International Journal of Technology Research and Management
ISSN (Online): 2348-9006
Vol 5 Issue 3 March 2018

An Efficient Feature Selection Approach for Network- based Intrusion Detection System Using Machine Learning Algorithm

Yogadhar Pandey[1], Prof. Shailendra Singh[2]
Senior Member, IEEE, NITTTR, Bhopal, India[2]
P_yogadhar@yahoo.co.in[1], ssingh@nitttrbpl.ac.in[2]

Abstract: *Network-based intrusion detection systems (NIDS) have become important and widely used tools for ensuring network security. Processing huge amount of audit data is a challenging task for NIDS, because these data contains large amount of irrelevant or redundant features. To improve the accuracy and efficiency of NIDS, relevant features are essential to be extracted from original dataset with the help of feature selection methods. This paper proposes reduced feature selection algorithm (RFSA) an efficient feature selection approach for network based intrusion detection system. The RFSA gives better detection rate, accuracy and false alarm rate when compared to full feature set. RFSA also outperform when compared with the existing algorithms. Empirical results endorse the overall performance and accuracy of the system. This work uses KDD99 dataset for the evaluation of the experiments.*

Keywords: *NIDS;Feture Selection;Machine Learning;Classification;KDD'99Dataset.,RFSA.*

1. INTRODUCTION

The rapid advancement of the Internet has made it a potent platform for communication, business, entertainment, research, investigation, and other uses. Advancement in Internet connectivity resulted in data breaches and security vulnerabilities. Internet-based security attacks have multiplied in the last few years.

Intrusion Detection System (IDSs) [1] is employed to check and evaluate network traffic for a given set of parameters to discover potentially destructive network communication. An IDS is a software application that examines all activities happening over the network and identifies suspicious patterns that may indicate that there has been a system or network attack by someone attempting to bypass the security mechanisms in place.

An IDS is categorized in many ways, based on a collection of data, analysis of data, and actions needed to be taken. It can also be classified, based on the position of installation in the network, into two types: HIDS (Host-based IDS) and NIDS (Network-based IDS).

HIDS (Host-based Intrusion Detection System)

HIDS [2] runs on the machine it monitors, it can theoretically observe and log any event occurring on the machine. HIDSs are confronted with difficulties arising from potential tampering by the attacker. A safe and sound logging mechanism is essential to avoid logs from being erased if the attacker compromises the machine. If such a mechanism is available, an attacker gaining super user privilege on the host can disable the HIDS: if a user process is running on HIDS, an attacker can easily abort the process. If it is set in the kernel, the attacker can adjust the kernel by writing a kernel module or by putting it directly in kernel memory. That way, an HIDS can only be relied upon to the extent where the system was compromised.

NIDS (Network-based Intrusion Detection System)

A network-based intrusion detection system (NIDS) [2] is used to monitor and analyze network traffic to protect a system from network-based threats.

An NIDS reads all inbound packets and searches for any suspicious patterns. When threats are discovered, based on its severity, the system can take action such as notifying administrators, or barring the source IP address from accessing the network.

NIDS is very useful in detecting network related attacks. Although there are still many issues in the research of NIDS, the following two issues appear to be the most challenging.

The first challenge is the high false alarm rate in the anomaly NIDS, which has formed a hurdle for practical applications; therefore, reducing this high false alarm rate has become an intensive ongoing research topic. The second challenge is the detection of attacks that is likely to generate little network traffic and attacks originating from inside the protected network. There are some harmful network attacks that do not generate significant network traffic [4].

Intrusion Detection Techniques

Principally Intrusion Detection Systems use one of two detection techniques [4] statistical anomaly based and/or signature based. The signature based IDS [5] are also known as misuse detection. It monitors the traffic for a given signature to match, indicating an intrusion. With the help of provided signatures or patterns, it can identify many or all known attack patterns. A signature based intrusion detection system has some drawbacks also. A signature is to be formed for each and every attack and they are able to detect those attacks. They are not capable to identify newer attacks as their signatures are unaware of the detection mechanism.

