Cluster Computing (2020) 23:1397–1418
https://doi.org/10.1007/s10586-019-03008-x

An integrated rule based intrusion detection system: analysis on UNSW-NB15 data set and the real time online dataset

Vikash Kumar[1] · Ditipriya Sinha[1] · Ayan Kumar Das[2] · Subhash Chandra Pandey[2] · Radha Tamal Goswami[3]

Received: 15 August 2018 / Revised: 21 September 2019 / Accepted: 21 October 2019 / Published online: 29 October 2019

Abstract

Intrusion detection system (IDS) has been developed to protect the resources in the network from different types of threats. Existing IDS methods can be classified as either anomaly based or misuse (signature) based or sometimes combination of both. This paper proposes a novel misuse based intrusion detection system to detect five categories such as: Exploit, DOS, Probe, Generic and Normal in a network. Further, most of the related works on IDS are based on KDD99 or NSL-KDD 99 data set. These data sets are considered obsolete to detect recent types of attacks and have no significance. In this paper UNSW-NB15 data set is considered as the offline dataset to design own integrated classification based model for detecting malicious activities in the network. Performance of the proposed integrated classification based model is considerably high compared to other existing decision tree based models to detect these five categories. Moreover, this paper generates its own real time data set at NIT Patna CSE lab (RTNITP18) which acts as the working example of proposed intrusion detection model. This RTNITP18 dataset is considered as a test data set to evaluate the performance of the proposed intrusion detection model. The performance analysis of the proposed model with UNSW-NB15 (benchmark data set) and real time data set (RTNITP18) shows higher accuracy, attack detection rate, mean F-measure, average accuracy, attack accuracy, and false alarm rate in comparison to other existing approaches. Proposed IDS model acts as the dog watcher to detect different types of threat in the network.

Keywords Intrusion detection system · Signature based · Attack detection rate · False alarm rate · Integrated rule based model

1 Introduction

Internet has become the most essential tool in this modern era. Applications like Local Area Network (LAN), Wide Area Network (WAN), Wireless Local Area Network (WLAN) etc., made computer networking attractive for different enterprises, security services, health care and other emergency services. Internet plays an inevitable role in our daily life. It is obvious that attackers may take advantage of our dependency on internet and threaten it with security threats like botnets based DDoS [4] attacks, Virus, Trojan, Worm, Spyware etc. These types of attacks in the internet are becoming more sophisticated and the number of attacks is also increasing day by day. Several models and mechanisms [12, 21] have already been proposed to defend against different attacks. However, now-a-days providing security to every environment in the network is a big challenging issue. Indeed, intrusion detection

✉ Ditipriya Sinha
ditipriyasinha87@gmail.com

Vikash Kumar
vika96snz@gmail.com

Ayan Kumar Das
das.ayan@bitmesra.ac.in

Subhash Chandra Pandey
s.pandey@bitmesra.ac.in

Radha Tamal Goswami
tamal.goswami@gmail.com

[1] Department of Computer Science and Engineering, National Institute of Technology Patna, Patna 800005, India

[2] Department of Computer Science and Engineering, Birla Institute of Technology, Mesra 800014, India

[3] Department of Computer Science and Engineering, Techno India College of Technology, Newtown, Kolkata 700156, India

system (IDS) has been developed to defend against the threats. It can be considered as a set of techniques and methods that are used to detect suspicious activities in the network. Existing IDS methods.

This paper proposes a misuse based IDS which detects five categories such as: Exploit, DOS, Probe, Generic and Normal. Though firewall can provide good control to access the resources in the internet yet attackers have developed various techniques to bypass it. The proposed system is based on misuse-based technique, which permits it to act as a firewall with some extra information added to it. Thus the system is not limited to the functionality of IDS. The advantage of proposed IDS is that it assists the network administrator to classify the traffic captured into five categories including normal category, which causes lower false alarm rate (FAR) than anomaly based IDS.

The most of the research on IDS are based on KDD99 or NSL-KDD [2, 3, 8, 16, 18, 20, 22, 28] data set. These data sets are considered obsolete to detect recent types of attacks. In the proposed work, UNSW-NB15 [23] is treated as the offline data set to design the proposed IDS model. Moustafa et al. [24] have applied UNSW-NB15 and KDD99 data set for intrusion detection and compared the performance on both the data sets. It shows that IDS applying UNSW-NB15 dataset covers recent attacks compared to KDD99 data set. This paper also proves that KDD99 dataset has no significance now a day. Their work suggests that the decision tree model has best performance on the introduced data set. However, performance evaluation of [24] using UNSW-NB15 shows that intrusion detection rate of [24] is not high due to over lapping nature of several attacks. In [26], authors have presented decision tree approach in combination with genetic algorithm to design a misuse based IDS, where three different datasets KDDCup'99, NSL-KDD and UNSW-NB15 are used to design the models separately and subsequently tested those models using the respective testing set. In this paper, we propose our own integrated model whose accuracy is higher compared to ENADS [24] and Dendron [26]. The performance of this proposed integrated model is also evaluated on the real time data set which is generated in from NIT Patna lab. Sangkatsanee et al. [28] proposed a real time IDS using machine learning techniques on KDD99 data set as offline dataset and real time data set RLD09. In [28], authors considered KDD99 as offline dataset which is obsolete now a day and has significant biasness and also created its own real time dataset which was evaluated on existing decision tree based model considering three categories (Normal, Probe, DoS) only. In our proposed work, we have also designed the real time dataset RTNITP18 which is evaluated on our own proposed integrated model to detect five categories (Normal, Probe, DoS, Exploit, Generic). Forty nodes have been chosen for this purpose at the organization NIT Patna. All of these nodes are connected in a LAN and among them some are attackers and some are victims. We have observed the packet flows in that network for 7 days, from Monday to Sunday and captured data packets from it applying Wireshark packet sniffer tool [5]. Features are extracted from the captured data packets and we export the captured pnpcap file into.txt. Then we use python script to extract features and mapping those features to traditional data set features and save it to.csv file. Thus we are able to design a real time data set. We evaluate the performance of proposed model on real time data set and it is observed that accuracy of our proposed model is 85.8%. It is also observed that value of several evaluation metrics (e.g. ADR = 90.32, FAR = 2.01% etc.) have higher performance on proposed integrated model compared to other existing [24, 26] models. Hence the novelty of the proposed IDS is the development of a model on the basis of recent dataset (UNSW-NB15) which has highly complex behaviour of attacks compared to that of old one (KDD99, NSL-KDDD etc.). The proposed model is created on the basis of different decision tree models by considering rules with high confidence factor which in turn reduces the FAR of the proposed model. In addition, the proposed model is able to detect five categories of attack including normal category with high detection rate (ADR) and low FAR as compared to other recent approaches. The model is also evaluated on the real-time dataset generated by setting up virtual environment at NIT Patna CSE lab.

The proposed model can successfully be used in different domains of industrial applications. Industrial control systems (ICS) are widely used in different domains and it entails real-time data acquisition and system monitoring. It also incorporates automatic control and management of industrial processes. However, ICS is an attractive target for hackers and thus the security issue of ICS is of the paramount importance. The proposed IDS is designed for the automatic detection of malicious attacks. The proposed IDS can collect and analyse different attributes such as the network traffic, security logs. Further, the proposed IDS can also check if there exists security infringement in the system by auditing the data and information from the key points of the computer system. Moreover, evidence collection using digital forensics is an important domain where IDS can substantially be used. The modified version of IDS can be used to notify the administrator by sending an alert as well as it can also activate the digital forensic tool to capture the current state of the system. It is pertinent to mention that this captured system image will include the entire information pertaining to the system at the moment when attack was taking place. And thus these images can be used as evidence in legal proceedings. It is thus obvious that the proposed IDS can successfully be implemented for

maintaining the security of ICS and in the domain of digital forensics. The proposed work can be used to provide security to such system against different threats in the network. It can also be used for any organization where it will be installed on a network device to protect the organization. Any malicious activity found will be reported to security administrator for further action. In an IoT environment the proposed model can also be used to provide security. Hence our proposed IDS acts as the dog watcher for detecting threats in internet.

The key observations of our proposed approach are given below:

1. Most of the related works on IDS are based on KDD cup99 or NSL-KDD [2, 3, 8, 16, 18, 20, 22, 28] data set which is not up to date in the sense that most of recent attacks are not covered. In this paper, we have used a new data set (UNSW-NB15) which covers the most recent attacks compared to KDD cup99 data set.
2. In this paper, an integrated rule based model for IDS has been proposed. The detection rate of this proposed model is high in comparison to other traditional decision tree based model and existing state-of-art works on IDS [24, 26]. Several other metrics (discussed in Sect. 3.5) are also used to evaluate and compare the proposed work with other state-of-art techniques.
3. This paper generates a real-time dataset at NIT Patna CSE lab (RTNITP18) and it acts as a working example to evaluate the performance of the proposed model for real-time environment.
4. Proposed model considers five categories in such a way that some of the other attacks can also be identified, whereas maximum related works on IDS only consider two to three types of attacks in the network. The performance of the proposed model with the UNSW-NB15 (benchmark dataset) and real-time dataset (RTNITP18) shows higher accuracy and ADR in comparison to other existing approaches.
5. The proposed IDS model acts as a compliment to firewall which collect the traffic data incident to the network from the Internet as well as the traffic of the organization and analyses it for any malicious activity.
6. The proposed IDS model acts as out of band device to the network hence it will not create any jitter to the network which is an advantage over the IPS.

