IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 8, Issue 3, March 2019

Signature Based Intrusion Detection Systems by Using Random Forest Algorithm

Jhalak Jain[1], Himanshu Yadav[2], Chetan Agrawal[3]

Department of Computer Science & Engineering, RITS, Bhopal[1,2,3]

Abstract: To improve network security different steps has been taken as size and importance of the network has increases day by day. At that point opportunity of a system attacks increases. Network is for the most part assaulted by intrusions that are distinguished by system intrusion recognition framework. This paper attempts to build up an intrusion location framework which uses the signature and identity of the intrusion for distinguishing various types of intrusions. Here random forest tree algorithm was used for finding the patterns in the input data. In this work use of Gini index was done for the decision tree construction in recursive manner. Experiment was done on NSL-KDD dataset which was real. Comparison was done with latest RNN (Recurrent Neural Network). Result obtained after analyzing this system improved precision value by 12.06%, while recall value by 1.15% and accuracy values were improved by 6.87%.

Keywords*:* Clustering, Gini-Index, Intrusion Detection, Random Forest, Pattern generation.

I. INTRODUCTION

Setting up network security to different advantages on the web, network structures, communication organizations done by implying some steps like encryption, firewall, virtual private framework, etc. This help for Intrusion disclosure system which is an imperative development among those. Interruption disclosure field ascends out of latest couple of years and developed a lot which utilizes the assembled information from different kind of interruption attacks and dependent on those various business, open source programming things seem to verify your framework to improve security of the particular correspondence, advantage giving frameworks. As the amount of framework customers and machine are extending well ordered to give unmistakable kind of organizations and ease for the smoothness of the world. In any case, some unapproved customers or activities from various types of aggressors which may be internal user or external interlopers in order to interrupt the running system, which are known as software engineers or programmers, show up. The reason for such point of view software engineer is to chop down cumbersome frameworks and web organizations. As a result of augmentation in interest of system security of various sorts of strikes, various researchers have incorporated their excitement for this field and wide arrangement of calculations and conventions has been created by them, in order to give secure organizations to the end customers. Among various kind of strike interruptions is a sort of assault that develop a business interest. Interruption revelation structure is displayed for the security from interruption attacks.

From the above talk this work can finish the guideline purpose of the Intrusion detection system is to distinguish all conceivable intrusion which perform malevolent movement, PC attack, spread of infections, PC abuse, and so on so a system intrusion discovery framework examinations diverse information parcels as well as screen them that movement over the web for such sort of vindictive action. Thusly, the smooth running of all things considered framework different server needs to settle all things considered framework which go about as framework interruption area system that screen all of the groups improvements and perceive their direct with the toxic exercises. One progressively sorts of framework Intrusion disclosure structure is made that can be presented in a joined server which also work in the near form of examining and watching the unmistakable bundle data units for their framework interruption lead. Framework Intrusion acknowledgment structure can be made by two unmistakable techniques which can be named as signature based and abnormality based. In the event that there ought to be an event of imprint based Network Intrusion disclosure structure it develops an aggregation of security chance mark. So as shown by the profile of each hazard the data stream of different packages in the framework are perceived and the most organizing profile is distributed to that particular groups. If the profile is malevolent, that data pack goes under interruption and it needs to remove from the framework in order to stop his out of line works out.

II. RELATED WORK

Chuanlong Yin [1] "A Deep Learning Approach for Intrusion Detection Using Recurrent Neural Networks" this paper, examines how to show an interruption acknowledgment structure subject to significant learning, and propose a

significant learning approach for interruption discovery utilizing Intermittent Neural Systems (RNN-IDS). In addition, this paper has considered the execution of the model in twofold arrangement and multiclass characterization, and the quantity of neurons and distinctive learning rate impacts on the execution of the proposed model.

A.R. Jakhale, et. al [2] In this work the creator depicts an anomaly disclosure system and its two phases especially preparing and testing. The slipping window and grouping is familiar with nursing the system development by mining the monotonous precedents using counts. The estimations are so real and used as a piece of steady watching. The typical multi-structure getting count has high area rate. At last, increase the distinctive verification rate and diminished the bogus caution rate.

Research by Jiefei, Lobo and Russo [3] explores the event of Multi-way-controlled ambush where a strike is parceled and sent over different courses to endeavor to trap an IDS structure. This is influenced conceivable due to multi way TCP (MPTCP) which engages transmissions to course finished different courses between a source and target.

Barolli et al [4] investigates the usage of IDS using neural system for giving IDS course of action in a Tor (The Onion Router) compose. Tests utilized a Tor server and client with back inducing NN to repeat trades over the Tor sort out while getting for examination. The framework proposed is a prepared ANN with information caught from Wireshark, at that point the server and customer information are analyzed, contrasts will recognize an interruption or misuse. The outcomes from testing were fruitful in giving viable exactness when assessed in the test condition.