The Anomaly based intrusion detection system [6] analyzes the behavior of the network traffic. Such detection possesses the potential to distinguish anomalous behavior by analyzing the huge volume of traffic. A sudden flow in traffic from a specific host or to a specific host causes load imbalance in the network. The main drawback of this method is that if the malicious activities are categorized as normal network behavior then it would lead to anomaly. Major benefit of anomaly detection over the signature based detection is that a new attack for which a signature does not exist can also be detected if it behaves in a different way from normal traffic behavior. There are various techniques proposed for network intrusion detection to explore the deficiency and research gap in certain algorithms which perform better for certain attack classes.

Machine Learning

The field of machine learning is concerned with the higher-level question of how to construct computer programs that automatically learn with experience.

Machine Learning Techniques

In recent years, machine learning systems have been extended for implementing efficient intrusion detection methodologies. Machine learning techniques are very efficient and enhanced for current intrusion detection. Support vector machines [7], neural networks [8] and decision trees exhibit efficient functional mechanisms in anomaly detection systems.

Various well-known machine learning techniques can be used in for intrusion detection approach. The advantage of IDS (Intrusion Detection system) can greatly reduce the time for network administrators/users to analyze network data and protect the network from illicit attacks. Intrusion detection system (IDS) is used to detect various kinds of attacks in interconnected network. Hacker’s probe and attack computer networks each day. These attacks range from relatively benign ping sweeps to sophisticated techniques exploiting security vulnerabilities. Intrusion detection is the task of detecting and responding to this kind of computer misuse, by detecting unauthorized access to a computer network. Intrusion detection systems are “systems that collect information from a variety of network sources, and then analyse the information for signs of intrusion and misuse”.IDS is a device, usually a designated computer system that monitors activity to identify malicious or suspicious alerts. IDS can be compared with a spam filter that raises an alarm if specific things occur. Intrusion Detection System (IDS) are software or hardware systems that automate the process of monitoring and analyzing the events that occur in a computer network, to detect malicious activity. Since the severity of attacks occurring in the network has increased drastically, Intrusion detection system have become a necessary addition to security infrastructure of most organizations.

2. LITERATURE SURVEY

Intrusion detection systems suffer from a problem associated with dimensionality. Enormous datasets which mimic real network data impose increased overheads of training and testing in IDS. Poor detection ability is caused due to enormous datasets which also leads to consumption of resources. Before processing, the data that is not responsible for detection must be eliminated. This helps in the development of efficient feature extraction and reduction methods, which reduces the training time and also provides better accuracy. Different authors proposed different methods for feature selection.

Olusola, Adetunmbi A. et al. [5] presented an approach for the selection of relevance features carried out on KDD cup99 dataset for the detection of each class. To determine the most discriminating features for each class, Rough set degree of dependency and dependency ratio is used. Results show that some features have no relevance in intrusion detection. Features comprise 20 and 21 (outbound command count for FTP session and hot login) while features 13, 15, 17, 22 and 40 (no. of compromised conditions, su attempted, number of file creation operations, is guest login, dst host rerror rate respectively) are less significant for the intrusion detection .It is observed that seven features were not found relevant in the detection of any class. So an effective feature reduction approach is required before passes to the classification algorithms.

Amiri et al.[6] Proposed feature selection algorithm and compared the performance with the mutual information based feature selection method. In this approach author uses feature goodness measure, where linear and non linear measures has been investigated. Feature measure like: mutual information and linear correlation coefficient used for feature selection. Proposed mutual information based feature selection based method detect intrusion with great accuracy for R2L and U2R with 90.91% and 93.16% respectively.