Figure 1 shows the working environment of IDS, where data comes from Internet goes to the firewall as well as to the IDS in order to find any malicious activity which is not found by the firewall. The proposed IDS will also monitor the traffic inside the organization for internal intruders.

Rest of the paper is organized as follows. Section 2 describes the related work briefly. Proposed model along with result analysis is discussed in Sect. 3. In Sect. 4, a working example for evaluation of proposed model is described. Finally, the concluding remarks pertaining to the proposed model are given in Sect. 5.

2 Related work

This section preludes the state of art studies on IDS system. Most of the previous works is based on KDD99 [18] data set. KDD99 data set is an old dataset and it does not consider most updated or recent type of attack categories. All the research work on IDS can be divided into two categories: (1) works on IDS in which researchers try to develop system where detection is done based on the provided signature. (2) Works in which a normal profile is generated and any deviation from that profile is reported as attack. This is called behavioural attack also. In first type of category, false reporting is very less, but the system is less prone to the new type of attack whose signature is not known yet. In second category, the system is more prone to false reporting, though a novel attack can be detected.

In [2], a very important model of feature selection has been proposed for IDS which uses Ant Colony Optimization (ACO) concept. In this process, features are treated as nodes in graph representation and edge between nodes represent next choice of feature. The optimal subset of features is selected by traversing the graph. In this model, the pheromone and heuristics are not associated with the links as in the actual ACO traversal; rather it is associated with features itself. Pheromone represents the attractiveness of features. Wang et al. [32] also proposed a function for selecting features using the neighbourhood discernibility matrix. This matrix is used to show the ability of classification for a feature subset. This function helps to determine significance of candidate attribute. Here, dependency function is used to analyse the relevance between features and the decision made applying that feature.

Several recent works on IDS/IPS have been proposed to protect against DoS/DDoS attack. Agarwal et al. [1], have proposed an IDS along with Intrusion Prevention System (IPS) for detecting and recovering from DoS attack in Wi-Fi network. They have proposed the work in IEEE 802.11 standard for Wi-Fi. They have also used Angel of Arrival (AoA) [19] algorithm over RSSI [31] to find the location of attacker. In [25], the authors have proposed an enhanced Confidence Based Filtering (CBF) method to protect cloud services from the DDoS attack by using the concept of correlation pattern. Gupta et al. [13] have proposed a Flow and Volume based approach to detect DDoS attack in an ISP domain and the simulation is provided by network simulator (NS-2) which shows good detection with low

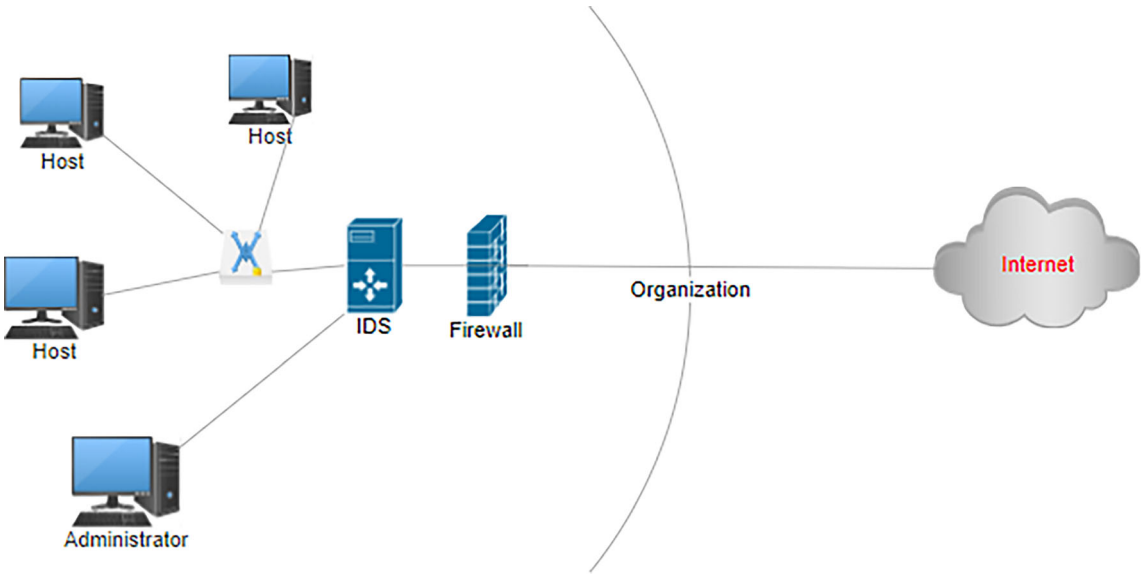

Fig. 1 IDS implemented in an organization connecting to Internet

false alarm rate. Gou et al. [11] have proposed a framework for IDS based on Petri network that consists of two different functions for attacks detection and the model up gradation.

The authors of [20] have proposed an IDS system using the concept of fuzzy set theory with the combination of association of rule mining and genetic network programing (GNP) using KDD99 dataset. The concept of sub-attribute utilization is used for extracting discrete and continuous attributes in order to avoid data loss and gives effective rule mining using GNP. Wattanapongsakorn et al. [33] have proposed an Intrusion Detection and Prevention System (IDPS) in which they covered four categories (Normal, DoS, Probe, Worm). Fuzzy Genetic Algorithm is used for unknown attacks and for known attacks with several machine learning techniques (C4.5 Decision Tree, Random Forest, Ripple Rule, Bayesian Network, and Back Propagation Neural Network). In [37], authors have introduced intuitionistic fuzzy rough graph which is used to handle uncertainty and incomplete information in information system and also proposed an algorithm that can efficiently solve decision making problems.

Das et al. [7] proposed an NIDS model which detects the port scan attack using machine learning concept of SVM. They trained their model using the pattern of frequency change in normal and attack packet. Data is captured every 4 s by the NIDS for analysis. They have used Rough set method as an optimal feature selection method over PCA and only one type of attack has been considered. An IDS model is proposed in [16] by applying SVM on the data set NSL-KDD [27]. They have presented a framework which selects the features of NSL-KDD data to characterize normal traffic more accurately from those of abnormal traffic. Framework uses the method of filter and wrapper for feature selection and ranked those selected feature using information gain ratio. Chowdhury et al. [6], have proposed combination of two machine learning algorithm for classification of anomaly based intrusion. This paper is applying simulated annealing that generate random set of 3 features for each time and then SVM is applied on the selected set to detect the anomalous behaviour. The algorithm has used the dataset from Australian centre for cyber security [23].

Fares et al. [8] proposed an NIDS model using the concept of Neural Network (NN) which is divided in 3 phases. They reduced the dataset applying some pre-processing phase so that over fit due to dominating attack categories do not occur. The attack considered by their model includes Normal, Dos, Probe, R2L, U2R. They compared the performance of their proposed model using both the data sets (10% of KDD99 and reduced data set). The model was trained and tested only on offline data set. In [3], IDS using NN with Genetic algorithm is proposed to improve the accuracy of proposed model have using KDD99 as benchmark dataset. They used NN with resilient back propagation with sigmoid function. In [30], authors have proposed a light weight IDS for anomaly detection using KDD99 by focusing on three major fields such as: 1. Removing redundant data from data set, 2. Feature extraction, and 3. Realization of proposed IDS. The IDS is proposed using a wrapper approach for feature selection. Bagging approach is used to generate multiple training data set which is then used to train multiple neural networks and using the output of these NN new training data set is generated by replacing the class label of original data set

with the output labels. Newly generated data set is then used with C4.5 model.

Many researchers are working to design efficient IDS by using machine learning techniques. In [28], the authors have proposed real-time IDS in which they worked on three categories of attacks such as- Normal, DoS and Probe. The work is divided in three phases: 1) pre-processing phase 2) classification phase and 3) post-processing phase. They have compared the performance with KDD99 and RLD09 [28] on different machine learning techniques and only two types of attacks have been considered. Kalekar et al. [17] proposed a real-time IDS using Naïve Bayes classifier. Their proposed model classifies any packet as normal or abnormal, whereas the performance evaluation of the proposed model is not on any data set. An algorithm for removing outlier from KDD cup99 is proposed in [22]. It makes the algorithm compatible with Weka tool [34]. After executing the proposed algorithm, they have used 10% of the data set and evaluated performance on different machine learning algorithms (Bayesian network, naïve Bayes classifier, J48, J48 Graft, and Random forest). Performance evaluation is done using precision, recall and F-measure parameters [10]. In [14], the authors have proposed a host based IDS using hidden Markova model. Proposed model is evaluated using publically available data set by University of New Mexico (UNM) and Massachusetts Institute of Technology (MIT) Artificial Intelligence laboratory. In this model, training data set is divided into K equal sized sub sequences and using sub sequences which have less correlation between them is used to train sub models. Trained sub models are then merged incrementally in order to design the final model. Moustafa et al. [24], have shown a new data set (UNSW-NB15) for testing any IDS. According to authors, the data set covers most recent attacks. They have described the completeness of the data set and also evaluated the performance on different machine learning algorithms. They also compared the analysis with existing KDD99 data set. Though their work suggests that the decision tree model has best performance on the introduced data set, the accuracy is not very high. In [35], a method for host-based anomaly detection has been presented which uses k-Means clustering technique with ID3 decision tree model. First, k-Means clustering is applied for partitioning the training data which uses Euclidean distance for similarity measurement and then ID3 is applied to each cluster to make decision tree. Results of these two phases are combined using a special algorithm.