ChuanLong [5] In this paper, author investigate how to display an interruption recognition framework in light of profound learning, and this work propose a profound learning approach for intrusion identification utilizing recurrent neural networks (RNN-IDS). Additionally, this work examines the execution of the model in paired classification and multiclass classification, and the quantity of neurons and distinctive learning rate impacts on the execution of the proposed display. This work contrasts it and those of J48, artificial neural network, arbitrary woodland, bolster vector machine, and other machine learning strategies proposed by past analysts on the benchmark data index.

Yogitha et. al. [6] Offered interruption discovery framework with Support Vector Machine (SVM). Affirmation is finished by coordinating explores on NSL-KDD Cup'99 data collection which is reformer type of KDD Cup'99 data index. By utilizing this NSLKDD Cup'99 data collection they have condensed wide time obligatory to shape SVM exemplary by achievement proper pre-training on data collection. In this association SVM made clustering of data. By obligation appropriate part accumulation assault location rate is opened up and false positive rate (FPT) is lessened. In this proposed work author has utilized Gaussian Circular Basis.

Grunt [15] item gives high adaptability that permit to the client to self-design and alter its source code by utilizing source fire. The real downside of Snort is that it utilizes just mark based procedure to recognize the intrusion however on the off chance that abnormality lead happen, SNORT won't be able to distinguish that inconsistency assault [5].

This paper [16] gives a method of secure versatile operator in IDPS for the security of framework. Secure portable operator screens the framework, forms the logs, distinguishes the assaults, and ensures the host via robotized constant reaction. Real hindrance is that on the off chance that the objective of the assailants is portable specialist, it will be hard to shield the framework from being hacked. Thus, it needs to receive some security foundations for the assurance of versatile operator.

David and Paolo in [17] inspected the procedure which demonstrates that how application connects with the working framework and how (PH) IDS can be broken without recognition, by utilizing the system of arrangement coordinating, embeddings malevolent grouping and embeddings no-operation. This method is unconscious about that how much exertion and information is required to create such an assault and furthermore ignorant about that how assailants can foresee that how IDS really functions.

Problem Identified:

- Whole process was supervised, so for training of recurrent neural network one has to know about prior class of the intrusion [1, 4].
- In recurrent neural network weight updating was done after forwarding all training features, which affect the training [5].
- In SVM [6] two class classification was good but in case of multiclass results are not good.
- Detection accuracy was very low [15].
- Execution time was high [1]. Considering the text features of the dataset increase the confusion of the neural learning.

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319-5940
International Journal of Advanced Research in Computer and Communication Engineering
Vol. 8, Issue 3, March 2019

III. PROPOSED SOLUTION

Whole work RFGIID (Random Forest Gini Index Based Intrusion Detection) is divide into two modules, first is training shown in fig 1 and 2, second is testing. So as to improve the investigation include dataset need to be prepare for training the neural network of the current updated dataset sessions.