Shafigh Parsazad et al.[7] proposed a very simple and fast feature selection method to eliminate features which has no useful information. They compared this method with three most successful similarity based feature selection algorithm including Correlation Coefficient, Least Square Regression Error and Maximal Information Compression Index. After that recommended features used by each of these algorithms in two popular classifiers including: Bayes and KNN classifier to measure the quality of the recommendations. Result shows that classification accuracy and detection rate didn’t improved much, but execution rate of FFR is better than the other feature selection methods. Some of the feature which are not relevance are needed to be extracted. So an efficient algorithm is needed to solve the problems of feature reduction.

M R G Raman et al. [8] developed a rough set Hyper-graph technique for key feature identification in intrusion detection systems. Minimal traversal and Vertex linearity properties of Hyper-graph are used for Identification of featured subset. Training and testing dataset obtained from the 10 % of KDD99 dataset. A result of proposed technique dominates over the existing techniques in terms of classification accuracy and computation time.

Md. Mehedi Hasan et al.[9] have presented a two-step approach for feature selection based on Random Forest. In the beginning features with higher variable importance score are selected and the initialization of search process is guided for the next step outputs of which are the required feature subset for classification. The effectiveness of this algorithm is demonstrated on KDD’99 intrusion detection dataset. In KDD’99 dataset huge number of redundant records is present. Therefore they derived a data set RRE-KDD by eliminating redundant record from KDD’99 training and test dataset, so the classifiers and feature selection method will not be biased towards more frequent records. RRE-KDD comprises of KDD’99 training and test dataset for training and testing respectively. It has been observed that the Random Forest based proposed method can select only important and relevant features which are useful for classification, that not only reduces the number of input features and time but also enhances the classification accuracy.

Ishfaq Manzoor et al. [10] proposed a feature reduction method using ANN-based classification. Features have been reduced using ranker attribute selection methods like: InfoGain, Correlation-based Feature selection. Proposed method for U2R and R2L achieve detection rate of 86.6% and 91.9% respectively. Precision for U2R and R2L classes are 42.88% and 87.5% respectively. Features still needs to be reduced to get an optimal featured set and precision should be improved which in turn reduce the model building time. Thus an efficient approach is needed for feature reduction.

In literature different authors has applied different feature selection methods to decrease the dimensionality of the KDD99 dataset. When features are reduced, time taken for model generation also gets reduced. But for U2R and R2L classes the precision is not much satisfactory .Thus an effective feature selection method is required to improve the accuracy and lower the false alarm rate (FAR) of the system, which in turn improve the overall performance of the intrusion detection system.

Paper in Elsevier: In this paper the author tries to obtain a reduced feature set for NBIDS. The main shortcoming of the paper [10] is that this work does not perform experiment on full dataset. They have worked on a subset of the original dataset which has lower imbalance than the original problem. This simplification decreases the merit of the results obtained.

3. FEATURE SELECTION

Feature selection [11] is a very efficient way to reduce the dimensionality of a problem. Redundant and irrelevant variables are removed from the data before being fed to the machine learning algorithm used as a classifier. Feature selection is a preprocessing step which can be independent of the choice of the learning algorithm or not. It can be used in order to improve the computational speed with minimum reduction of accuracy.

Feature selection process involves four basic steps in a typical feature selection method shown in Figure 1.

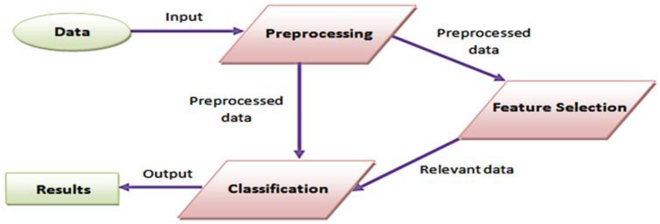


Figure 1: Feature Selection Process

First is subset generation procedure to generate the next candidate subset; second one is an evaluation function to evaluate the subset and third one is a stopping criterion to decide when to stop; and a validation procedure to check whether the subset is valid.

4. KDD’99 Dataset

The data set in our experiment is the data set for 1999 KDD cup machine learning competition[12], which is a subset of the 1998 DARPA Intrusion Detection Evaluation data set, and is processed, extracting 41 features from the raw data of DARPA 98 data set. Attacks fall into four main categories: DoS U2R R2L PROBE. In Table 1&2 details of KDD99 has been presented.