Ibrahim et al. [15], have proposed a layered-model approach for NIDS which is divided into two stages. First stage detects data traffic either as normal or abnormal due to some attacks. In second stage, the attacks are classified individually. This model is inspired by airport security model. It uses tcpdump data for analysis using data mining techniques. Sasan and Sharma [29] proposed a hybrid model for IDS using J48 and CART. They have implemented their proposed model in Weka tool and tested the model using NSL-KDD data set for evaluation of performance. Recently Yin et al. [36] have proposed an improved clonal selection algorithm of artificial immune system which is inspired by biological immune system of humans to improve the accuracy of IDS. They have used KDDcup99 for the performance evaluation of their proposed work.

The state of the art describes that most of the works on IDS uses KDD cup 99 and NSL-KDD as the benchmark dataset which do not cover recent attacks and consider as the old dataset. On the other hand, very few papers design their own real time data set to evaluate the intrusion detection rate of their IDS in the network. Intrusion detection rate of the most of the IDS are evaluated on traditional decision tree based model to detect only two to three attacks.

In, our proposed work we design the signature based IDS on a new data set (UNSW-NB15) which covers the most recent attacks compared to KDD cup99 and NSL-KDD. An integrated rule based model for IDS has been designed and it shows higher intrusion detection rate compared to other existing decision tree based models for five categories. We have also designed our own real time data set (RTNITP18) which acts as the testing dataset to evaluate the performance of our proposed integrated model.

3 Proposed work

In the proposed system UNSW-NB15 dataset has been used as benchmark dataset. This section proposes an integrated rule based model to optimize the attack detection rate (ADR) and FAR in the network. The working diagram for the proposed work is shown in Fig. 2 which is having two parts. In first part the IDS model is proposed which starts with analysing the UNSW-NB15 dataset, then preprocessing of data and finally proposing the integrated rule based model and testing it with benchmark test dataset. In second part a working example is considered where a real-time dataset is generated by setting of virtual environment and the performance of the proposed model is evaluated on the dataset.

3.1 Dataset description

The dataset was created [23] by applying IXIA PerfectStorm tool. It [23] includes nine categories of the modern attack types and involves realistic activities of normal

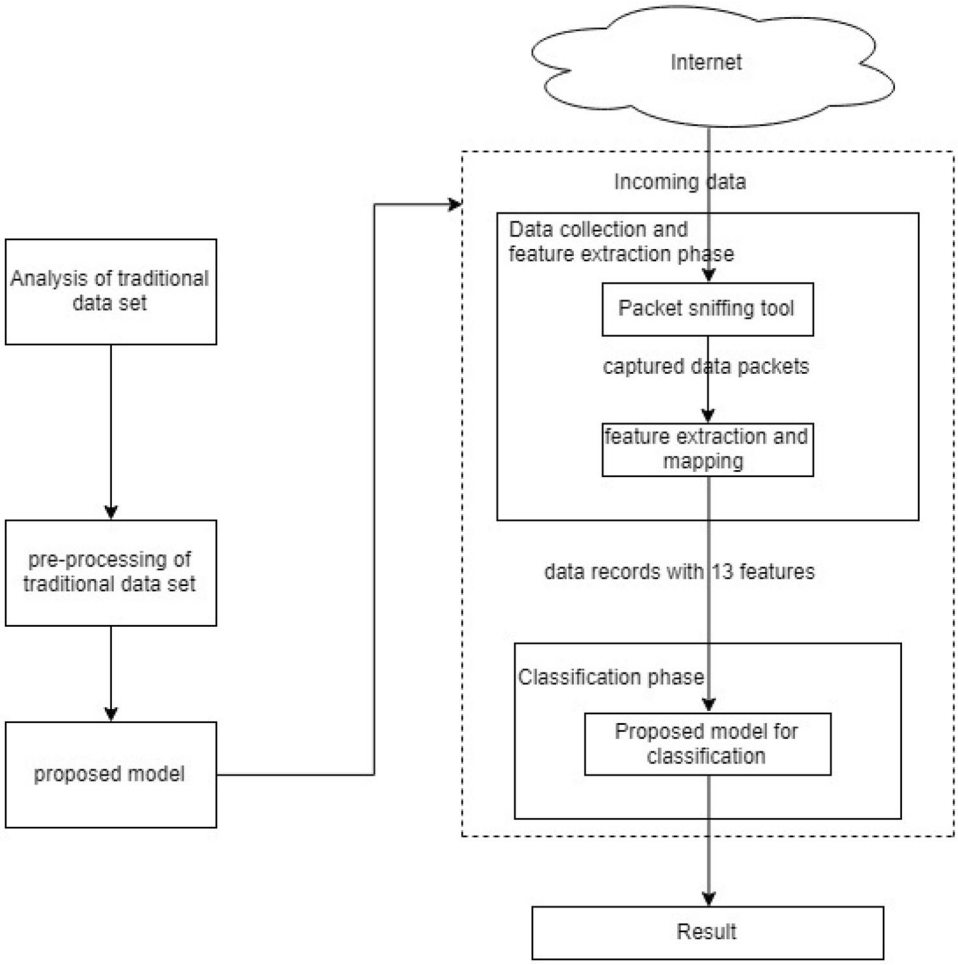

Fig. 2 Flow diagram of proposed work

Table 1 Description of UNSW-NB15 dataset

Number of instances in training set	Number of instances in testing set	Number of attack categories
175,341	82,332	9

traffic. This data set [23] contains 49 features that comprised of several categories. Though there are several datasets available for IDS evaluations like KDD98, NSLKDD, etc., all of these do not cover the latest types of attacks. Recent researches on IDS [9] comment that these datasets do not inclusively reflect the real network traffic behaviour and modern attacks for the recent network threats. Features of UNSW-NB15 are categorized in five ways—(a) Flow features (b) Basic features, (c) Content features, (d) Time features and (e) Additional generated features. Dataset overview is shown in Tables 1 and 2. In Table 3, the definition of attacks is given.

3.2 Pre-processing phase

This phase is divided into following sub-phases:

3.2.1 Dataset reduction

In this phase, the size of original UNSW-NB15 dataset has been reduced by eliminating redundant data from the dataset. Clusters are constructed to detect similar types of data in the dataset. Here, 15 numbers of clusters have been chosen using trial and error method. Silhouette coefficient is used as the measure for determining cluster quality. The number of cluster (i.e. 15) for the dataset analysis is chosen by running the k-mean algorithm for several value of k to generate corresponding cluster configurations. Generated cluster configurations are then tested for the best one using silhouette measurement. The cluster configuration for which silhouette measurement value is maximum for least value of k is selected for the further analysis. It is confirmed from Fig. 3 that if number of clusters is 15, the result is best compared to others. UNSW-NB15 dataset

Table 2 Features of the dataset [23]

#	Features	Description	Category
1	Srcip	Source IP address	Flow features
2	Sport	Source port address	
3	Dstip	Destination IP address	
4	Dsport	Destination port address	
5	Proto	Transaction protocol	
6	State	The state and its dependent protocol, e.g. Acc, clo, else	Basic features
7	Dur	Record total duration	
8	Sbytes	Source to destination bytes	
9	Dbytes	Destination to source bytes	
10	Sstl	Source to destination time to live	
11	Dttl	Destination to source time to live	
12	Sloss	Source packets retransmitted or dropped	
13	Dloss	Destination packets retransmitted or dropped	
14	Service	http, ftp, smtp	
15	Sload	Source bits per second	
16	dload	Destination bits per second	
17	spkts	Source to destination packet count	
18	Dpkts	Destination to source packet count	
19	Swin	Source tcp window advertisement	Content features
20	Dwin	Destination tcp window advertisement	
21	Stcpb	Source tcp sequence number	
22	Dtcpb	Destination tcp sequence number	
23	Smeanz	Mean of the packet size transmitted by the src	
24	Dmeanz	Mean of the packet size transmitted by the dst	
25	Trans_depth	The depth into the connection of http request response transaction	
26	Res_bdy_len	The connection size of the data transferred from the server's http service	
27	Sjit	Source jitter	Time feature
28	Djit	Destination jitter	
29	Stime	Record start time	
30	Ltime	Record last time	
31	Sinpkt	Source interpacket arrival time	
32	Dintpkt	Destination inter packet arrival time	
33	Tcprtt	Tcp connection setup round trip time, sum of synack and acid	
34	Synack	Tcp connection setup round trip time, sum of syn and syn_ack	
35	Ackdat	Tcp connection setup time the time between the syn_ack and ack packets	
36	Is_sm_ips_parts	If srcip (1) equals to dstip (3) and sport (2) equals to dsport (4), this variable assigns to 1 otherwise 0	Additional generated feature
37	Ct_state_ttl	No. for each state (6) according to specific range of values of sttl (10) and dttl (11)	
38	Ct_ftw_http_mthd	No. of flows that has methods such as Get and Post in http service	
39	Is_ftp_login	If the ftp session is accessed by user and password then 1 else	
40	Ct_ftp_cmd	No. of flows that has a command in ftp session	
41	Ct_srv_src	No. of records that contain the same service (14) and srcip (1) in 100 records according to the ltime (30)	
42	Ct_srv_dst	No. of records that contain the same service (14) and dstip (3) in 100 records according to the ltime (30)	
43	Ct_dst_itm	No. of records that contain the same service (14) and dstip (3) in 100 records according to the ltime (30)	
44	Ct_src_itm	No. of records of the srcip (1) in 100 records according to the ltime (30)	

Table 2 continued

#	Features	Description	Category
45	Ct_src_dsport_itm	No of records of the same srcip (1) and the dsport (4) in 100 records according to the ltime (30)	
46	Ct_dst_sport_itm	No of records of the same srcip (1) and the dsport (4) in 100 records according to the ltime (30)	
47	Ct_dst_src_itm	No of records of the same srcip (1) and the dstip (3) in in 100 records according to the ltime (30)	

Table 3 Depicts different type of attacks which act as the class labels in the training dataset

Attack	Description
Analysis	Used to penetrate web applications through emails using spam, web scripts e.g. using HTML files, and port scans.
Backdoor	A technique used to bypass authentication process of system which allow remote access and lead to an unauthorized access to a computer or device, which gives attacker an opportunity to issue commands remotely.
DoS	An attack, through which attacker tries to bring down the services of a computer network (or server) or by making resources unavailable for authorised user requests.
Exploit	Sequence of steps taken by an attacker in order to take advantage of any vulnerability, glitch or bug present in a system or network.
Fuzzers	Activity through which attacker tries to find security vulnerability in system, program, network, or operating system by flooding it with random data in order to crash it.
Generic	It is a type of activity in which an attacker does not bother about the crypto-graphical implementation of any primitives and runs the attack. As an example consider a cipher text with K bit key, in the generic attack of brute force, attacker tries every combination possible using k bits i.e. 2^K combinations and try to decrypt the text.
Probe	Process of gathering information related to computer system or network for evading security controls.
Shellcode	An attacker writes code and inject it to any application which triggers command shell in order to take control of compromised machine
Worm	Attackers try to replicate their functional copies and uses system vulnerability or any social engineering techniques to enter the system.