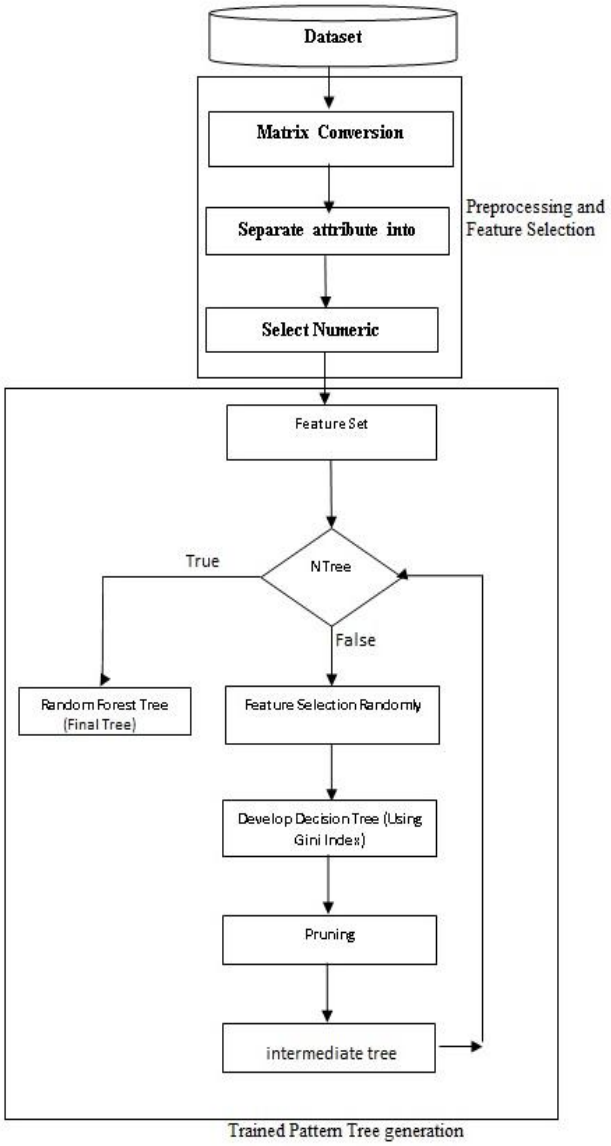

Fig. 1 Flow Chart of proposed work

Dataset Preprocessing Module So as to build the proficiency of the work dataset ought to be pre-process as the preprocessing of Raw Dataset. Instead of direct contribution of crude dataset to choose classifier, crude dataset is preprocessed in various approaches to beat diverse issues like preparing overhead, classifier perplexity, false cautions and recognition rate proportions.

Separating feature space from one another is very necessary and arrange in vector. Let us consider single vector Ds of the dataset and n number of events load in the Vs vector.

{0, tcp, ftp_data,
SF,491,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,2,0.00,0.00,0.00,0.00,1.00,0.00,0.00,150,25,0.17,0.03,0.17,0.00,0.00,0.00,0.05,0.00, normal,20}

In above vector presence of comma ‘,’ and discarding symbolic characters that are of three kind s of symbolic features (tcp, ftp_data and SF etc.) in feature space of 41 features. As symbolic values are not of interest to our research, these

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319-5940
International Journal of Advanced Research in Computer and Communication Engineering
Vol. 8, Issue 3, March 2019

three feature vectors are discarded to get the feature space this is shown in fig 1. So, after the preprocessing the obtain vector is where all element is requiring for dataset analysis.

Pv[] ←Pre-Process (Vs)

{491,0,1,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,2,2,0,0,0,0,1,0,0,110,25,0.17,0.03,0.17,1,0.0,0.00,0.05, 0, normal}

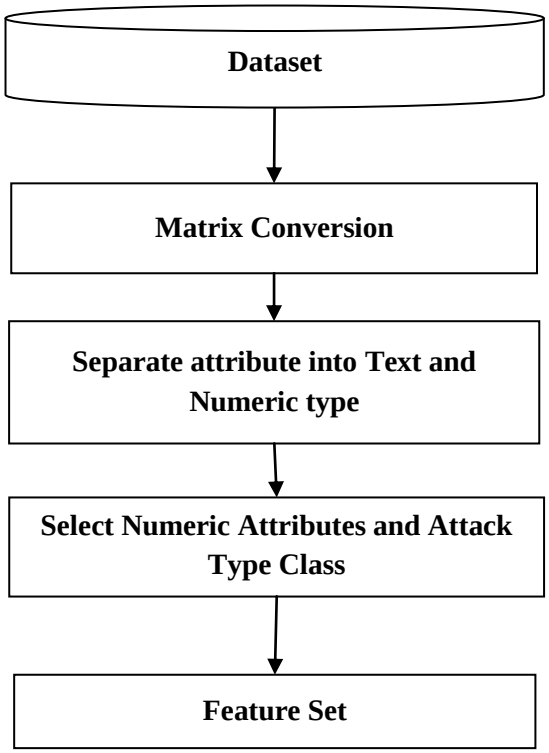

Fig. 2 Dataset Preprocessing and Feature Selection

Features Selection

Feature choice is a vital factor in Network interruption discovery framework for starting training. Since, the vast quantities of features that can be observed considering the substantial attacks of conceivable qualities particularly for consistent element notwithstanding for a little system. Presently the acquire vector contain two imperative features for choosing the features, first is the example of the diverse sort of class in numeric, for example,

{491, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 4
0,0,0,0,0,2,0,2,2,0.00,0.00,0,0.00,1.00,0.00,0.00,150,25,0.17,0.03,0.17,1.00,0.00,0.00,0.05,0.00}

what's more, other is the class name, for example, {normal}. In the comparable style diverse example of same class are gather in the single vector and use them to choose the sort of assault or ordinary system. Second component from the vector is the class name, for example,

Ci [] = {normal, neptune, ipsweep, portsweep}.

Random Forest

The random forests [7] is an ensemble of unpruned classification or regression trees. Random forest generates many classification trees. Each tree is constructed by a different bootstrap sample from the original data using a tree classification algorithm. After the forest is formed, a new object that needs to be classified is put down each of the tree in the forest for classification. Each tree gives a vote that indicates the tree's decision about the class of the object. The forest chooses the class with the most votes for the object.