Table 1: Attack Types in KDD99 Dataset

Class	**Attack Type**
DoS	apache2, back, land, mailbomb, neptune , pod, processtable, smurf, teardrop,dpstrom
Probe	ipsweep,mscan,nmap, portsweep, saint
R2L	ftp_write, guess_passwd, imap, multihop, named, phf, sendmail, spy, snmpgetattack, snmpguess , war ezclient, warez master, worm, xlock, xsnoop
U2R	buffer_over flow, httptunnel, loadmodule, perl, ps, rootkit, sqlattack, xtern

Table 2: 10% KDD99 Dataset (training)

Dataset **(KDD99)**	Normal	DoS	Probe	U2R	R2L	Total **494021**
Instances	97278	391458	4107	52	1126	

Table 2 reprsents the number of instances of each attack classes in 10% KDD99 dataset.total 494021 instances are present for training purposes.

Table 3: KDD99 Test Dataset

Dataset KDD99	Normal	DoS	Probe	U2R	R2L	Total
Instances	60593	237594	4156	70	8606	**311029**

Initially we haveKDD99 10% training dataset, then Proposed feature selection algorithms will be applied on it. Table 3 represents the Test dataset instances.

5. PROPOSED REDUCED FEATURE SELECTION ALGORITHM(RFSA)

Proposed feature selection Approach effectively reduced the Dimensionality of the KDD99 dataset from 41 features to 22 features. We evaluated the ranking of all 41 attributes using the different existing attribute selection methods like Gain Ratio Attribute Eval, One R Attribute Eval, Classifier Attribute Eval[13] with the Rank Search method. We have considered 1:10 and then next 11:20 attributes of each featured set of Gain Ratio Attribute Eval, One R Attribute Eval,

Classifier Attribute Eval respectively. We have evaluated (Fs1UFs3UFs5)

The ranked attributes gives subset of the relevant features of each algorithm. Flow chart of the Proposed Algorithm is given in the figure as.

Algorithm

Step1→Sort the features in order of their importance using the Gain Ratio, One R Attribute, Classifier Attribute algorithm

Step2→ this work assumes that the first 25% of the ranked attributes should be present in the reduced feature set. Let Fs1, Fs3 and Fs5 represent the top 25 % ranked features obtained using Gain Ratio, One R Attribute, Classifier Attribute feature selection methods. The reduced feature set should include all these features so we have to compute Fs1 U Fs3 U Fs5)

Step3→the reduced feature set which are common in the next 25 % of the ranked attributes should also be included in the reduced feature set. . Let Fs2, Fs4 and Fs6 represent the next 25 % ranked features obtained using Gain Ratio, One R Attribute, Classifier Attribute feature selection methods. Compute (Fs2∩ Fs4∩fs6)

Step4→ Calculate the Reduced feature set (Fs1UFs3UFs5) U (Fs2∩ Fs4∩fs6)

Step5→obtained RFS(reduced feature set) from step4 which includes 22 featured set

Step6→Apply Different classification Algorithms like (J48, Random Tree, Random Forest)

Step7→Evaluate Results from the Confusion Matrix (output of the classifiers).

We have applied proposed algorithm on dataset KDD99 step1 to step7as per above discussion.

Computation for Gain Ratio:
First 1-10: 14, 12, 11, 22, 9, 6, 37, 17, 18, 3→Fs1
Next 11-20: 32, 31, 5, 2, 1, 16, 10, 36, 23, 19→Fs2

Computation for One R Attribute:
First 1-10: 5, 23, 3, 6, 12, 36, 32, 37, 24, 31 →Fs3
Next 11-20 :33,35,34,2,1,39,41,38,40,30 →Fs4

Computation for Classifier Attribute:
First 1-10: 41,13,12,20,14,15,16,17,18,11 →Fs5
Next 11-20: 10, 9, 4, 2, 3, 5, 8, 6, 7, 19 →Fs6

RFS→14,12,11,22,9,6,37,17,18,3,5,23,36,32,24,31,41,13,20, 15,16,2.

