and visual studio 2010 platform. In order to examine the system effieincy and robustness in terms of time and accuracy, various datatsets have been taken into consideration. In the following figure (Figure 1) the intrusion detecton time for 100 data packets using neural network based data mining have been presented. Similarly, in figure (Figure 2) the detection time for ESVM based data mining has been presented. Considering the intrusion detection analysis (Table 3) it can be stated that the proposed Evolutionary Support Vector Machine (ESVM) based system is approximately 83.2% more faster as compared to concventional neural network (NN) based system.

10.48047/jocaaa.2024.33.05.65

Predictiv positive	Predictiv negative	True Positive	False Positive	Sensit-ivity	Specif-icity	Efficie-ncey	Accurr-acy	False Positive	False discovery	Mathew C.C
100	0	46	54	1	0	0.5	0.46	1	0.54	0
97	3	46	51	1	0.0555	0.5277	0.49	0.9444	0.5257	0.1623
94	6	46	48	1	0.111	0.5555	0.52	0.8888	0.5106	0.233
90	10	46	44	1	0.1851	0.5925	0.56	0.8148	0.4888	0.3076
87	13	46	41	1	0.240	0.6203	0.59	0.7592	0.4712	0.3567
84	16	46	38	1	0.296	0.6481	0.62	0.7037	0.4523	0.4028
76	24	46	30	1	0.444	0.7222	0.7	0.5555	0.3947	0.5186
76	24	46	30	1	0.4444	0.7222	0.7	0.5555	0.3947	0.5186
69	31	46	23	1	0.5740	0.7870	0.77	0.4259	0.3333	0.6186
65	35	46	19	1	0.6481	0.8240	0.81	0.3518	0.2923	0.6772
63	37	46	17	1	0.6851	0.8425	0.83	0.3148	0.2698	0.707
60	40	46	14	1	0.7407	0.8703	0.86	0.2592	0.2333	0.753
57	43	46	11	1	0.7962	0.8981	0.89	0.2037	0.1929	0.8016
51	49	46	5	1	0.9074	0.9537	0.95	0.0925	0.0980	0.904
48	52	46	2	1	0.9629	0.9814	0.98	0.0370	0.0416	0.9606
46	54	46	0	1	1	1	1	0	0	1
46	54	46	0	1	1	1	1	0	0	1
46	54	46	0	1	1	1	1	0	0	1
46	54	46	0	1	1	1	1	0	0	1
46	54	46	0	1	1	1	1	0	0	1
35	65	35	0	0.760	1	0.8804	0.89	0	0	0.795
33	67	33	0	0.717	1	0.8586	0.87	0	0	0.760
36	64	36	0	0.782	1	0.8913	0.9	0	0	0.8126
46	54	46	0	1	1	1	1	0	0	1
38	62	38	0	0.826	1	0.9130	0.92	0	0	0.848

Table 1 ROC Using Neural Network

10.48047/jocaaa.2024.33.05.65

Predictiv positive	Predictiv negative	True Positive	False Positive	Sensit-ivity	Specif-icity	Efficie-ncey	Accurr-acy	False Positive	False discovery	Mathew C.C
100	0	50	50	1	0	0.5	0.5	1	0.5	0
0	100	0	0	0	1	0.5	0.5	0	1	0
7	93	7	0	0.14	1	0.57	0.57	0	0	0.274
91	9	50	41	1	0.18	0.59	0.59	0.82	0.451	0.3148
13	87	13	0	0.26	1	0.63	0.63	0	0	0.386
81	19	50	31	1	0.38	0.69	0.69	0.62	0.3827	0.4843
19	81	19	0	0.38	1	0.69	0.69	0	0	0.4843
22	78	22	0	0.44	1	0.72	0.72	0	0	0.5310
25	75	25	0	0.5	1	0.75	0.75	0	0	0.5773
29	71	29	0	0.58	1	0.79	0.79	0	0	0.639
70	30	50	20	1	0.6	0.8	0.8	0.4	0.285	0.6546
33	67	33	0	0.66	1	0.83	0.83	0	0	0.7018
36	64	36	0	0.72	1	0.86	0.86	0	0	0.75
59	41	50	9	1	0.82	0.91	0.91	0.18	0.1525	0.8336
41	59	41	0	0.82	1	0.91	0.91	0	0	0.8336
50	50	50	0	1	1	1	1	0	0	1
50	50	50	0	1	1	1	1	0	0	1
50	50	50	0	1	1	1	1	0	0	1
50	50	50	0	1	1	1	1	0	0	1
50	50	50	0	1	1	1	1	0	0	1
50	50	50	0	1	1	1	1	0	0	1
50	50	50	0	1	1	1	1	0	0	1

Table 2 ROC Using ESVM

❖ Enhanced Support vector machine experimental results In the ESVM technique the Experimental results we got using ready data is given below

100 records results using Enhanced Support Vector Machine(SVM)	
TP=47	FP=1
FN=3	TN=49

Table 3 ESVM dataset

Roc calculation using ESVM of 100 LIVE records	
Sensitivity= 0.9418	Specificity= 0.9706
Efficiency= 0.9562	Accuracy= 0.9562