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319-5940
International Journal of Advanced Research in Computer and Communication Engineering
Vol. 8, Issue 3, March 2019

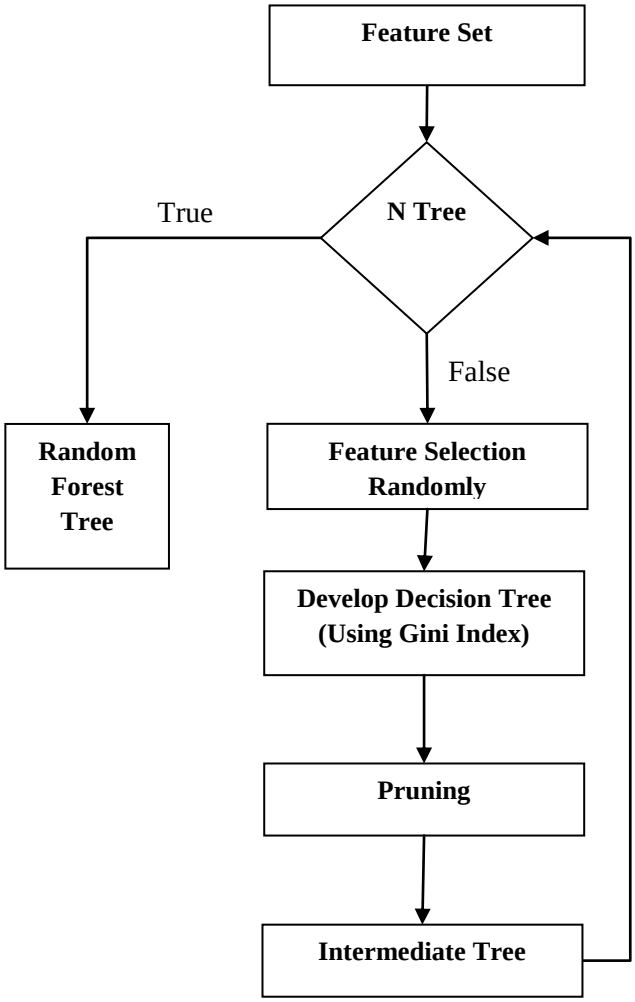

Fig. 3 Trained Pattern Tree generation

Feature Selection Randomly: As random forest has n number of tree and their feature set is different. In this work for each tree in forest random features were select. This can set of columns like [5, 10, 12, 17, 19, 20, 27, 28, 29, 33], OR [2, 6, 8, 11, 15, 22, 17, 18, 29, 33], etc. Based on this feature set random tree was built in next step.

Develop Decision Tree: The process of tree building begins by splitting the root node into two child nodes. CART computes the best split by considering all probable splits for each independent or explanatory variable. The best split is obtained when the impurity function, which exists between the parent node and two child nodes, is minimized. The best split equation is given as:

$$GI \quad \sum_{C\epsilon\ \{Y,N\}} -P_{c,L} \log \quad (P_{c,L}) + \sum_{C\epsilon\ \{Y,N\}} -P_{c,R} \log \quad (P_{c,R})$$

Where $P_{C,L}$ is proportion of total number of elements move towards Left side of tree to the total number of elements in the input dataset. In similar fashion $P_{C,R}$ is proportion of total number of elements move towards Right side of tree to the total number of elements in the input dataset. Where C is number of class element which need to be finally classified. In this way one value of the Gini index obtain for the feature set column value. In similar fashion other values of the Gini index were obtained from the other set of feature column. At last the highest gain or Gini Index value is consider as the final node value for the partition.

Pruning: For a complex or larger tree grown on the initial step of CART, though the prediction of data is described correctly, the prediction accuracy of the tree is low for new samples. Therefore, there is a need to build a tree with better accuracy and predictive ability. Pruning develops an optimal tree, by shedding off the branches of the large tree. The pruning procedure develops a sequence of smaller trees and computes cost complexity for each tree. Based on the cost-complexity parameter, the pruning procedure determines the optimal tree with high accuracy. The cost-complexity

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319-5940
International Journal of Advanced Research in Computer and Communication Engineering
Vol. 8, Issue 3, March 2019

parameter R is set forth as a linear combination of tree complexity and cost associated with the tree. Complexity is given by the following equation:

$$Cn = \frac{Misclassified\_Elements}{Total\_Elements}$$

Where n is number of nodes in a tree and elements are number of sessions classified by the node. Misclassified means elements (session) which are incorrectly classified in the tree.

Testing of Random Forest

As for testing the trained network dataset is again required with different vector, of different or may be of same pattern of the classes. Here it also needs to make the feature vector of all the vector for testing from the trained random forest pattern, but only numeric feature is collected in the Fv then as per training the values of the network is obtained that the input vector is belong to which class. Here feature is pass as per random tree feature set. Each tree gives its own output and majority of tree output is consider as final class of the input session. It may be normal or intrusion.