This reduced feature set is named as reduced feature set will be input for classification process. Flow chart of proposed reduced feature set algorithm is given below to explain the working procedure.

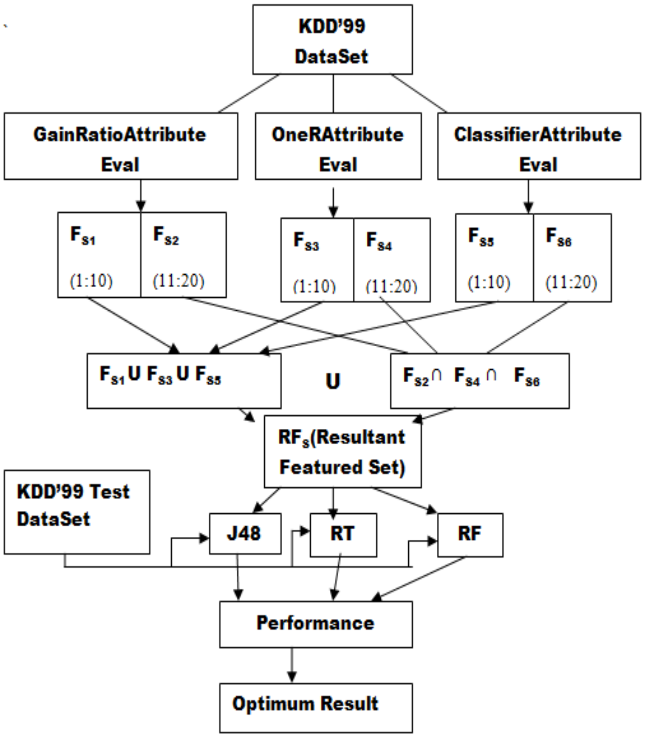


Figure 2: Flow chart of Proposed Feature Selection Approach

6. PERFORMANCE MEASURES

1. The true positives (TP) and true negatives (TN) are correct classifications [14].
2. A false positive (FP) occurs when the outcome is incorrectly predicted as yes (or positive) when it is actually no (negative).
3. A false negative (FN) occurs when the outcome is incorrectly predicted as negative when it is actually positive.
4. Recall: The percentage of the total relevant documents in a database retrieved by your search. If you knew that there were 1000 relevant documents in a database and your search retrieved 100 of these relevant documents, your recall would be 10%.
   Recall = TP/(TP+FN)
5. Precision: The percentage of relevant documents in relation to the number of documents retrieved. If your search retrieves 100 documents and 20 of these are relevant, your precision is 20%.
   Precision =TP/(TP+FP)

6. F-measure: The harmonic mean of precision and recall F = 2 * Recall * Precision / (Recall + Precision)
7. The true positive rate is TP divided by the total number of positives, which are TP + FN.
8. The false positive rate is FP divided by the total number of negatives, FP + TN.
9. ROC area In ROC analysis we plot true positive ratio (tpr) against, false positive ratio (fpr).
10. The overall success rate also called as accuracy is the number of correct classifications divided by the total number of classifications:

$$\frac{TP+TN}{TP+TN+FP+FN}$$

Finally, the error rate is one minus this.

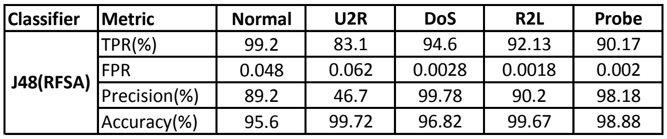
11. In a multiclass prediction, the result on a test set is often displayed as a two dimensional confusion matrix with a row and column for each class. Each matrix element shows the number of test examples for which the actual class is the row and the predicted class is the column. Good results correspond to large numbers down the main diagonal and small, ideally zero, off-diagonal elements.

