A Decision Tree Based Intrusion Detection System for Identification of Malicious Web Attacks

Ratul Chowdhury[1], Pallabi Banerjee[2], Soumya Deep Dey[3], Banani Saha[4], and Samir Kumar Bandyopadhyay[5]

1 Department of Computer Science and Engineering, Future Institute Of Engineering and Management, Kolkata, India, ratul.chowdhury@teamfuture.in
2 Department of Computer Science and Engineering, Future Institute Of Engineering and Management, Kolkata, India, pallabibnj@gmail.com
3 Department of Computer Science and Engineering, Future Institute Of Engineering and Management, Kolkata, India, soumyadeepdey48@gmail.com,
4 Department of Computer Science and Engineering, University Of Calcutta, Kolkata, India, bsaha_29 @yahoo.com
5 Department of Computer Science and Engineering, University Of Calcutta, Kolkata, India, 1954 samir@gmail.com

Abstract. In today's world, cyber attack is one of the major issues concerning the organizations that deal with technologies like cloud computing, big data, IoT etc. In the area of cyber security, intrusion detection system (IDS) plays a crucial role to identify suspicious activities in the network traffic. Over the past few years, a lot of research has been done in this area but in the current scenario, network attacks are diversifying in both volume and variety. In this regard, this research article proposes a novel IDS where a combination of information gain and decision tree algorithm has been used for the purpose of dimension reduction and classification. For experimental purpose the NSL-KDD dataset has been used. Initially out of 41 features present in the dataset only 5 high information gain valued features are selected for classification purpose. The applicability of the selected features are evaluated through various machine learning based algorithms. The experimental result shows that the decision tree based algorithm records highest recognition accuracy among all the classifiers. Based on the initial classification result a novel methodology based on decision tree has been further developed which is capable of identifying multiple attacks by analyzing the packets of various transactions in real time.

Keywords: Intrusion Detection System· NSL-KDD Dataset· One Hot Encoding· Information Gain· Decision Tree

1 Introduction

Internet Information resources are actively growing, penetrating many spheres of social life. With the increasing dependence of human over the Internet technology and World Wide Web, it becomes challenging to secure the information and confidential data flowing over the network. Valuable information attracts more attackers and is always prone to a huge number of attacks over the network [7]. The rapid use of internet creates some serious issues:

- Man in the Middle Attacks: This type of attack occurs when an attacker gets access over a two-party transaction. This type of cyber security threats are made by cyber-criminals who set up fake public Wi-Fi networks or install malware on a victim's network.
- Internet of Things Insecurity: Internet of Things connects network devices across the world. Weak or guessable password, insecure network services and ecosystem interfaces, insecure data transfer and storage are the major threats for IoT.

- Changing the Signature of Attacks: In today's world, the nature of the attacks are changing day by day with the increase in data and these new signature based attacks are really difficult to detect [13].

In the area of cyber security, the role of IDSs are to analyze all the upcoming and outgoing packets of a system or a network and restrict unauthorized access. In the past few decades, researchers have adopted various machine learning and deep learning platforms for the construction of intelligent IDSs but due to the divergence nature of various attacks it is always challenging to select appropriate features with minimum detection time. The learning techniques of IDSs are broadly classified into three categories: supervised, unsupervised and semi supervised learning[1]. In supervised learning all the instances in the dataset have a specific label, in unsupervised learning the data has no label, where as semi supervised is a combination of both supervised and unsupervised learning. Based on the nature of the attacks the IDSs are further classified into three types [6] namely signature based, anomaly based and hybrid. Signature based or misuse based IDSs are used to identify known attacks, that means detection of attacks by looking for specific patterns, such as byte sequences in the network traffic. As a result it is unable to detect zero day attacks .Anomaly based IDSs are used to detect various unknown attacks. It creates the model according to the system behavior and identifies anomalies which deviates from normal behavior. Increasing false alarm rate is the major issue of anomaly based IDS. Hybrid IDS is a combination of both signature and anomaly based IDS. Snort is one of the popular and freely available IDS used in UNIX or Linux operating system. So the main contribution of this research work is:

- Selection of a more upright feature set containing only 5 high gain valued features using information gain.
- Exhaustive investigation on the applicability of the derived features is carried out through various machine learning frameworks. This establishes the feasibility and part superiority of the proposed methodology.
- In tune with the previous contributions, a novel methodology based on decision tree is further proposed to identify various attacks in real time.