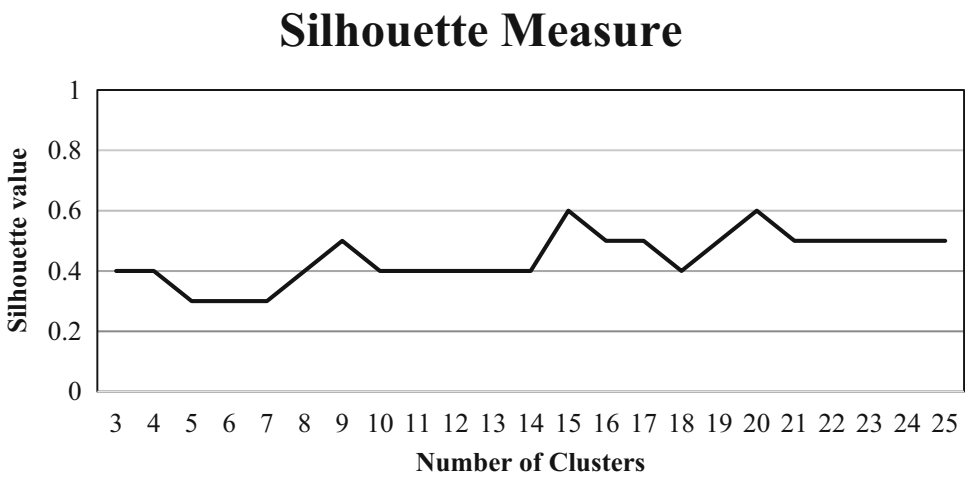

Fig. 3 Silhouette coefficient graph for selecting optimal number of clusters

considers 10 categories (9 attack types and 1 normal). Among them some attacks behave in similar way due to which attacks are overlapped with each other. To solve this, the total dataset is divided into 15 numbers of clusters. Only instances from dominating class have been selected from each class, which results in reduction of data size.

It is found from Table 4 that maximum instances of DoS, Analysis, Backdoor, Fuzzers are in cluster-1 and DoS dominates Analysis and Fuzzers attacks. On the other hand, maximum instances of Exploit and Shellcode attacks are in the same clusters and Exploit dominates Shellcode. Cluster 5 contains with maximum instances of Generic attack. Cluster 2 and 9 contain with maximum instances of Probe

Table 4 Describes similarity between different types of attacks

	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15
Analysis	1436					6			7			551			
Backdoor	1746		5						72			22			
DoS	10,473						1791								
Exploit			1615			5647						8677		17,454	
Fuzzersric	6301		1		4045				1785		15	3680	2306		
Gene	503	1231			37,886					380					
Normal	6633			30,716		4128		166	225			5120			9012
Probe		8788							1703						
Shellcode														1133	
Worm	16	110										4			

Table 5 Reduced dataset

Number of instances in training set	Number of instances in testing set	Number of categories
152,148	74,588	5

Table 6 Describes average information gain value of 47 features

Feature	Information gain
Srcip, Sport, Dstip, Dstport, State, Sload, dload, Swin, Dwin, Stcpb, Dtcpb, Trans_depth, Res_bdy_len, Djit, Stime, Ltime, Sinpkt, Tcprtt, Synack, Ackdat, Is_sm_ips_parts, Ct_ftw_http_mthd, Is_ftp_login, Ct_ftp_cmd, Ct_src_ltm	0
Proto	0.001325
Dur	0.000075
Sbytes	0.31735
Dbytes	0.02955
Sttl	0.19435
Dttl	0.000094
Sloss	0.000075
Dloss	0.004525
Service	0.240125
spkts	0.000075
Dpkts	0.002125
Smeanz	0.000231
Dmeanz	0.0003325
Sjit	0.000075
Dinpkt	0.00925
Ct_state_ttl	0.01805
Ct_srv_src	0.0063
Ct_srv_dst	0.02654
Ct_dst_ltm	0.002025
Ct_src_dport_ltm	0.00215
Ct_dst_sport_ltm	0.0718
Ct_dst_src_ltm	0.020675

attack. Probe dominates worm. Same thing is true for Normal which is completely distinguishable with compared to others. As a result, DoS, Exploit, Generic, Probe, and Normal are dominating on rest of the attacks. Table 5 depicts the detail of our reduced dataset. The proposed model is misuse-based IDS model. It will detect the attacks for which the signature is available. Moreover, some other attacks (other than those present in training set for proposed model) having similar behaviour (for example, Analysis, Backdoor, Fuzzers behave similar to DoS and shellcode behaves similar to Exploit) can also be detected by this model.

3.2.2 Feature reduction

UNSW-NB15 data set contains 47 features. The information gain value of each feature has been computed in order to find the effective set of features for decision making. Information gain value of a feature is defined as its contribution for classifying the data set. If D is the size of a given dataset and A is a feature, then information gain value for feature A is calculated as in Eq. 1.

$$\text{Information gain (A)} = \text{Entropy(D)} - \text{Entropy}_{\text{A}}\text{(D)} \quad (1)$$

where Entropy (D) = Expected information needed to classify a tuple in D and is defined in Eq. 2. $\text{Entropy}_{\text{A}}$ (D) = Extra needed expected information for exact classification when feature A is selected and is defined in Eq. 3.

No. of features	Features	C5			CHAID			CART			QUEST		
		Features Used	Importance	Accuracy	Features Used	Importance	Accuracy	Features Used	Importance	Accuracy	Features Used	Importance	Accuracy
22	sttl spkts sbytes ct_srv_src service proto smean dpkts dinpkt dbytes sjit ct_srv_dst ct_dst_ltm ct_dst_src_ltm ct_dst_sport_ltm dur sloss dmean dloss	dpkt smean dmean ct_srv_dst dur ct_srv_src ct_dst_src_ltm dbytes sbytes sttl	0.0085 0.0109 0.0112 0.0113 0.0119 0.0184 0.0211 0.0382 0.3722 0.4588	89.86	dur ct_dst_ltm ct_srv_src dmean ct_srv_dst ct_state_ttl dttl ct_src_dport_ltm sttl smean	0.0028 0.003 0.0038 0.0134 0.0168 0.0309 0.0358 0.0864 0.0994 0.7015	83.68	sjit dinpkt spkts sloss dur ct_dst_src_ltm dbytes smean sttl ct_dst_sport_ltm	0.0003 0.0003 0.0003 0.0003 0.0003 0.0338 0.0495 0.0934 0.3599 0.4588	82.91	sbytes ct_dst_sport_ltm ct_src_dport_ltm ct_srv_dst ct_dst_src_ltm service	0.009 0.009 0.009 0.009 0.009 0.9548	57.59
13	sttl sbytes ct_srv_src service proto dbytes dinpkt ct_srv_dst ct_dst_ltm ct_dst_src_ltm ct_dst_sport_ltm ct_src_dport_ltm dloss	service dur dloss ct_srv_src ct_srv_dst dinpkt ct_dst_src_ltm dbytes sttl sbytes	0.0042 0.0044 0.0053 0.006 0.0074 0.008 0.0252 0.0713 0.3726 0.4682	89.76	dinpkt service ct_dst_ltm dur proto dbytes ct_dst_src_ltm ct_srv_dst sttl sbytes	0.0013 0.0016 0.0018 0.0021 0.003 0.0033 0.0167 0.0174 0.2009 0.752	81.76	dloss service dur proto dinpkt dbytes ct_dst_src_ltm sbytes sttl ct_dst_sport_ltm	0.0003 0.0003 0.0003 0.0003 0.0003 0.0448 0.0597 0.0618 0.329 0.4949	80.95	sbytes ct_dst_sport_ltm ct_src_dport_ltm ct_srv_dst ct_dst_src_ltm service	0.009 0.009 0.009 0.009 0.009 0.9548	57.59
6	sttl sbytes ct_srv_dst smean ct_dst_sport_ltm dbytes	ct_dst_sport_ltm ct_srv_dst smean dbytes sttl sbytes	0.0044 0.0213 0.0386 0.0503 0.4067 0.4786	84.35	ct_dst_sport_ltm dbytes sbytes ct_srv_dst sttl smean	0.002 0.0055 0.0302 0.691 0.0965 0.7967	76.23	sbytes ct_srv_dst dbytes smean sttl ct_dst_sport_ltm	0.0011 0.249 0.583 0.1179 0.3424 0.4554	75.6	sbytes smean dbytes ct_srv_dst ct_dst_sport_ltm sttl	0.0011 0.0011 0.0011 0.0768 0.4268 0.493	53.79

Fig. 4 Comparative analysis on training data set

$$\text{Entropy}\ (\mathrm{D}) = -\sum_{\mathrm{i}=1}^{\mathrm{m}} \mathrm{p}_i \times \log_2(\mathrm{p}_i) \tag{2}$$

$$\text{Entropy}_{\mathrm{A}}\ (\mathrm{D}) = -\sum_{\mathrm{i}=1}^{\mathrm{v}} \left|\frac{\mathrm{D_i}}{\mathrm{D}}\right| \times \text{Entropy}\ \left(\mathrm{D_j}\right) \tag{3}$$

where feature A has 'v' distinct values and D_j is number of tuples belonging to each distinct feature value of A. p_i can be defined as in Eq. 4.