Table 4 ROC ESVM dataset

10.48047/jocaaa.2024.33.05.65

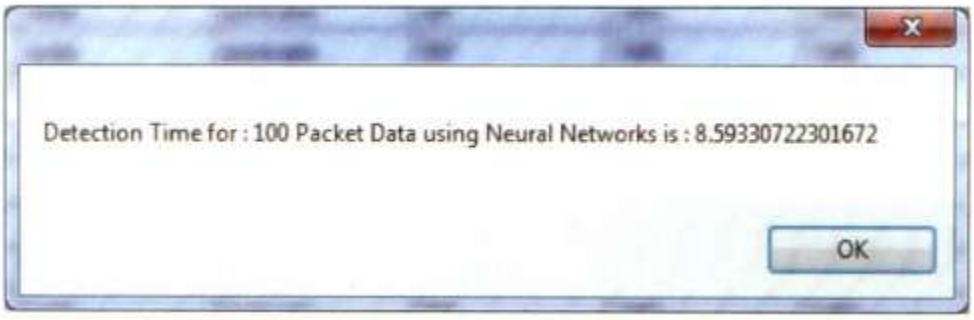


Figure: 1 Intrusion detection time for proposed NN based mining approach

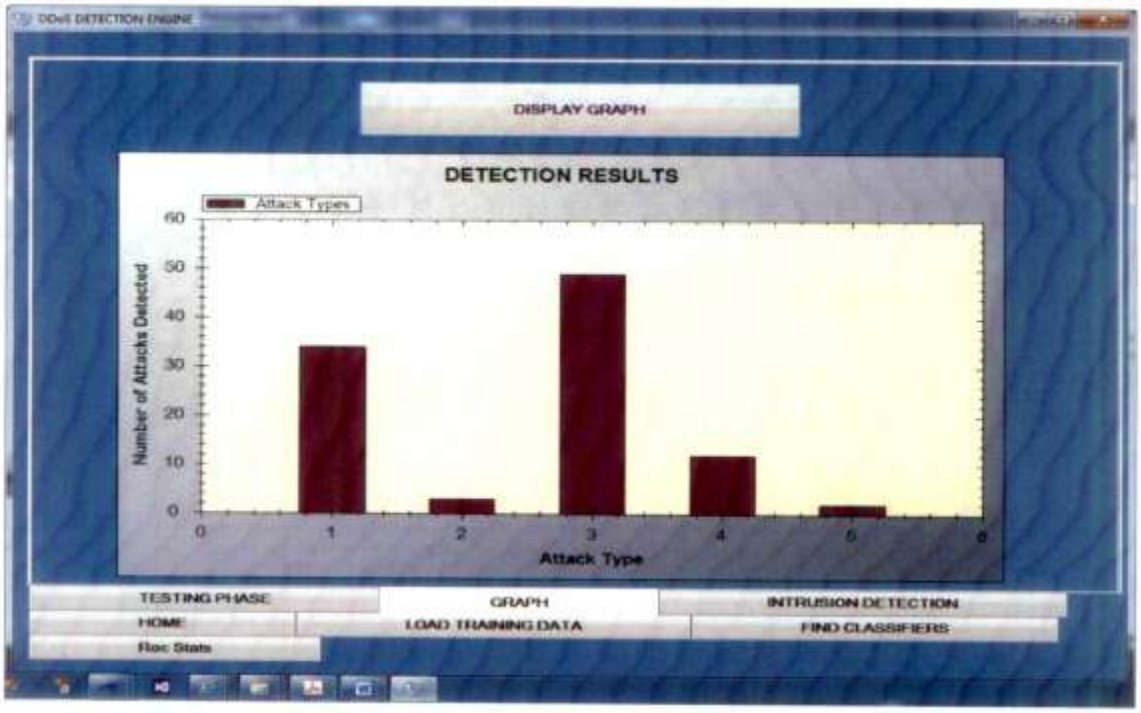


Figure 2 Analysis for number of attacks detected using NN mining model

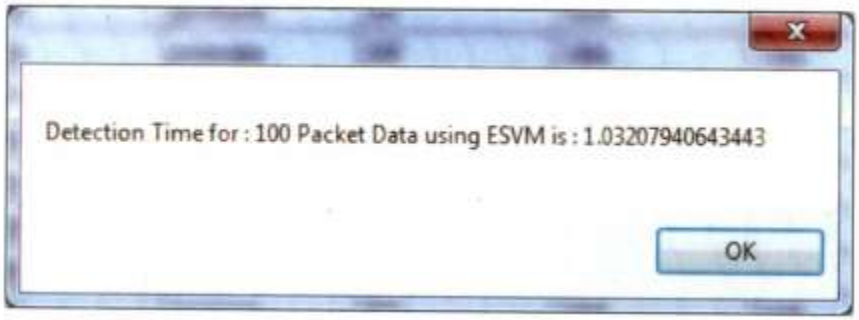


Figure 3 Intrusion detection time for proposed ESVM based mining approach

10.48047/jocaaa.2024.33.05.65

As mentioned in Figure 1, the efficiency of the proposed ESVM based system is better as compared to NN paradigm. The better accuracy detection using machine learning and evolutionry approahc makes ESVM robust for behaving linearly.