7. RESULT ANALYSIS & DISCUSSION

Classifiers used for the experiments are J48, random tree, random forest.494021 connections are used for training set and 311029 connections are used for testing set [15, 16, and 17]. This work uses weka 3.8[19] for performing the experiment. Performance of the different classifiers have been evaluated and summarized in table (4, 5, and 6).

J48(C4.5)

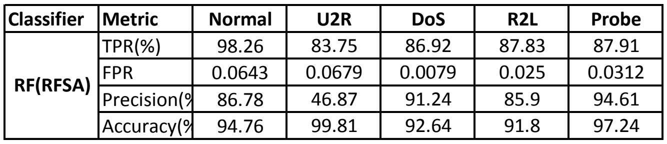
Table 4: Performance evaluation of J48 on Reduced Featureset



Classifier	Metric	Normal	U2R	DoS	R2L	Probe
J48(RFSA)	TPR(%)	99.2	83.1	94.6	92.13	90.17
	FPR	0.048	0.062	0.0028	0.0018	0.002
	Precision(%)	89.2	46.7	99.78	90.2	98.18
	Accuracy(%)	95.6	99.72	96.82	99.67	98.88

RF

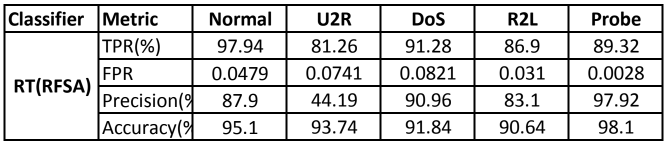
Table 5: Performance evaluation of RF on Reduced Featureset



Classifier	Metric	Normal	U2R	DoS	R2L	Probe
RF(RFSA)	TPR(%)	98.26	83.75	86.92	87.83	87.91
	FPR	0.0643	0.0679	0.0079	0.025	0.0312
	Precision(%	86.78	46.87	91.24	85.9	94.61
	Accuracy(%	94.76	99.81	92.64	91.8	97.24

RT

Table 6: Performance evaluation of RT on Reduced Featureset

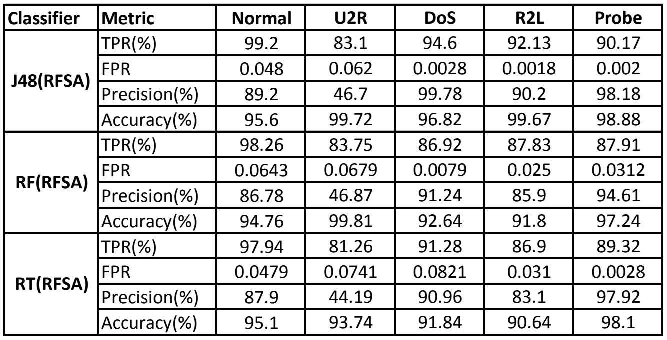


Classifier	Metric	Normal	U2R	DoS	R2L	Probe
RT(RFSA)	TPR(%)	97.94	81.26	91.28	86.9	89.32
	FPR	0.0479	0.0741	0.0821	0.031	0.0028
	Precision(%	87.9	44.19	90.96	83.1	97.92
	Accuracy(%	95.1	93.74	91.84	90.64	98.1

Performance Comparission (J48,RT,RF)

After evaluation we compare the performance of the classifiers on the basis of performance metric (Accuracy, Precision, TPR and FPR).Comparison results summarized in table7. It is Shown that J48 (RFSA) Outperforms for normal, DoS , Probe and R2L classes While accuracy and precision for U2R class, RF shows Slightly better, But when we see Overall performance of the system J48 Outperforms than RT and RF.