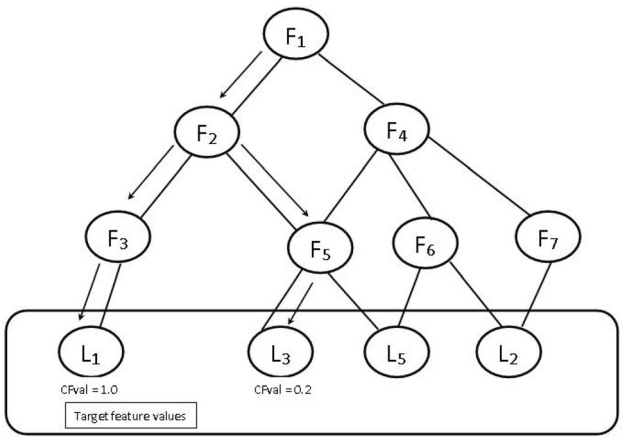

Fig. 5 Rule generation from decision tree

$$p_i = \frac{|c_i|}{|D|} = \text{Probability that an arbitrary tuple in } D \text{ belongs to class } C_i \tag{4}$$

Information gain value is taken averaged over several decision tree models (C5, CHAID, CART, QUEST) as shown in Eq. 5.

$$\text{Information gain}_{\mathrm{AVG}} = \frac{\sum_{\text{for each model}} \text{information gain}}{\text{Total number of model}} \tag{5}$$

Table 6 shows the detailed of information gain. Among 47 features, average information gain value of 25 features in decision tree models are 0 and rest 22 features are greater than 0. We will consider only those 22 features.

3.3 Classification

The dataset contains with 22 numbers of features and 5 numbers of classes. We analyse the performance of IDS on several existing classification models (C5, CHAID, CART, QUEST) and observe the accuracy of each model. Figure 4 describes the performance of each classification model on different number of features i.e. 22, 13 and 6 features respectively. It is shown from Fig. 4 that for 13 features accuracy is marginally decremented compared to 22 numbers of features. Whereas, when the number of features

are 6 the accuracy is significantly less compared to others. Finally, 13 features are considered to design the proposed model. Figure 4 depicts accuracy of the data set on different decision tree based models with different features. Thus, these 13 numbers of features are selected for designing of proposed integrated model. It is observed that for 13 numbers of features, accuracy of C5 is 89.76, CART is 80.95, CHAID is 81.76 and QUEST is 57.79.

3.4 Proposed integrated rule based model

Different decision tree models (C5, CHAID, CART, QUEST) are trained with selected 13 features of the dataset. Rules(R) are derived from different decision tree models in the form of rule (no. of instance; confidence factor). Rules are selected from each model on the basis of their threshold confidence factor.

Figure 5 shows the rule generation process. Leaf nodes represent the target class and the path with confidence factor that satisfies the threshold is used as a rule. For example, according to Fig. 5, path F1–F2–F3–L1 satisfies the threshold value. As a result, the rule is generated in the form "If (F1 and F2 and F3) THEN L1", otherwise the path is rejected.

Confidence factor of a leaf node is defined as the number of instances truly classified by the leaf to total number of instances classified by that leaf. Mathematically, it is defined by Eq. 6.

$$\mathrm{CF}\ (\mathrm{C_i}) = \frac{\text{Number of instances of class } \mathrm{C_i} \text{ truly classified by the leaf node} + 1}{\text{Number of instances of class } \mathrm{C_i} \text{ truly classified by the leaf node} + \text{Number of instance misclassified by the leaf node} + \mathrm{K}} \tag{6}$$

where CF(Ci) = Confidence factor of the leaf node for predicting the class i, K = Number of output classes.

Addition of 1 in numerator and K in the denominator is due to the Laplace correction. Different decision tree models (C5, CHAID, CART, QUEST) are trained and rules with highest confidence factors are selected from each model for each category to design our proposed integrated model. The format of the rules generated by decision tree models is: *Rule$_i$ for Attack_Type* (*number of instances classified, Confidence factor*). For example, suppose C5 generates following rules for DoS attack:

Rule1 for DoS (5,1.0)
Rule2 for DoS (3, 0.6)
Rule3 for DoS (8, 1.0)
Rule4 for DoS (5, 0.5)
Rule4 for DoS (2, 0.7)

According to this example, threshold confidence factor to select the rule for DoS attack on C5 model is 1.0. As a result, Rule1 and Rule3 are considered as a rule for DoS attack on C5 model. In this way it is considered that the threshold confidence factor for C5 model is 1.0, for CHAID it is 0.92, for CART it is 0.88 and for QUEST it is 0.83. Rules are chosen for each attack from each model applying the threshold confidence factor of that model. Table 7 describe the rules for different attack from each decision tree based models on the basis of threshold confidence factor of the rules.

From Table 7, rule compositions are made for each model. Rules are combined using logical OR operation in each category of particular model. The composition of rules is shown in Table 8. Rules from Table 8 are then combined categories wise which is shown in Table 9.

3.5 Performance analysis on traditional dataset (UNSW-NB15)

The proposed integrated rule-based model is evaluated on the basis of Accuracy (*Acc*), mean F-measure (*MFM*), average accuracy (*AvgAcc*), attack accuracy (*AttAcc*), *ADR* and *FAR* [26]. These metrics are computed using the equations mentioned from 7 to15.

Accuracy (*Acc*) Accuracy measures the frequency of correct classification of a category. It is measured by the fraction of the correct classification of category among all classes divided by the total number of samples in the dataset which is shown in Eq. 7.

$$\text{Accuracy} = \frac{\sum_{i=1}^{|c|} TP_i}{N} \tag{7}$$

F-measure (*MFM*) F-measure is implanted to measure the balance between precision and recall. MFM is computed using Eq. 8.

$$\text{MFM} = \frac{\sum_{i=1}^{|c|} FMeasure_i}{|c|} \tag{8}$$

where

$$\text{FMeasure} = \frac{2 \cdot REcall_i \cdot PREcission_i}{REcall_i + PREcission_i} \tag{9}$$

$$\text{Precission}_i = \frac{TP_i}{TP_i + FP_i} \tag{10}$$

$$\text{Recall}_i = \frac{TP_i}{TP_i + FN_i} \tag{11}$$

FP_i = Represent instances with the actual class other than ith class, TP_i = Number of instances actualy belong to class*i*and predicted class *i*, FN_i = Number of instances actualy belong to class *i*and falsely predicted to belong to another class.

Average accuracy (*AvgAcc*) It is calculated by taking the average of recall of all the classes of a dataset by using Eq. 12.

$$\text{AvgAcc} = \frac{1}{C}\sum_{i=1}^{|C|} Recall_i \tag{12}$$

Attack Accuracy (*AttAcc*) It is used to measure the efficiency of a model to detect only attack classes excluding normal traffic. It can be computed as in Eq. 13.

$$\text{AttAcc} = \frac{1}{C-1}\sum_{i=2}^{|C|} Recall_i \tag{13}$$

Attack detection rate (*ADR*) Accuracy rate for the attack classes can be defined as in Eq. 14.

$$\text{ADR} = \frac{\sum_{i=2}^{|C|} TP_i}{\sum_{i=2}^{|C|} TP_i + FP_i} \tag{14}$$

False alarm rate (*FAR*) It defines normal instances misclassified as attack and can be measured as in Eq. 15.

$$\text{FAR} = \frac{FN_1}{TP_1 + FN_1} \tag{15}$$

In this section, the performance of the proposed model is evaluated using the UNSW-NB15 test dataset and compared with other existing techniques. Confusion matrix is obtained for UNSW-NB15 test dataset on both the C5 and proposed integrated model which is shown in Table 10 and Table 11 respectively. Table 12 shows the confusion matrix for UNSW-NB15 based on Dendron proposed in [26]. Confusion matrix is an N × N matrix, where *N* is total number of classes. It is introduced to visualize that how instances of dataset are classified. Diagonal entries show the number of instances correctly classified. Row class indicates the actual class label of a data instance and the column class indicates the predicted class label of a dataset.

Figures 6 and 7 depict precision and recall of different categories on UNSW-NB15 testing dataset in proposed integrated model and C5 model respectively. From both of these figures and Table 13, it can be observed that the proposed model is showing higher precision and recall for Probe, Normal, Generic, Exploit and average for DoS with the testing data set compared to C5 model.