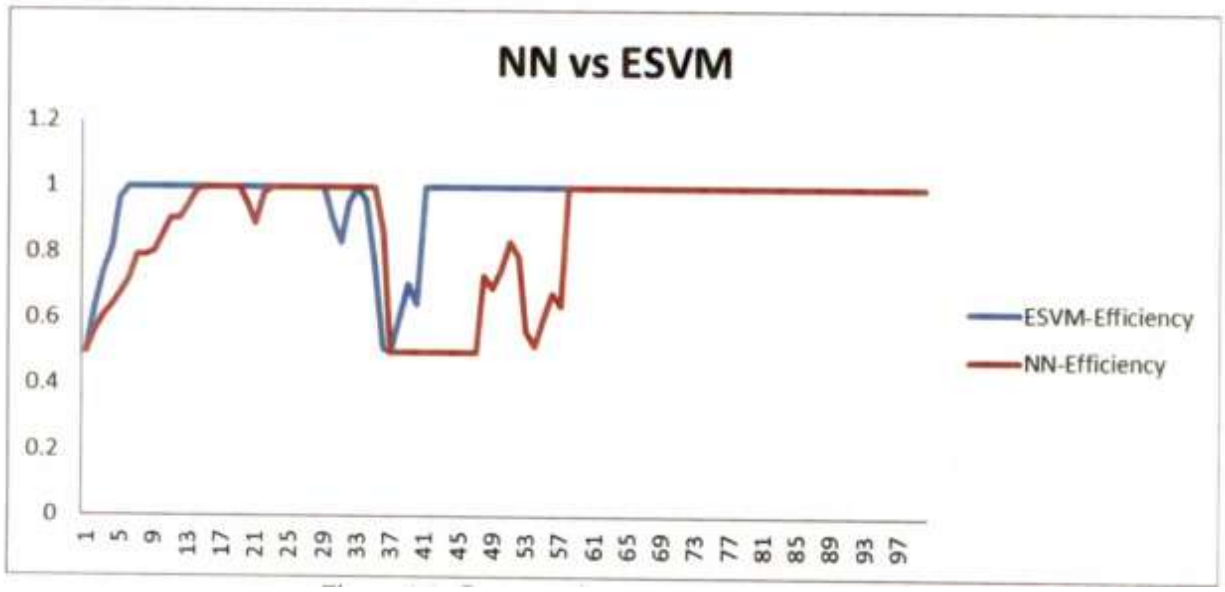


Figure4 Comparative analysis for efficiency

8. CONCLUSION

Software reliability is one ofthe 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

10.48047/jocaaa.2024.33.05.65

generalization capability. This optimization has made the system much robust for swift operation and higher accurate performance even with varying datasets. In order to examine the system efficiency for various attacks scenario, here in this research work' the distributed denial of service (DDoS), reply attacks and many more kinds of attacks have been performed so as to ensure reliable function ofthe proposed system to ensure software reliability. The proposed system has exhibited better results for intrusion detection while exhibiting data mining on huge datasets available on software networks. The proposed ESVM scheme has exhibited better efficiency in terms of faster intrusion detection and sensitivity. This development and system implementation strengthens the reliability ofsoftware. Thus the developed system has presented a novel scheme for software reliability assurance.

9. FUTURE SCOPE

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. 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 NaeemSeliya: 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] Mitchell, T. (1997), Machine Learning, McGraw-Hill Ltd.

[6] 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

10.48047/jocaaa.2024.33.05.65

[7] Tim Menzies, Justin DiStefano, Andres OrregoS, Robert (Mike) Chapman: Assessing Predictors of Software Defects, Workshop on Predictive Software Models www.site.uottawa.ca/psm2004.http://menzies.us/pdf/04psm.pdf.

[8] JC Riquelme, Rodriguez, D, R. Ruiz, JS Aguilar-Ruiz: Searching for Rules to find Defective Modules in Unbalanced Data Sets, SSBSE '09, 1st International Symposium on Search Based Software Engineering, 2009, pp. 89-92.

[9] Cagatay Catal, Banu Diri; Software Defect Prediction using Artificial Immune Recognition System, Proceeding of the 25th IASTED International Multi-Conference, Feb 13-15, 2007. pp. 285-290

[10] Dennis J, Drown, Taghi M. Khoshgoftaarand NaeemSeliya: Evolutionary Sampling and Software Quality Modeling of High-Assurance Systems, IEEE Transactions on Systems, Man, and Cybernetics—Part A: Systems And Humans, Vol. 39, No. 5, September 2009, pp. 1097-1107

[11] Fenton. N.E. and M. Neil: A Critique of Software Defect Prediction Models, IEEE Transactions on Software Engineering, Vol. 25, No. 5, Sept./Oct. 1999, pp. 675-689.

[12] Jorge A.Achcar, DipakK.Dey and MuraliNiverthi, Saocarlos, s.p., (1991), 'A Bayesian Approach using Nonhomogeneous Poission Process for Software Reliability Models', Brazi