Proposed Training Algorithm
Input: D // Dataset
Output: RFT // Random Forest Tree

1. Pv←Pre_Process(D) // Pv: Preprocessed Dataset
2. Loop 1:n // n: number of training dataset sessions
3. FS[N, C] ←Generate_Feature(Pv) // FS: Feature Set, N: Numeric value, C: class
4. EndLoop
5. Loop 1:m // m: number of trees in forest
6. Rfs ←Feature_Selection_Randomly(N) // Rfs: Random Feature set
7. DT ←Decision_Tree (Rfs, FS) // DT: Decision Tree
8. DT←Prunning (DT)
9. RFT ←DT
10. EndLoop

Proposed Testing Algorithm
Input: RFT, TD// Testing Dataset
Output: PC // Predicted Class

1. Pv←Pre_Process (TD) // Pv: Preprocessed Dataset
2. Loop 1:n // n: number of training dataset sessions
3. FS←Feature(Pv) // FS: Feature Set, N: Numeric value, C: class
4. Loop 1:m
5. DT←RFT[m]
6. C[m]←DT(FS)
7. EndLoop
8. PC←Max(C)
9. EndLoop

IV. EXPERIMENT AND RESULTS

Data Set For the evaluation of the whole work the dataset is NSL KDD [12] about which previous chapter has already explained and the collection of the all evaluating vectors look like. Where numeric terms are used for feature learning and at the end of each vector it has the corresponding class. The pre-processing step and its requirement have been already explained.

Evaluation Parameter
To test our result this, work use following measures the accuracy of the, that is to say Precision, Recall and F-score. These parameters are depending on the TP, TN, FP and FN.

$$\text{Precision} = \frac{True\_Positive}{True\_Positive + False\_Positive}$$

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 8, Issue 3, March 2019

$$\text{Recall} = \frac{True\_Positive}{True\_Positive + False\_Negative}$$

$$F\_Score = \frac{2 * Precision * Recall}{Precision + Recall}$$

Here True_Positive is counter which get increase when system says intrusion for a session and actually session was intrusion. False_Positive is counter which get increase when system says non-intrusion class for a session and actually session was intrusion. True_Negative is counter which get increase when system says intrusion for a session and actually session was non-intrusion class. False Negative is counter which get increase when system says non-intrusion for a session and actually session was non-intrusion. In order to make the better evaluation for this work one more parameter has introduced that is accuracy of the class of the intrusion. Accuracy of the work is calculating by:

Accuracy = (true positives +false negatives) *100/ (Total_Normal + Total_Intrusion)

Results

Here proposed algorithm results were compared with previous work specified in [1] where Recurrent Neural Network was used for identifying the Dataset intrusion class.

Table 1. Precision value comparison of RNN and RFGIID at different Dataset Sizes.

Data-Set Size	Precision Value Comparison		
	RNN (Existing)	RFGIID (Proposed)	Improvement
3000	0.879694	0.998333	0.118837
4000	0.88109	0.9981	0.117233
5000	0.8795	0.9984	0.119091
6000	0.877468	0.998433	0.121155
7000	0.8777	0.9978	0.120365
8000	0.874707	0.99718	0.122819
9000	0.8763	0.99644	0.120569

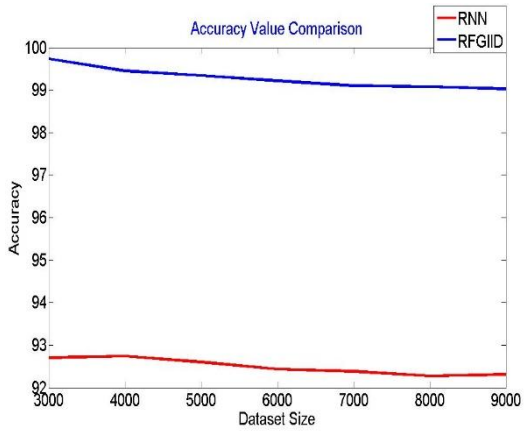

Fig. 3 Comparison of precision values of RNN and RFGIID algorithms.

From above Table1, and fig. 3 it is obtained that with the increase in dataset size precision value rate increase. As number of patterns are more in the dataset so results are more accurate. Here it was shown that use of random forest tree with Gini index increase the precision value.

Table 2 Recall value comparison of RNN and RFGIID at different Dataset Sizes.

Data-Set Size	Recall Value Comparison		
	RNN (Existing)	RFGIID (Proposed)	Improvement
3000	0.978754	0.994934	0.016262
4000	0.9789	0.9910	0.01221

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319-5940
International Journal of Advanced Research in Computer and Communication Engineering
Vol. 8, Issue 3, March 2019

5000	0.9785	0.9891	0.010717
6000	0.977995	0.986989	0.009113
7000	0.9775	0.9854	0.008017
8000	0.976427	0.9858	0.009508
9000	0.9767	0.9855	0.008929

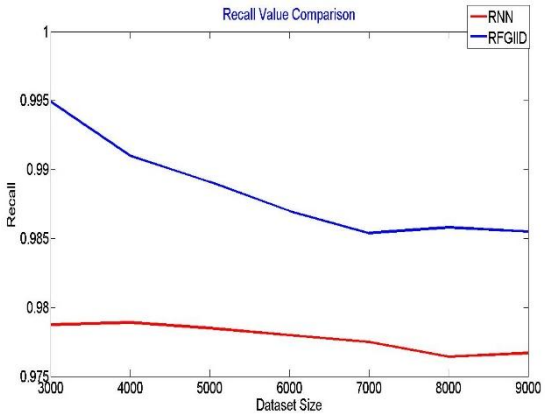

Fig. 4 Comparison of recall values of RNN and RFGIID algorithms.