In this paper, it is found that accuracy of C5 decision tree based model is high compared to other existing decision tree based model. This paper proposes its own model and compares the performance of proposed model with C5 decision tree based model. Analysis of the test dataset on C5 and proposed integrated model is shown in Table 14

Table 7 Generated rules for different attacks from each decision tree based models

Model	Attacks	Rule Set
C5	DoS	Rule1 (3;1.0)
		Rule2 (2;1.0)
		Rule3 (15;1.0)
		…
		…
		Rule68 (2;1.0)
		Rule69 (5;1.0)
	Exploit	Rule1 (2;1.0)
		Rule2 (5;1.0)
		Rule3 (9;1.0)
		…
		…
		Rule128 (7;1.0)
		Rule129 (4;1.0)
	Generic	Rule1 (4;1.0)
		Rule2 (52;1.0)
		…
		…
		Rule36 (37,672;1.0)
	Normal	Rule1 (2670;1.0)
		…
		…
		Rule88 (1953;1.0)
	Probe	Rule1 (3926;1.0)
		…
		Rule20 (3;1.0)
CHAID	Exploit	Rule1(2611;0.922)
		Rule2(18,854;0.94)
		Rule3(3046;0.93)
	Generic	Rule1(39,111;1.0)
	Normal	Rule1 (3926;1.0)
		…
		Rule9 (8260;1.0)
CART	Exploit	Rule1(9443;0.88)
	Generic	Rule1 (26,348;1.0)
	Normal	Rule1 (29,868;0.993)
QUEST	Generic	Rule1 (32,890;0.834)

Table 8 Depicts the composition of rules category wise for each model

	Number of rules attack wise	Total number of rules
C5 rule composition		
$R_{C5}D1 + R_{C5}D2 + R_{C5}D3 + RC5D4 \cdots R_{C5}D65 + R_{C5}D66 \rightarrow R_{C5}D$	Dos(69R)	342R
$R_{C5}E1 + \cdots + R_{C5}E126 \rightarrow R_{C5}E$	Exploit(129R)	
$R_{C5}G1 + \cdots + R_{C5}G36 \rightarrow R_{C5}G$	Generic(36R)	
$R_{C5}P1 + \cdots + R_{C5}P20 \rightarrow R_{C5}P$	Probe(20R)	
$R_{C5}N1 + \cdots + R_{C5}N88 \rightarrow R_{C5}N$	Normal(88R)	
CHAID rule composition		13R
$R_{CHAID}E1 + \cdots + R_{CART}E3 \rightarrow R_{CHAID}E$	Exploit(3R)	
$R_{CHAID}G1 \rightarrow R_{CHAID}G$	Generic(1R)	
$R_{CHAID}N1 + \cdots + R_{CHAID}N9 \rightarrow R_{CHAID}N$	Normal(9R)	
CART rule composition		3R
$R_{CART}E1 \rightarrow R_{CART}E$	Exploit(1R)	
$R_{CART}G1 \rightarrow R_{CART}G$	Generic(1R)	
$R_{CART}N1 \rightarrow R_{CART}N$	Normal(1R)	
QUEST rule composition		
$R_{QUEST}G1 \rightarrow R_{QUEST}G$	Quest(1R)	1R

Table 9 Rule composition for proposed model

	Number of rules attack wise	Total number of rules
$R_{C5}D + R_{CHAID}D + R_{CART}D + R_{QUEST}D \rightarrow R_D$	DoS (69R)	359R
$R_{C5}E + R_{CHAID}E + R_{CART}E + R_{QUEST}E \rightarrow R_E$	Exploit (133R)	
$R_{C5}N + R_{CHAID}N + R_{CART}N + R_{QUEST}N \rightarrow R_N$	Normal (98R)	
$R_{C5}P + R_{CHAID}P + R_{CART}P + R_{QUEST}P \rightarrow R_P$	Probe (20R)	
$R_{C5}G + R_{CHAID}G + R_{CART}G + R_{QUEST}G \rightarrow R_G$	Generic(39R)	

and the graphical visualization is shown in Fig. 8. The proposed IDS model is also compared with the IDS of ENADS [24], which also applies decision tree based model on UNSW-NB15 dataset shown in Fig. 9. It is observed that proposed system shows lower FAR due to lower false negatives compared to ENADS [24].

From Fig. 10 it is observed that performance of proposed integrated model is better than Dendron [26]. The reason behind it that Dendron [26] considers 10 categories of attacks including Normal which are overlapped with each other. In our proposed integrated model 5 categories of attacks including Normal are considered and they represent rest of categories due to their overlapped nature. Further, the proposed model shows low misclassification rate in comparison to Dendron.

Average detection rate (average accuracy (AvgAcc)) is calculated as the average of recall of all the classes present in the dataset. Now, from Fig. 8 it is observed that the average detection rate (AvgAcc) of proposed system (i.e. 65.21%) shows lower value compared to C5 (i.e. 75.8%) due to the lower recall for DoS and Exploits attack. On the other hand, Fig. 10 shows that the average accuracy of proposed system (i.e. 65.21%) is higher compared to Dendron (i.e. 52.21%) due to higher recall of classes of proposed system. Whereas, it is observed from Figs. 8 and 10 that the ADR of proposed system (i.e. 90.32%) is higher compared to C5 (i.e. 83.47%) and Dendron (i.e. 63.76%) due to the higher precision of the all predicted classes of the proposed system.

4 Working example of proposed model

In this module the real time data set is designed and the performance of that dataset on the proposed model is evaluated. Data set is collected from the setup at the CSE lab in NIT Patna. This phase mainly consists of three parts—(1) Data collection and Feature Extraction, (2) Dataset description and (3) Performance Evaluation on Real time dataset (RTNITP18).

Table 10 Confusion matrix of C5 on UNSW test dataset

	Dos	Exploit	Normal	Probe	Generic	Recall (%)
DoS	471	3411	137	22	48	11.52
Exploit	233	10,365	326	162	46	93.12
Normal	138	1206	35,636	7	13	96.31
Probe	14	618	44	2818	2	80.61
Generic	44	376	54	7	18,390	97.45
Precision (%)	52.33	64.88	98.45	93.43	99.41	

Table 11 Confusion matrix of the Proposed Integrated model on UNSW test dataset

	DoS	Exploit	Normal	Probe	Generic	Recall (%)
DoS	206	1219	2628	2	32	5.0
Exploit	156	6085	4864	3	21	54.64
Normal	19	735	36,223	13	1	98.0
Probe	3	444	542	2506	1	71.7
Generic	25	222	370	2	18,252	96.72
Precision (%)	50.37	69.9	81.17	99.21	99.7	

Table 12 Confusion matrix of Dendron for UNSW-NB15 [26]

	Normal	Backdoor	Analysis	Fuzzers	Shellcode	Recon.	Exploits	DoS	Worms	Generic	Recall (%)
Normal	29,982	1	0	313	63	125	257	43	1	1	97.39
Backdoor	25	206	0	7	0	9	28	29	1	1	67.32
Analysis	32	149	82	7	0	0	7	124	0	0	20.45
Fuzzers	999	247	5	2804	31	66	62	135	0	4	64.42
Shellcode	23	0	0	44	123	116	30	2	0	0	36.39
Reconnaissance (Probe)	148	63	0	280	261	1121	375	167	16	2	46.04
Exploits	361	255	26	233	127	225	5220	303	66	33	76.22
DoS	270	242	6	57	72	66	599	221	4	9	14.29
Worms	0	0	0	2	4	2	8	0	4	2	18.37
Generic	109	2	1	46	83	34	263	67	8	2678	81.37
Precision (%)	93.84	17.68	68.33	73.93	16.10	63.55	76.22	20.26	4.00	98.10	

4.1 Data collection and feature extraction phase

In this phase, we have generated our own real time data set to evaluate the proposed model. Data set is generated by establishing a setup at the laboratory of CSE department, in NIT Patna. This lab consists of 40 systems out of which few act as attacker and rest act as normal user or victim and we observe the packet flow in the network for 7 days. Kali Linux is installed on each system for the purpose of observation. Kali Linux is a popular open-source platform which provides set of security tools for hackers. It is an open source and its official webpage is https://www.kali.org. We install metasploitable operating system on victim nodes in the lab. To perform the real time data generation, msfconsole (Kali) is used as an attack generator in the network and used metasploitable an intentionally vulnerable version of Ubuntu as a victim. Feature extraction of Probe, Exploit, DoS, and Generic attacks are given below.

A. Probe NMAP tool is used for Probe attack. This tool is mainly used in scanning phase of attack process. A screenshot for feature extraction of Probe attack is given in Fig. 11.

B. Exploit Information gathered by using Probe attack (or probing) is further used to perform the attack called Exploit. In this attack the attacker node tries to use this information and send packets accordingly in order to attack the victim without being noticed.