Table 7: Comparission of (J48,RT,RF)



Classifier	Metric	Normal	U2R	DoS	R2L	Probe
J48(RFSA)	TPR(%)	99.2	83.1	94.6	92.13	90.17
	FPR	0.048	0.062	0.0028	0.0018	0.002
	Precision(%)	89.2	46.7	99.78	90.2	98.18
	Accuracy(%)	95.6	99.72	96.82	99.67	98.88
RF(RFSA)	TPR(%)	98.26	83.75	86.92	87.83	87.91
	FPR	0.0643	0.0679	0.0079	0.025	0.0312
	Precision(%)	86.78	46.87	91.24	85.9	94.61
	Accuracy(%)	94.76	99.81	92.64	91.8	97.24
RT(RFSA)	TPR(%)	97.94	81.26	91.28	86.9	89.32
	FPR	0.0479	0.0741	0.0821	0.031	0.0028
	Precision(%)	87.9	44.19	90.96	83.1	97.92
	Accuracy(%)	95.1	93.74	91.84	90.64	98.1

Comparison of J48 Performance RFSA(proposed) with Full featured Dataset:

Table 8: Comparison J48(Reduced Vs Full Featured)

Classifier	Metric	Normal	U2R	DoS	R2L	Probe
J48(RFSA)	TPR(%)	99.2	83.1	94.6	92.13	90.17
	FPR	0.048	0.062	0.0028	0.0018	0.002
	Precision(%)	89.2	46.7	99.78	90.2	98.18
	Accuracy(%)	95.6	99.72	96.82	99.67	98.88
J48(With Full Features)	TPR(%)	96.4	76.58	93.52	78.2	83.7
	FPR	0.0679	0.038	0.0042	0.0036	0.0067
	Precision(%)	83.5	42.3	92.67	81.7	92.64
	Accuracy(%)	91.68	86.8	91.89	88.9	93.7

When we compare performance of J48 Reduced featured dataset with the full featured datset, Table8 shows that our proposed (RFSA) algorithm outperforms in all the performance metric (Accuracy, Precision, TPR and FPR).

Comparison of classifer performance between RFSA(proposed) Vs Ishfaq et al[10]

The work in [10] also presents a features selection algorithm which has a major shortcoming that it works on the reduced training and test dataset. The results of our proposed algorithms are compared with [10] are listed in Table 9.

Table 9: Comparison of classifers performance between RFSA(proposed) Vs Ishfaq et al[10]

Classifier	Metric	Normal	U2R	DoS	R2L	Probe
Ishfaq et al.[]	TPR(%)	98.8	86.6	93.8	91.9	89.8
	FPR	0.06558	0.0005	0.0004	0.0028	0.0014
	Precision(%)	88.9	42.88	99.9	87.5	98.14
	Accuracy(%)	94.8	99.93	96.51	99.54	98.79
J48(RFSA)	TPR(%)	99.2	83.1	94.6	92.13	90.17
	FPR	0.048	0.062	0.0028	0.0018	0.002
	Precision(%)	89.2	46.7	99.78	90.2	98.18
	Accuracy(%)	95.6	99.72	96.82	99.67	98.88

8. CONCLUSION AND FUTURE SCOPE

In our proposed reduced, feature selection algorithm (RFSA) reduces 41 features to 22 features. This work includes: Evaluation of reduced feature set using RFSA, which further compared with the full featured set and shows better performance than the full featured set. This work is also compared with the existing work and outperform in all the performance metric. Empirical results endorse the overall system performance. KDD99dataset used for the evaluation of the experiments.

The future work includes to further refine the feature selection algorithm leading for reduction in the number of features. The future work also includes the study of sampling techniques to improve the performance of the minority classes like U2R and R2L.

REFERENCES

[1] Y. Bai and H. Kobayashi, “Intrusion detection systems: technology and development,” in Advanced Information Networking and Applications, 2003. AINA 2003. 17th International Conference on, 2003, pp. 710–715.

[2] C. M. Chen, Y. L. Chen, and H. C. Lin, “An efficient network intrusion detection,” Computer Communications, vol. 33, no. 4, pp. 477–484, 2010.