So, the rest of the paper is structured as follows, section 2 reflects the literature survey portion, section 3 describes the proposed methodology, section 4 analyzes the results along with various comparisons and the conclusion is presented in section 5.

2 Literature Survey

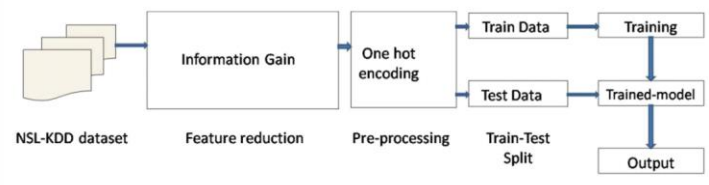
Malicious web attack poses various significant threats for our network and becomes a challenging issue. As discussed earlier, current protection techniques are potentially designed based on three types of mechanism: signature based, anomaly based and hybrid. Ioulianou et al [10]. proposed a signature based IDS for IOT networks which involved both centralized and distributed IDS modules using Cooja simulator. A Denial of Service (DOS) attack scenario was created on IOT devices. They concluded that it supports application development for Contiki OS, but did not import the IDS modules to Contiki OS to test its performance in real world IOT environment. In another work Almutairi et al. [3] introduced a solution in parallel processing environment with the help of most frequent features and an updating agent. They proved that this module can be used for both host and network based IDS. It is a challenging task to identify new signature based attacks which are not frequent enough. Shah et al [19]. implemented a signature based IDS using SNORT and WTNPCAP in windows environment. Using this powerful software SNORT, real time traffic analysis has been performed and packet logging being carried out. In this work, there is a scope of improvement in processing time. In [16], the authors proposed an IDS using several Decisions trees and decision Rules. The prediction accuracy of classifiers had been evaluated using 10-fold cross validation. They have extracted 65534 instances from NSL-KDD dataset for their experiment and all the 41 features

of the dataset were used for classification which are very much expensive to calculate. A combined Multi-Layer Perceptron (MLP) and Decision tree based technique were introduced by Esmaily et al. [9]. All the 41 features of KDDCup99 dataset were used for experimental purpose that not only generated a significant low false alarm value but were also difficult to implement in real time traffic analysis. In another interesting work Rai et al. [18] proposed a new algorithm based on Decision Tree Split method and made a comparative study with some existing classifiers like C4.5, CART etc. Depending on information gain values, the top 16 features were selected out of 41 features of NSL-KDD dataset that estimated a low yielding accuracy of 79%. Bajaj et al. [5] showed a completely different approach where several feature reduction techniques like Information Gain Attribute Evaluation, Gain Ratio Attribute, and Correlation Attribute Evaluation were used. They showed the applicability of various feature reduction algorithms with the help of different classifiers like Na¨ıve Bayes, SVM, MLP etc , but the accuracy level was not up to the mark. In [17] the authors used Principal Component Analysis (PCA) algorithm for dimension reduction. After dimension reduction the 16 relevant features of KDD99 dataset were further passed into Na¨ıve Bayes classifier for prediction but the experiment was conducted with a few number of instances. Lakhina et al. [14] used PCA along with various machine learning tools for Effective Anomaly–Based Intrusion Detection. After applying PCA the 8 foremost features of NSL-KDD dataset were fed through various machine learning frameworks and the model outperforms. In [11] the authors have built a model with the help of Support Vector Machine (SVM). The top 30 features of NSL-KDD dataset were selected using information gain technique which were further fed through SVM for classification. Their proposed model gives more than 90% accuracy. In [4] the authors have used a probabilistic approach for analyzing intrusion in the network traffic .This approach deploys first order Markov chain process to predict the anomalous activities in the network. The total procedure was carried out in three steps such as defining the states, building the state transition matrix and probability distribution of event occurrence. The FOMC model performance was evaluated through KDDCup99 dataset and their accuracy was up to the mark. Shanmugavadivu et al. [20] introduced a system based on fuzzy logic to detect an intrusion detection behavior within a network. They have used automated strategy for generation of fuzzy rules, which are obtained from the definite rules using frequent items. KDDCup99 dataset was taken for the experiment and after dimension reduction the model gives approximately 90% detection rate.