C. DoS DoS attack is performed using Ettercap option present in Kali Linux. Ettercap can be opened using command *sudoettercap−G*. After opening Ettercap go to sniff menu, after that go to unified sniffing which will pop

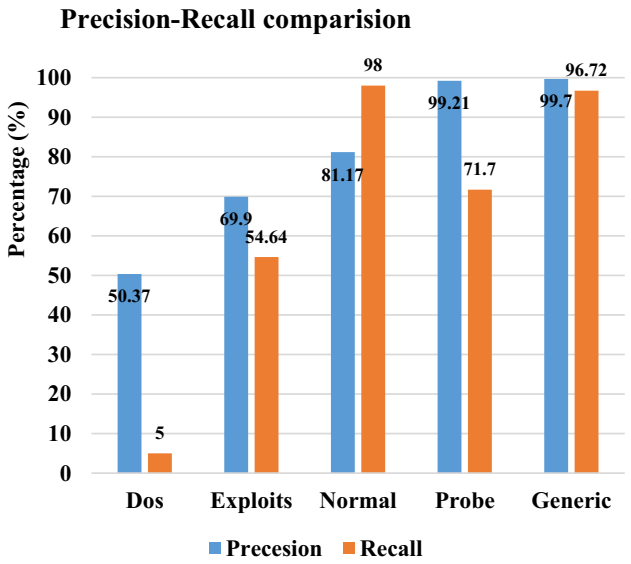

Fig. 6 Precision and Recall comparison of different categories on Proposed Integrated model

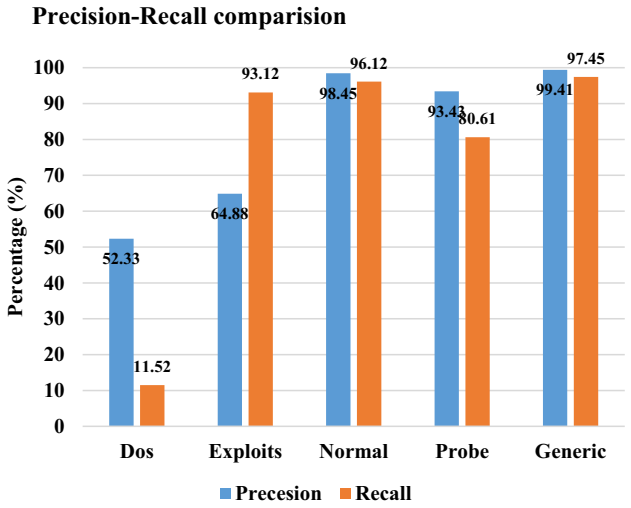

Fig. 7 Precision and Recall comparison of different categories in C5 model

up Ettercap input window with eth0. After this, plugin, protocol dissectors, ports monitored information will be visible. DoS attack can be started by using the plugin. A screenshot showing the attack is given in Fig. 12.

D. Generic It is a type of activity in which an attacker does not bother about the crypto-graphical implementation of any primitives and runs the attack. As an example consider a cipher text with K bit key, in the generic attack of brute force, attacker tries every combination possible using k bits i.e. 2^K combinations and try to decrypt the text. This attack is performed by using hydra in Kali Linux. Hydra is broadly used as login cracker which provides a way to attack several protocols such as: Cisco AAA, FTP, Cisco auth, XMPP etc.

To open hydra, go to Applications → Password Attacks → Online Attacks → hydra. It will open the terminal console. We attack FTP service of metasploiTable machine, which has IP 192.168.1.101. In Kali Linux a word list is created with extension 'lst' in the path usr\share\wordlist\metasploit. Now to perform the attack we use the command *hydra –l/usr/share/wordlists/metasploit/user—p/user/share/wordlists/metasploit/passwords ftp://192:168:1:101—v*. Figure 13 depicts the screen shot of Generic attack where the user name and password is successfully decrypted.

E. normal This category of data does not contain with any malicious activity and is collected from LAN in regular working environment.

Data for all categories are captured using Wireshark packet sniffing tool which can be found at http://git.kali.org/gitweb/?p=packages/wireshark.git;a=summary. A snapshot of Wireshark capturing process is shown in Fig. 14. We have captured 2000 data for each category attack under consideration from 7 consecutive working days. To generate RTNITP dataset attacks are performed using the tools available in the kali Linux. Attack generation are automated either by some command line or directly using the tool and captured at the victim end. Total of 10,000 data packets are sampled for all categories through Wireshark at victim end. The behavior of data packets which are captured at the victim nodes are essential requirement rather than source nodes in order to create RTNITP dataset. Captured data samples are saved in separate file with extension *.pcapng* at victim end. Basic features present in the captured data samples are time, source IP, Destination IP, protocol, length, and info which contains some other information like port numbers, acknowledgement (ACK) bit, segment size, window size etc. Apart from the basic feature we need derived features for the data set. For this, we have exported .pcapng file to .txt which contains all the other detailed information like time to live field (TTL). We have used this *v* file to extract derived features using the concept of networking. e.g., to calculate ***sbytes*** we have

Table 13 Metrics summary of C5 and Proposed Integrated model using UNSW-NB15 test dataset

Model	Acc (%)	MFM (%)	AvgAcc (%)	AttAcc (%)	ADR (%)	FAR (%)
C5	90.74	75.54	75.8	70.65	83.47	3.7
Proposed	84.83	68.13	65.21	57.01	90.32	2.01

Table 14 Features extracted and mapped to the traditional data set features

Size of total data sample captured	10,000	
Features extracted	Basic features	1. protocol(proto)
	Derived features	2. ct_srv_dst
		3. ct_dst_src_ltm
		4. service
		5. sbytes
		6. dbytes
		7. sttl
		8. dloss
		9.dinpkt
		10.ct_srv_src
		11.ct_dst_ltm
		12.ct_src_dport_ltm
		13.ct_dst_sport_ltm

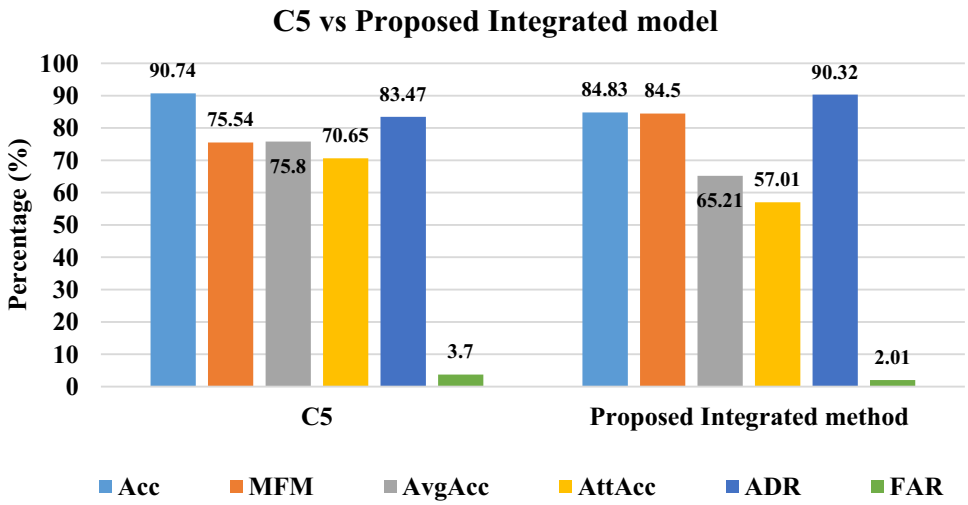

Fig. 8 Performance comparison of C5 and Proposed Integrated model on UNSW test dataset

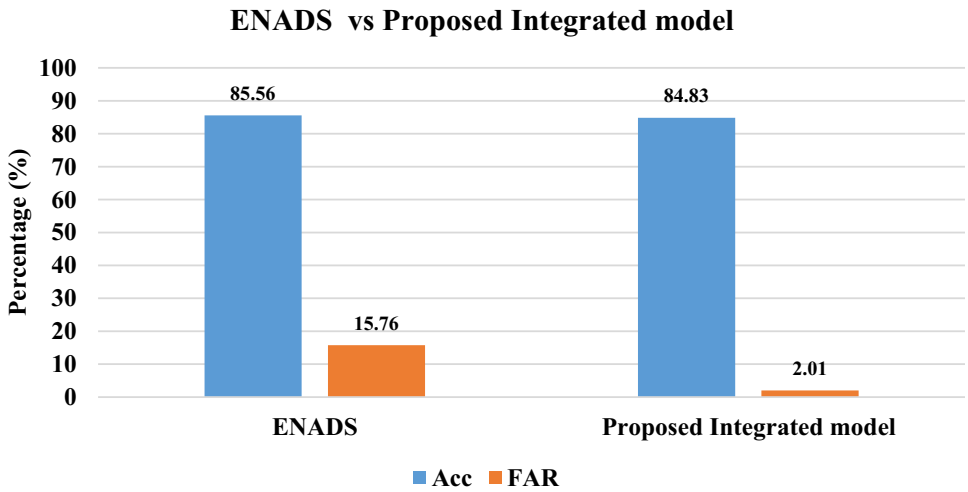

Fig. 9 Performance comparison of ENADS [24] and Proposed Integrated model on UNSW test dataset

chosen pairs of source IP and destination IP and calculated the total number of bytes sent from source to destination. In similar fashion, ***dbytes*** are calculated. We have used python script to automate the extraction of derived features and map those features to the 13 features of traditional dataset. This real time data set is known as RTNITP18 and

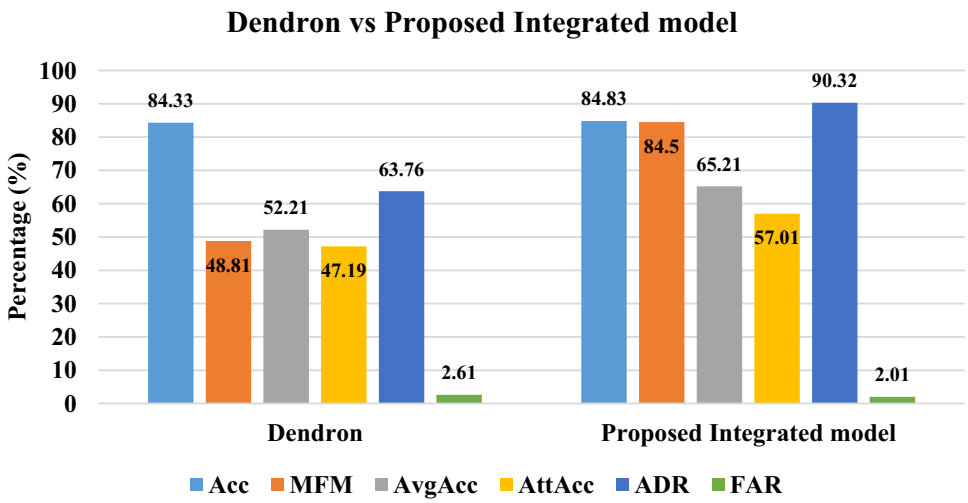

Fig. 10 Comparison of Dendron [26] versus Proposed Integrated model on UNSW-NB 15 test dataset