From above table 2 and fig. 4 it is obtained that with the increase in dataset size recall value rate increase. As number of patterns are more in the dataset so results are more accurate. Here it was shown that use of random forest tree with Gini index increase the recall value.

Table3 F-Measure value comparison of RNN and RFGIID at different Dataset Sizes.

Data-Set Size	F-Measure Value Comparison		
	RNN (Existing)	RFGIID (Proposed)	Improvement
3000	0.926584	0.9974	0.071001
4000	0.9274	0.9945	0.067471
5000	0.9264	0.9937	0.067727
6000	0.925	0.992678	0.068177
7000	0.9246	0.9916	0.067568
8000	0.922	0.99148	0.070077
9000	0.9238	0.999	0.075275

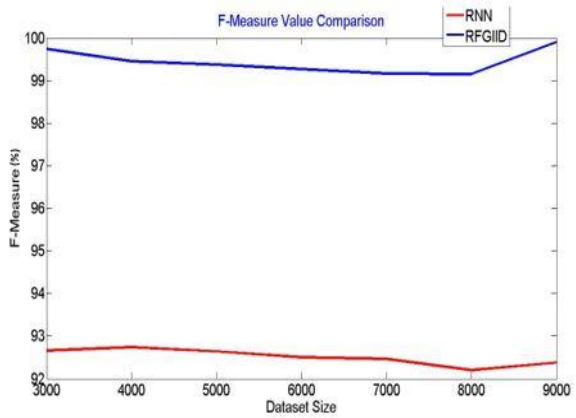

Fig. 5 Comparison of F-Measure values of RNN and RFGIID algorithms.

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 8, Issue 3, March 2019

From above table 3 and fig. 5 it is obtained that use of random forest tree with Gini index in proposed work has high F-measure value as compared to previous work. Here it was shown that use of new approach of neural network training reduce the execution time as compared to RNN used in previous method.

Table 4. Execution time value comparison of RNN and RFGIID at different Dataset Sizes

Data-Set Size	Training Execution time (second) Value Comparison	
	RNN (Existing)	RFGIID (Proposed)
3000	19.8547	18.6253
4000	17.62	17.3
5000	23.32	1738
6000	40.2612	23.6892
7000	35.36	29.68
8000	54.0972	33.5678
9000	74.33	22.21

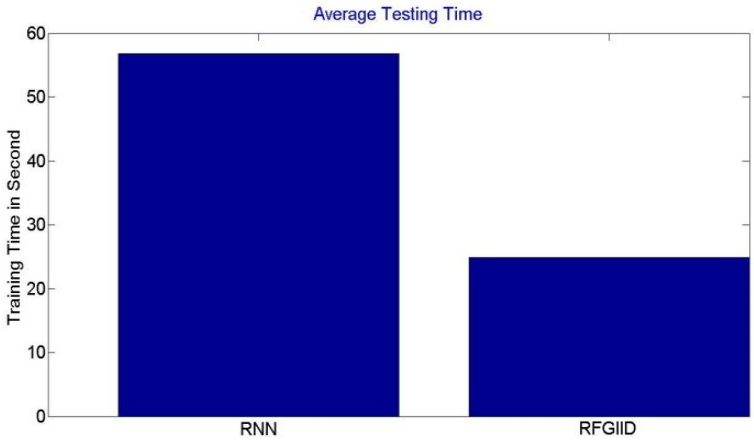

Fig. 6 Comparison of training time for RNN and RFGIID algorithms.

From above table 4 and fig. 6 it is obtained that with the increase in dataset size execution time value increase. Here it was shown that use of new approach of random forest tree with Gini index for training reduce the execution time as compared to RNN used in previous method.