On surveying and analyzing the above works, it is concluded that though various research attempts have been done in this area, there is still a scope for improvement. Since dimension reduction and proper feature selection place a pivotal role for real time traffic analysis, so improvement can be done in feature reduction and enhancing recognition accuracy.

3 Proposed Methodology

In today's world safety and privacy of data is a major issue and IDSs provide a first level defense for our system and network .Various rule based procedure or machine learning frameworks are used for the construction of intelligent IDS . This work uses a decision tree based classifier for detecting various malicious attacks. As illustrated in Figure 1 the proposed methodology consists of 4 primary sections namely i) Dataset Selection ii) Dimension Reduction iii) Preprocessing and iv) Classification. .



Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 9 July 2020 doi:10.20944/preprints202007.0191.v1

Fig.1. Detailed Decision tree based model

This paper uses the NSL-KDD dataset which not only contains 125973 number of records but also contains 22 different types of modern attack. In the next phase, out of 41 features only top 5 relevant features have been selected using information gain technique. Finally after preprocessing, the selected features are evaluated through various machine learning frameworks.The individual segments of the proposed methodology are briefly explained in the following subsections

3.1 Dataset Description

The NSL-KDD dataset [2][21][15][8] was generated in 2009 that has significant advantage over KDD'99 dataset . It reduces the size of the KDD99 dataset by eliminating the redundant and duplicate records from the dataset and hence allowing the complete dataset to be used for experimental purpose. The data files containing in NSL-KDD dataset cover KDDTrain+ for training and KDDTest+ and KDDTest-21 for testing. The number of records in each dataset is shown in table 1.

Table 1. Basic NSL-KDD Dataset

Dataset	Total	Normal	DoS	Probe	R2L	U2R
KDDTrain+	125973	67343	45927	11656	995	52
KDDTest+	22544	9711	7458	2421	2754	200
KDDTest-21	11850	2152	4342	2402	2754	200

Table 2. Detailed feature description of NSL-KDD dataset

Intrinsic	Content	Time Based	Host Based	
1.Duration	10.Hot	23.Count	32. Dst host count	
2. Protocol type	11. Num failed logins		24. Srv count	33. Dst host srv count
3. Service	12. Logged in		25. Serror rate	34. Dst host same srv rate
4.Flag	13. Num comp romised		26. Srv serror rate	35. Dst host diff srv rate
5. Src bytes	14. Root shell		27. Rerror rate	36. Dst host same src port rate
6. Dst bytes	15. Su attempted		28. Srv rerror rate	37. Dst host srv diff host rate
7.Land	16. Num root		29. Same srv rate	38. Dst host serro r rate
8. Wrong fragment	17. Num file creations		30. Diff srv rate	39. Dst host srv s error rate
9. Urgent	18. Num shells		31. Srv diff host rate	40. Dst host rerror rate
	19. Num access files			41. Dst host srv r error rate
	20. Num outbound cmds			
	21. Is hot login			
	22. Is guest login			

The dataset consists of 41 features with one specific class label. The detailed description of the various features are shown in table 2. The feature set is diversified into four categories: features from no.1-9 are called intrinsic features, no.10-22 are called content features, no.23-.31 are called time based features and features from 32-41 are host based features. The dataset contains total 22 type of attacks which are categorized into four types namely: DOS (denial of service attack), R2L (Root to local attack), U2R (User to Root attack) and Probe (Probing attack).