[3] Casas, Pedro, Johan Mazel, and Philippe Owezarski. "Unsupervised network intrusion detection systems: Detecting the unknown without knowledge." Computer Communications 35, no. 7 (2012): 772-783.

[4] Garcia-Teodoro, Pedro, J. Diaz-Verdejo, Gabriel Maciá-Fernández, and Enrique Vázquez. "Anomaly-based network intrusion detection: Techniques, systems and challenges." computers & security 28, no. 1-2 (2009): 18-28.

[5] Olusola, Adetunmbi A., Adeola S. Oladele, and Daramola O. Abosede. "Analysis of KDD’99 intrusion detection dataset for selection of relevance features." In Proceedings of the World Congress on Engineering and Computer Science, vol. 1, pp. 20-22. 2010.

[6] Amiri, Fatemeh, Mohammad Mahdi Rezaei Yousefi, Caro Lucas, Azadeh Shakery, and Nasser Yazdani. "Mutual information-based feature selection for intrusion detection systems." Journal of Network and Computer Applications 34, no. 4 (2011): 1184-1199.

[7] Parsazad, Shafigh, Ehsan Saboori, and Amin Allahyar. "Fast feature reduction in intrusion detection datasets." In MIPRO, 2012 Proceedings of the 35th International Convention, pp. 1023-1029. IEEE, 2012.

[8] Raman, MR Gauthama, Kannan Kirthivasan, and VS Shankar Sriram. "Development of rough set–hypergraph technique for key feature identification in intrusion detection systems." Computers & Electrical Engineering 59 (2017): 189-200.

[9] Hasan, Md Al Mehedi, Mohammed Nasser, Shamim Ahmad, and Khademul Islam Molla. "Feature selection for intrusion detection using random forest." Journal of information security 7, no. 03 (2016): 129.

[10] Ishfaq Manzoor, Akashdeep and Neeraj Kumar. "A feature reduced intrusion detection system using ANN classifier." Expert Systems with Applications 88 (2017): 249-257. http:/www.sciencedirect.com/science/article/S0957417417304748.(online).

[11] Hoa Dinh Nguyen and Qi Cheng, “An Efficient Feature Selection Method for Distributed Cyber Attack Detection and Classification”, IEEE Transactions, 978-1-4244-9848-2/11 February 2011.

[12] Hettich, Seth, and S. D. Bay. "The UCI KDD Archive [http://kdd. ics. uci. edu]. Irvine, CA: University of California." Department of Information and Computer Science 152 (1999).

[13] Mukherjee, Saurabh, and Neelam Sharma. "Intrusion detection using naive Bayes classifier with feature reduction." Procedia Technology 4 (2012): 119-128.

[14] Gu, Guofei, Prahlad Fogla, David Dagon, Wenke Lee, and Boris Skorić. "Measuring intrusion detection capability: an information-theoretic approach." In Proceedings of the 2006 ACM Symposium on Information, computer and communications security, pp. 90-101. ACM, 2006.

[15] Sahu, Shailendra, and Babu M. Mehtre. "Network intrusion detection system using J48 Decision Tree." In Advances in Computing, Communications and Informatics (ICACCI), 2015 International Conference on, pp. 2023-2026. IEEE, 2015.

[16] Sabhnani, Maheshkumar, and Gürsel Serpen. "Application of Machine Learning Algorithms to KDD Intrusion Detection Dataset within Misuse Detection Context." In MLMTA, pp. 209-215. 2003.

[17] Al-Jarrah, O. Y., A. Siddiqui, M. Elsalamouny, Paul D. Yoo, Sami Muhaidat, and Kwangjo Kim. "Machine-learning-based feature selection techniques for large-scale network intrusion detection." In Distributed Computing Systems Workshops (ICDCSW), 2014 IEEE 34th International Conference on, pp. 177-181. IEEE, 2014.

[18] Hall, Mark, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. "The WEKA data mining software: an update." ACM SIGKDD explorations newsletter 11, no. 1 (2009): 10-18.