Table 5 Execution time value comparison of RNN and RFGIID at different Dataset Sizes

Data-Set Size	Testing Execution time (second) Value Comparison	
	RNN (Existing)	RFGIID (Proposed)
3000	38.7396	22.5932
4000	34.32	17.9
5000	36.76	19.05
6000	59.391	24.6259
7000	46.60	20.4
8000	72.1846	27.2998
9000	57.3	22.71

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319-5940
International Journal of Advanced Research in Computer and Communication Engineering
Vol. 8, Issue 3, March 2019

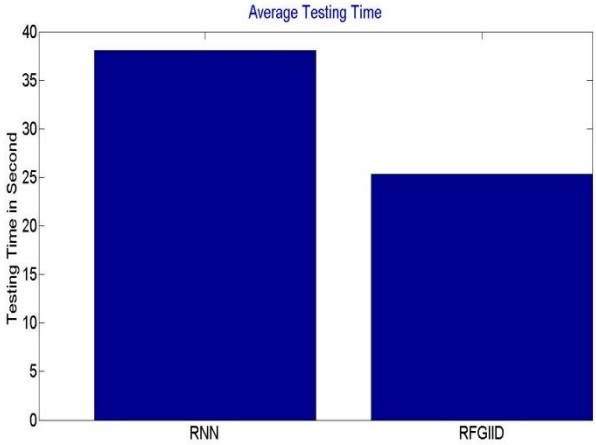

Fig. 7 Comparison of testing time of RNN and RFGIID algorithms

From above table 5 and fig. 7 it is obtained that with the increase in dataset size execution time value increase. Here it was shown that use of new approach of random forest tree with Gini index for training reduce the execution time as compared to RNN used in previous method.

Table 6. Accuracy value comparison of RNN and RFGIID at different Dataset Sizes

Data-Set Size	Accuracy Value Comparison		
	RNN (Existing)	RFGIID (Proposed)	Improvement
3000	92.7	99.733	7.051828
4000	92.74	99.45	6.747109
5000	92.6	99.34	6.78478
6000	92.433	99.2167	6.837256
7000	92.38	99.1	6.781029
8000	92.27	99.08	6.873234
9000	92.31	99.03	6.785822

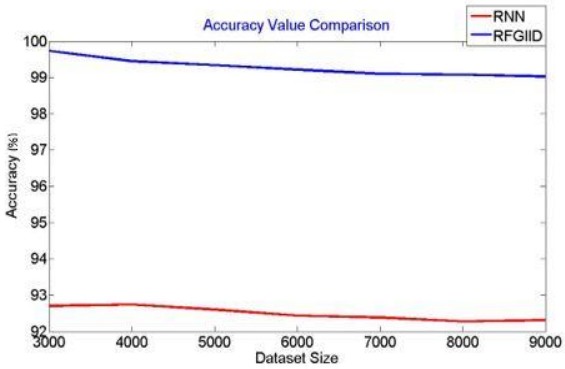

Fig. 8 Comparison of accuracy values of RNN and RFGIID algorithms.

From above table 6 and fig. 8 it is obtained that with the increase in dataset size execution time value increase. Here it was shown that use of new approach of random forest tree with Gini index for training reduce the execution time as compared to RNN used in previous method.

Table 7. RNN based Detection Rate values of various attacks at different Dataset sizes

Dataset Size	RNN (Previous work)			
	DOS	Probe	U2R	R2L
3000	96.9093	43.4783	48.889	43.157
4000	96.7596	40	50	40.678

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319-5940
International Journal of Advanced Research in Computer and Communication Engineering
Vol. 8, Issue 3, March 2019

5000	96.8637	33.8983	49.3671	38.775
6000	96.9207	30.7692	50.5376	40.588
7000	96.8939	28.7671	49.5413	41.052
8000	96.7269	27.2727	50	40.186
9000	96.693	24.418	50	37.759

Table 8. RNN based Detection Rate values of various attacks at different Dataset sizes.

Dataset Size	RFGIID (Proposed work)			
	DOS	Probe	U2R	R2L
3000	100	96.1538	49.6689	100
4000	99.9246	93.333	51.4706	100
5000	99.8794	89.7436	56	99.3197
6000	99.899	84.4444	53.6232	99.4118
7000	99.8274	83.0189	57.3333	99.4737
8000	99.7366	84.2105	51.676	99.5327
9000	99.7662	83.3333	54.712	98.3402

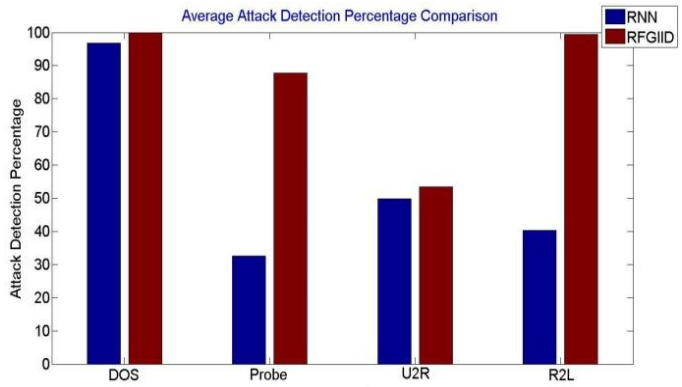

Fig. 9 Comparison of average attack detection percentage of RNN and RFGIID algorithms

From above fig. 9, table 7 and 8 it is obtained that proposed work intrusion detection system using RFGIID approach can find the attack class bitterly as compared to previous work. Here in some of case 100 percent attacks are distinguished also. Here it was demonstrated that utilization of new methodology of RFT with Gini concept for training builds the discovery precision when contrasted with RNN utilized in past strategy.