3.2 Dimension Reduction

Feature ranking and dimension reduction play a pivotal role in any machine learning framework [12]. Information Gain Attribute Evaluation is one type of filter method for feature reduction. It

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 9 July 2020 doi:10.20944/preprints202007.0191.v1

evaluates the worth of an attribute by measuring the information gain with respect to the target attribute and ranks the attributes according to their gained values in descending order. It basically measures how much feature contributes in decreasing the overall entropy .Entropy is defined as:

$$Entropy(X) = -\sum_{i=1}^{n} P_i * \log_2 P_i$$

where Pi is the probability of the i^{th} class in the dataset, and Entropy shows the degree of impurity . The closer to 0 it is, the lesser is the impurity in the dataset.

Information gain attribute evaluation algorithm has been applied on NSL-KDD dataset, which estimated the gain values of all the 41 features in descending order. We have selected the top ranked 5 features to build our model for better accuracy. The feature selection algorithm is described below:

i. n = No. of unique classes in the target attribute of the dataet

ii. k = No. of attributes in the dataset

iii. m = No. of different values in the i^{th} attribute

iv. $P(class\ i)$ = Probability of being class i in the outcome

v. $P(value\ j/class\ x)$ = Probability of being *value j* for a particular attribute when the outcome be *class x*

vi. $P(value\ j)$ = Probability of being *value j* for a particular attribute

vii. $Infogain(target)$ = Information gain of *target* attribute

viii. $Infogain(attribute\ i)$ = Information gain of i^{th} attribute **Algorithm 1**

Calculation of Information Gain

Input: NSL-KDD dataset

Output: Information gain values of each features present in the dataset

1: **for** (i in 1 to n) **do**

2: $Infogain(target) \leftarrow -^{P} P(class\ i) * \log_2(P(class\ i))$

3: **end for**

4: **for** (i in 1 to k) **do**

5: **for** (j in 1 to m) **do**

6: **for** (x in 1 to n) **do**

7: $Temp \leftarrow -^{P} P(value\ j/class\ x) * \log_2(P(value\ j/class\ x))$

8: **end for**

9: $Temp1 \leftarrow -^{P} P(value\ j) * Temp$

10: **end for**

11: $Infogain(attribute\ i) \leftarrow Temp1$

12: $Gain(attribute\ i) \leftarrow Infogain(target) - Infogain(attribute\ i)$

13: **end for**

14: Sort the Gain values for all the attributes in descending order to get the rankings in terms of effectiveness of the outcome.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 9 July 2020 doi:10.20944/preprints202007.0191.v1

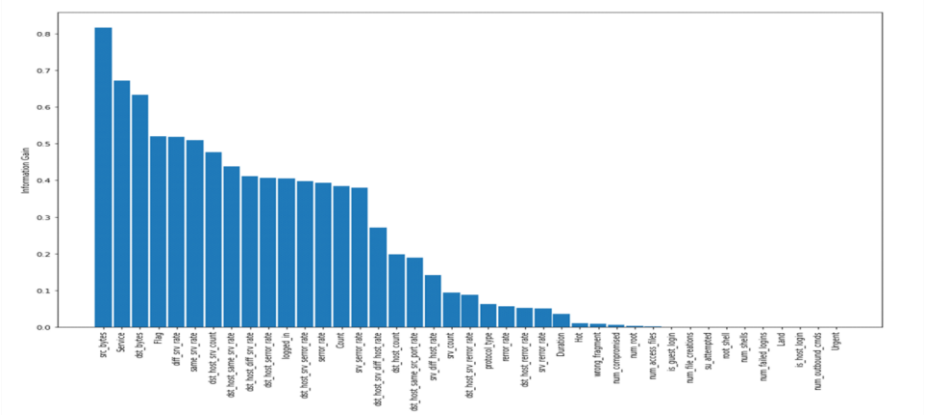


Fig.2. Various features of NSL-KDD dataset along with its information gain values

After applying algorithm 1 the various features of NSL-KDD dataset and their respective information gain values are shown in figure 2. Based on the figure depicted above, the top 5 features that are selected for classification purposes are : src bytes, service, dst bytes, flag and diff srv rate.

3.3 Preprocessing of The Dataset

Among the top 5 selected features mentioned above, service and flag are categorical in nature where as the others are numeric. As the various machine learning models taks only numeric value, we have to convert the non-numeric features into numeric form. The one hot encoding technique has been used to convert the non numeric features into numeric pattern. Categorical variable Flag contains 11 different types of flag values: 'OTH', 'S1', 'S2', 'RSTO', 'RSTRs', 'RSTOS0', 'SF', 'SH', 'REJ', 'S0', 'S3' where the flag value SF can be converted to binary vector (0,0,0,0,0,0,0,0,0,1,0) using one hot encoding method. Similarly the service field which contains 70 different types of service values are also encoded into binary form.

3.4 Methodology For Classification

The top 5 features of NSL-KDD dataset are further fed through decision tree based model for classification purpose. Decision tree [22] is a supervised learning approach where each internal node represents a test on an attribute; each edge denotes an outcome of the test and each terminal nodes holds a class level. It is a simple decision making algorithm does not requires any normalization and scaling procedure. The working principal of decision tree construction is shown in algorithm 2.

Algorithm 2 Decision Tree Construction

Input: Reduced NSL-KDD dataset

Output: A Decision tree using the dataset

1: Start

2: Calculate the Gini index for each attribute using *Gini index* $\leftarrow 1 - (P_i)^2$, where
Where P_i is being the probability of i^{th} class in a given branch

3: Split the dataset using a: Select the root node based on the attributes that has the lowest Gini index b: Determine the splitting criteria c: Partition the dataset into two subsets based on the splitting criteria

4: Repeat step 1 and 2 on each subset until leaf nodes can be found in all the branches of the tree

5: Stop

4 Experimental Setup and Evaluation Matrices

All the experiments have been conducted on Anaconda jupyter IDE platform in python 3.6. 4 environment with Intel(R) core (TM) i3-7020U CPU @ 2.00 GHz configuration and 4GB memory. The result analysis section is divided into two subsections namely binary and multiclass classification. In binary classification, dataset contains two classes: normal and attack. On the other hand in multi class classification, the attack type is further divided into 4 subclasses: DOS, U2L, R2L and Probing. For comparison purpose, the features have been evaluated through various machine learning frameworks. Finally, for real time classification a transaction based procedure along with some result outcome is shown.
We have used 5 important performance indicators to evaluate the performance of our model.

1. Accuracy (AC): It is the percentage ratio of correctly classified instances by the total number of instances.

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

2. Precision: It is the ratio of relevant instances by the retrieved instances.

$$Precision = \frac{TP}{TP + FP}$$

3. Recall: It is the ratio of relevant instances that have been retrieved by the total amount of relevant instances.

$$Recall = \frac{TP}{TP + FN}$$

4. F1 measure: It is the weighted average of precision and recall.

$$F1\ measure = \frac{2 * (Recall * Precision)}{Recall + Precision}$$

5. Support: Support is the true instances that lie in that class where,

- TP (True positive): is the number of positive instances that are corrected classified.
- TN(True negative):is the number of negative instances that are correctly classified.
- FP(False positive):is the number positive instances that are wrongly classified.
- FN(False negative):is the number of negative instances that are wrongly classified.

Hence a good IDS always have a good detection rate and low false alarm rate.

4.1 Result Analysis

In order to perform the result analysis the 5 relevant features of KDD Train+dataset have been used for training purpose and the testing for both binary and multiclass categories were performed through KDDTest+ and KDDTest-21 dataset respectively. The detection outcome of various classifiers along with its precision, recall, F1 measure and support values are shown in the following subsections.

4.1.1 Binary Classification In binary classification, all the 4 type of attacks have been categorized into one group called attack and the rest are defined as normal type. The model has been trained

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 9 July 2020 doi:10.20944/preprints202007.0191.v1

through KDDTrain+ dataset with 4 different machine learning framework Na¨ıve bayes, Logistic regression, Random forest and decision. Table 3 summarizes the corresponding accuracy, precision , recall, F1-measure and support values of KDDTest+ dataset for the above mentioned classifiers. The experimental result shows that the decision tree model records 90.9% accuracy with high precision , recall, F1 measure and support values among all other classifiers.