CONCLUSION

System security is a standout amongst the most critical nonfunctional basics in a framework. Throughout the years, numerous product arrangements have been created to upgrade organize security and this paper gives an effective framework which has been a promising one for distinguishing interruption of various kind where, one can get the detail of the class of attack also. Results demonstrates that all sort of attack are precisely distinguished by the framework as the exactness esteem is above 96%. In future it should be enhanced by putting information on the unsupervised system, so it naturally refreshes the new conduct of the attacker. One more issue remain in this work is to use dynamic adaptable technique for learning new type of attack.

REFERENCES

[1]. ChuanlongYin ,Yuefei Zhu, Jinlong Fei, And Xinzheng He. "A Deep Learning Approach for Intrusion Detection Using Recurrent Neural Networks" current version November 7,2017.
[2]. A.R. Jakhale, G.A. Patil, "Anomaly Detection System by Mining Frequent Pattern using Data Mining Algorithm from Network Flow", International Journal of Engineering Research and Technology, Vol. 3, No.1, ISSN. 2278-0181, January 2014.
[3]. Aljurayban, N.S.; Emam, A. Framework for Cloud Intrusion Detection System Service. Web Applications and Networking (WSWAN), 2015 2nd World Symposium on, p1-5, 2015.
[4]. Barolli, Leonard; Elmazi, Donald; Ishitaki, Oda, Tetsuya; Taro; Yi Liu, Uchida, Kazunori.. Application of Neural Networks for Intrusion Detection in Tor Networks. Advanced Information Networking and Applications Workshops (WAINA), 2015 IEEE 29th International Conference on, p67-72, March 2015.
[5]. KoushalKumar, Jaspreet Singh Batth "Network Intrusion Detection with Feature Selection Techniques using Machine-Learning Algorithms" International Journal of Computer Applications (0975 – 8887) Volume 150 – No.12, September 2016.
[6]. Yogita B. Bhavasar, Kalyani C. Waghmare "Intrusion Detection System Using Data Mining Technique: Support Vector Machine" 2013 International Journal of Emerging Technology and Advance Engineering volume 3, Issue 3, March 2013.

[7]. Premansusekhararath, Manisha mohanty, Silva acharya, Monica aich "optimization of ids algorithms using data mining technique" International Journal of Industrial Electronics and Electrical Engineering, ISSN: 2347-6982 Volume-4, Issue-3, Mar.-2016
[8]. MohammadrezaEktefa, Sara Memar, Fatimah Sidi, Lilly SurianiAffendey "Intrusion Detection Using Data Mining Techniques", 978-1-4244-5651-2/10/$26.00 IEEE 2010.
[9]. YU-XIN MENG," The Practice on Using Machine Learning for Network Anomaly Intrusion Detection" Department of Computer Science, City University of Hong Kong, Kowloon, Hong Kong, 978-1-4577-0308- IEEE2011
[10]. Liu Hui, CAOYonghui "Research Intrusion Detection Techniques from the Perspective ofMachineLearning" 2010 Second International Conference on MultiMedia and Information Technology 978-0-7695-4008-5/10 $26.00 IEEE 2010.
[11]. https://www.github.com/defcom17/ NSL_KDD/blob/ master/ Original%20NSL%20KDD%20Zip.zip.
[12]. Nouf Saleh Aljurayban, Ahmed Emam "Framework for cloud intrusion detection system service". **DOI:**10.1109/ WSWAN.2015. 7210298, IEEE, 20 August 2015.
[13]. Zhiyuan Tan, ArunaJamdagni, Xiangjian, Priyadarsi Nanda, Ren Ping Liu, "A System for Denial-of-Service Attack Detection Based on Multi-variate Correlation Analysis", IEEE Transactions On Parallel And Distributed Systems Vol:25 No:2 Year 2014.
[14]. Mr Mohit Tiwari,Raj Kumar, Akash Bharti, Jai Kishan. "Intrusion Detection System". International Journal of Technical Research and Applications e-ISSN: 2320-8163, Volume 5, Issue 2 March - April 2017.
[15]. Mario Guimaraes, Meg Murray. "Overview of Intrusion Detection and Intrusion Prevention". Information security curriculum development Conference by ACM (2008).
[16]. Muhammad Awais Shibli, SeadMuftic. "Intrusion Detection and Prevention System using Secure Mobile Agents". IEEE International Conference on Security & Cryptography 2008.
[17]. David Wagner, Paolo Soto. "Mimicry Attacks on Host Based Intrusion Detection Systems". 9th ACM Conference on Computer and Communications Security 2002.