Table 3. Accuracy, precision, recall, f1-measure, support values for KDDTest+ dataset

Classifier	Accuracy	Precision	Recall	F1-measure	Support
Naive bayes	0.393	0.810	0.005	0.009	22544
Logistic Regression	0.606	0.608	0.992	0.754	22544
Random forest	0.876	0.929	0.863	0.894	22544
Decision tree classifier	0.909	0.932	0.917	0.925	22544

Similarly table 4 describes the corresponding precision, recall, F1-measure and support values for KDDTest-21 dataset. This dataset is very critical and challenging as it contains numerous number of new records that are not present in training set. The table shows that the proposed model records 83.7% accuracy which is highest among all the classifiers.

Table 4. Accuracy, precision, recall, f1 measure, support for KDDTest-21

Classifier	Accuracy	Precision	Recall	F1-measure	Support
Naive bayes	0.152	0.792	0.004	0.008	11850
Random forest	0.803	0.914	0.848	0.880	11850
Logistic regression	0.826	0.849	0.968	0.905	11850
Decision tree	0.837	0.917	0.889	0.903	11850

4.1.2 Multiclass Classification Initially, in binary classification, the records of the NSL-KDD dataset were divided into two groups: Normal and Attack. In multiclass classification, the attack types are further classified into 4 sub categories : DOS, U2R, R2L and Probing. Table 5 depicts the accuracy, precision, recall, f1-measure and support values for KDDTest+ dataset. The model records pr´ecised accuracy of 85.4% as compared to other classifiers.

Table 5. Multiclass classification accuracy, precision, recall, f1 measure, support values for KDDTest+ dataset

Classifier	Accuracy	Precision	Recall	F1-measure	Support
Naive bayes	0.370	0.389	0.370	0.278	22544
Logistic Regression	0.597	0.517	0.597	0.509	22544
Random forest	0.838	0.838	0.838	0.838	22544
Decision tree	0.854	0.851	0.854	0.849	22544

Table 6 also shows the accuracy, precision, recall, f1 measure, support values for KDDTest-21 dataset. Here also the decision tree based methods outperform all the other classifiers with an accuracy of 74.4 %.

Table 6. Multiclass classification accuracy, precision, recall, f1 measure, support values for KDDTest-21 dataset

Classifier	Accuracy	Precision	Recall	F1-measure	Support
Logistic Regression	0.254	0.261	0.253	0.171	11850
Naive bayes	0.308	0.172	0.308	0.186	11850
Random forest	0.675	0.731	0.675	0.682	11850
Decision tree classifier	0.744	0.776	0.744	0.751	11850

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 9 July 2020 doi:10.20944/preprints202007.0191.v1

Table 7 shows the model construction time for all the mentioned classifiers. According to the table, it is shown that a proper feature reduction minimizes the model buildup time to a particular level. Though the Logistic Regression and Na¨ıve Bayes classifier take less time but there is a drastic reduction when the accuracy of the model is concerned. The result also implies that there is an optimal trade off between model build up time and detection rate which is achieved by decision tree method. In the next part, the detection rates for individual attack types have been estimated.

Table 7. Model construction time for binary and multiclass classification

Classifier	**Time Taken (in ms) for 5 features Binary**	**Time Taken (in ms) for 41 features Binary**	**Time Taken (in ms) for 5 features Multiclass**	**Time Taken (in ms) for 41 features Multiclass**
Decision Tree	1750	11800	2400	6500
Random forest	20400	31900	28700	60000
Logistic Regression	472	8750	11400	89000
Na¨ıve Bayes	558	794	520	810

Table 8 shows the respective count for all the 4 types of attack along with its accuracy, precision, recall, f1 measure and support values for KDDTest+ and KDDTest-21 dataset. The table also shows that in average the model records above 90% detection rate for all of the above mentioned attacks categories.

Table 8. Accuracy, precision, recall, f1 measure, support for KDDTest+ and KDDTest-21 for individual attack type

Classifier	**Dataset Used**	**Class**	**Accuracy**	**Precision**	**Recall**	F1-measure	Support
Decision tree classifier	KDDTest+	Normal	0.879	0.83	0.91	0.87	9711
		DOS	0.967	0.95	0.95	0.95	7458
		U2R	0.983	0.18	0.26	0.21	200
		R2L	0.903	0.67	0.41	0.51	2754
		Probing	0.975	0.88	0.89	0.89	2421
Decision tree classifier	KDDTest21+	Normal	0.776	0.42	0.62	0.50	2152
		DOS	0.952	0.95	0.92	0.93	4342
		U2R	0.962	0.18	0.26	0.21	200
		R2L	0.834	0.71	0.48	0.57	2754
		Probing	0.957	0.90	0.89	0.89	2402

According to the result analysis section the proposed model creates a complete tradeoff between the model construction time and detection rate. In order to justify the aforementioned statement we have prepared 10 transactions by taking random data from both of the test dataset which contains 100 packets each. Random selection helps to maintain the generic nature of our proposed technique. Instead of building a model at run time, few coefficient values have been obtained by using algorithm 2 in model buildup time that are further incorporated to estimate the detection rate of randomly selected sample. Figure 3 depicts the transaction time verses accuracy graph where the X-axis defines the various transactions and the red and green line along with Y-axis shows the corresponding transaction time and accuracy. According to the graph all the transactions have been performed within a time period between 0 to 18 milliseconds with a high detection rate.

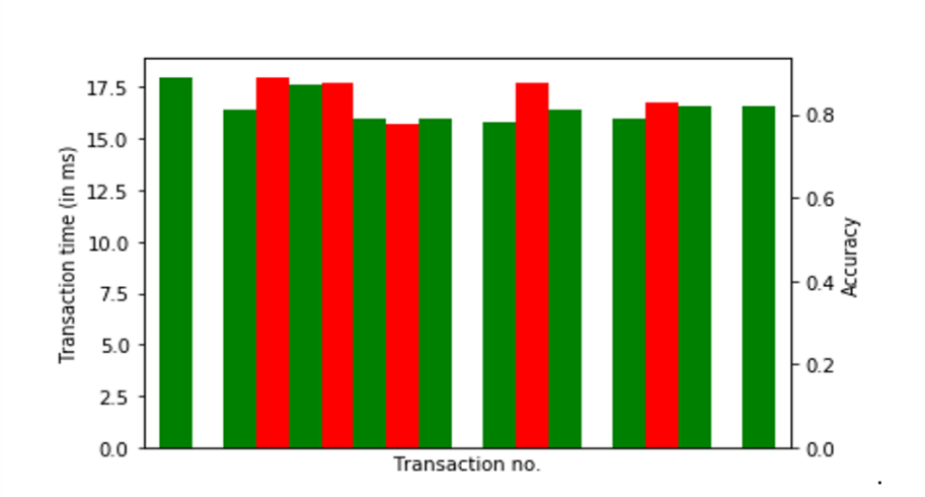


Fig.3. Accuracy verses Transaction time considering 100 packets/transaction

5 Conclusion

In this work, we have proposed a complete IDS based on the combination of information gain and decision tree based algorithm. The proposed method not only reduces the features up to a particular significant level, but also produces a high detection rate in both binary and multiclass categories. The main advantages of the aforementioned technique are, it creates a complete tradeoff between the detection rate and model construction time. It uses only 5 relevant features instead of 41 features which not only reduces the complexity of the model but also reduces the detection time This work also tried to incorporate most recent works that have been done in the current scenario and shows the efficiency of our model with respect to detection rate, precision, recall, F-measure and support. In future, we can effectively improve the detection rate and model construction time by introducing the concept of various deep learning and ensemble learning method.

References

1. Aburomman, A.A., Reaz, M.B.I.: Survey of learning methods in intrusion detection systems. In: 2016 international conference on advances in electrical, electronic and systems engineering (ICAEES). pp. 362–365. IEEE (2016)
2. Aggarwal, P., Sharma, S.K.: Analysis of kdd dataset attributes-class wise for intrusion detection. Procedia Computer Science **57**, 842–851 (2015)
3. Almutairi, A.H., Abdelmajeed, N.T.: Innovative signature based intrusion detection system: Parallel processing and minimized database. In: 2017 International Conference on the Frontiers and Advances in Data Science (FADS). pp. 114–119. IEEE (2017)
4. Aneetha, A., Bose, S.: Probabilistic approach for intrusion detection systemfomc technique. In: 2014 Sixth International Conference on Advanced Computing (ICoAC). pp. 178–183. IEEE (2014)
5. Bajaj, K., Arora, A.: Improving the intrusion detection using discriminative machine learning approach and improve the time complexity by data mining feature selection methods. International Journal of Computer Applications **76**(1), 5 – 11 (2013)
6. Buczak, A.L., Guven, E.: A survey of data mining and machine learning methods for cyber security intrusion detection. IEEE Communications surveys & tutorials **18**(2), 1153–1176 (2015)
7. Chen, C.M., Chen, Y.L., Lin, H.C.: An efficient network intrusion detection. Computer communications **33**(4), 477–484 (2010)

8. Dhanabal, L., Shantharajah, S.: A study on nsl-kdd dataset for intrusion detection system based on classification algorithms. International Journal of Advanced Research in Computer and Communication Engineering **4**(6), 446–452 (2015)
9. Esmaily, J., Moradinezhad, R., Ghasemi, J.: Intrusion detection system based on multi-layer perceptron neural networks and decision tree. In: 2015 7th Conference on Information and Knowledge Technology (IKT). pp. 1–5. IEEE (2015)
10. Ioulianou, P., Vasilakis, V., Moscholios, I., Logothetis, M.: A signature-based intrusion detection system for the internet of things. Information and Communication

    Technology Form (2018)
11. Jha, J., Ragha, L.: Intrusion detection system using support vector machine. International Journal of Applied Information Systems (IJAIS) **3**, 25–30 (2013)
12. Khalid, S., Khalil, T., Nasreen, S.: A survey of feature selection and feature extraction techniques in machine learning. In: 2014 Science and Information Conference.

    pp. 372–378. IEEE (2014)
13. Kim, Y., Kim, I., Park, N.: Analysis of cyber attacks and security intelligence. In: Mobile, Ubiquitous, and Intelligent Computing, pp. 489–494. Springer (2014)
14. Lakhina, S., Joseph, S., Verma, B.: Feature reduction using principal component analysis for effective anomaly–based intrusion detection on nsl-kdd (2010)
15. Meena, G., Choudhary, R.R.: A review paper on ids classification using kdd 99 and nsl kdd dataset in weka. In: 2017 International Conference on Computer, Communications and Electronics (Comptelix). pp. 553–558. IEEE (2017)
16. MeeraGandhi, G., Appavoo, K., Srivasta, S.: Effective network intrusion detection using classifiers decision trees and decision rules. Int. J. Advanced network and application, Vol2 (2010)
17. Neethu, B.: Classification of intrusion detection dataset using machine learning approaches. International Journal of Electronics and Computer Science Engineering **1**(3), 1044–1051 (2012)
18. Rai, K., Devi, M.S., Guleria, A.: Decision tree based algorithm for intrusion detection. International Journal of Advanced Networking and Applications **7**(4), 2828 (2016)
19. Shah, S.N., Singh, M.P.: Signature-based network intrusion detection system using snort and winpcap. International Journal of Engineering Research & Technology (IJERT) **1**(10), 1–7 (2012)
20. Shanmugavadivu, R., Nagarajan, N.: Network intrusion detection system using fuzzy logic. Indian Journal of Computer Science and Engineering (IJCSE) **2**(1) , 101–111 (2011)
21. Tavallaee, M., Bagheri, E., Lu, W., Ghorbani, A.A.: A detailed analysis of the kdd cup 99 data set. In: 2009 IEEE symposium on computational intelligence for security and defense applications. pp. 1–6. IEEE (2009)
22. Thenmozhi, K., Deepika, P.: Heart disease prediction using classification with different decision tree techniques. International Journal of Engineering Research and

    General Science **2**(6), 6–11 (2014)