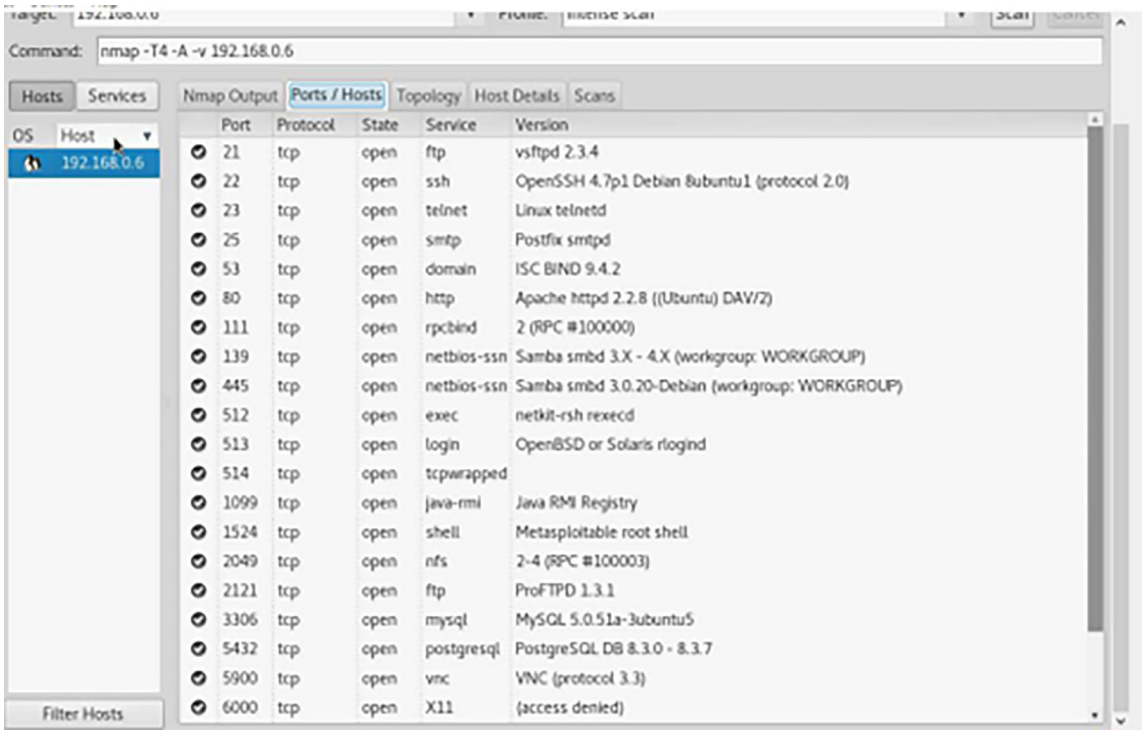

Fig. 11 Feature extraction for Probe attack

our proposed integrated model is evaluated on this dataset. Features detailed of RTNITP18 are shown in Table 14.

4.2 Dataset description

Dataset is generated at NIT Patna lab with 13 features, 10000 instances and five category of attacks (DoS, Probe, Generic, Exploit, Normal) are called Real Time Dataset at NIT Patna (RTNITP18). Now, RTNITP18 data set acts as the testing data set. We evaluate the performance of the proposed classification based model on theRTNITP18 data set.

RTNITP18 data set is captured in separate file for each category of attack. We have selected random data in same proportion i.e. 10% from each category. So we have chosen 200 randomly sampled data for each category shown in Table 15. We have stored each category in separate.csv file and for testing the proposed model merged all the file in single.csv file.

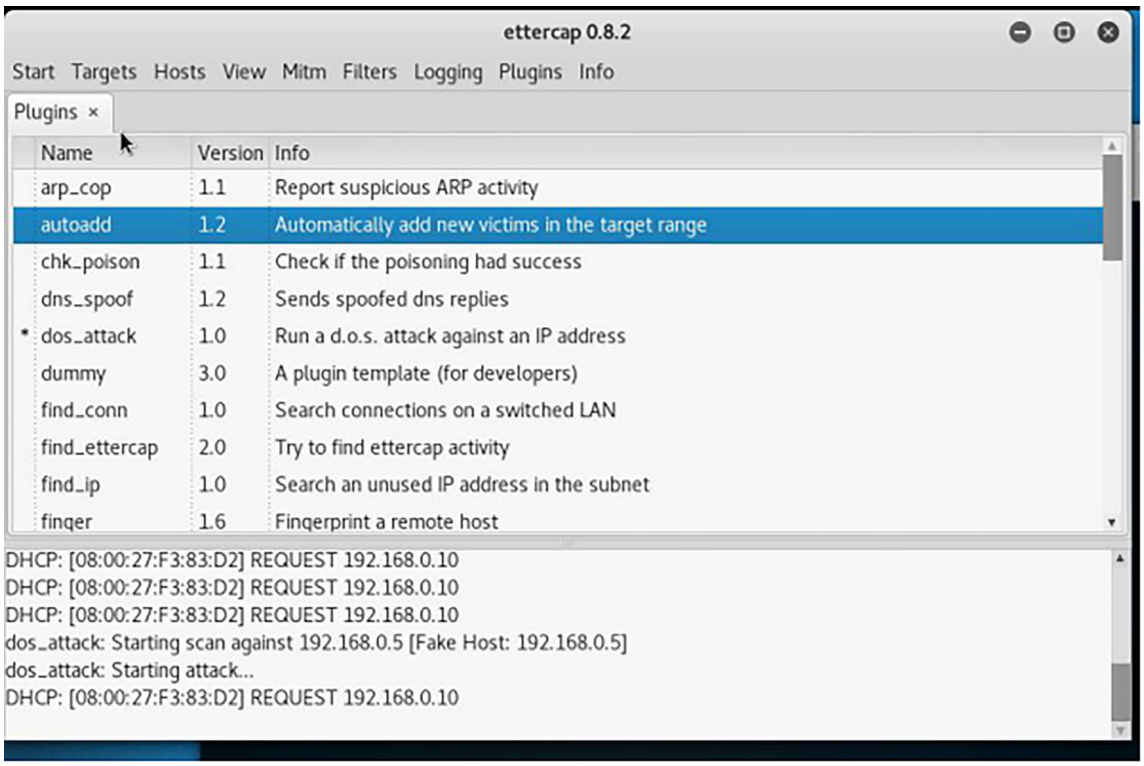

Fig. 12 Performing DoS attack

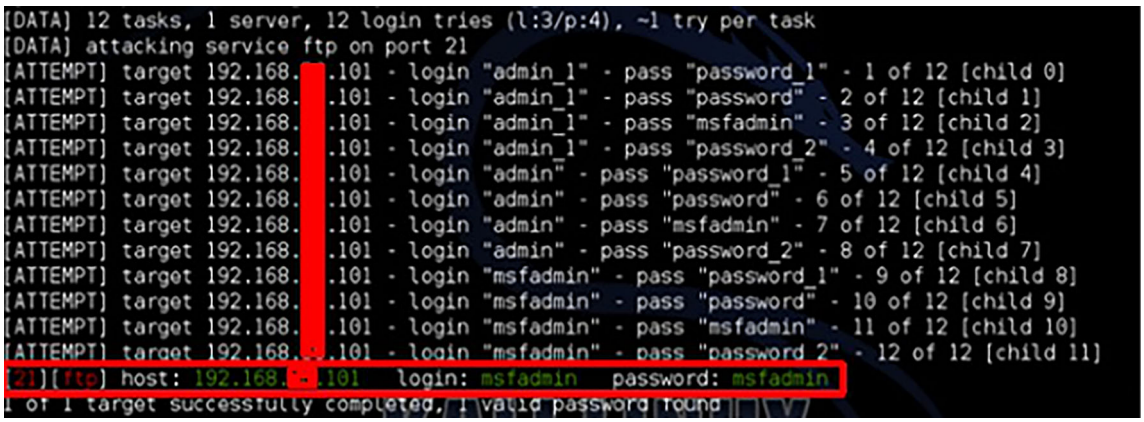

Fig. 13 Process performing Generic Attack

4.3 Performance evaluation on real time dataset (RTNITP18)

This part explains the performance of proposed model on the RTNITP18 data set. The set is supplied to the proposed model in order to evaluate its performance. Confusion matrix for the different types of attacks for the proposed IDS model is shown in Table 16. In Table 17 precision & recall is shown and Table 18 shows accuracy, ADR and FAR and other metrics value for RTNITP18 on proposed model.

It is observed from Fig. 15 that, Normal and Probe have highest precision and recall values compared to DoS and Exploit. Furthermore, DoS has higher recall compared to Exploit. On the real time data set (RTNITP18), proposed model gives an accuracy of 83.8% and ADR 88.29% for five categories as shown in Fig. 16. Hence it can be concluded that performance of the proposed model is also good enough on RTNITP18 dataset.

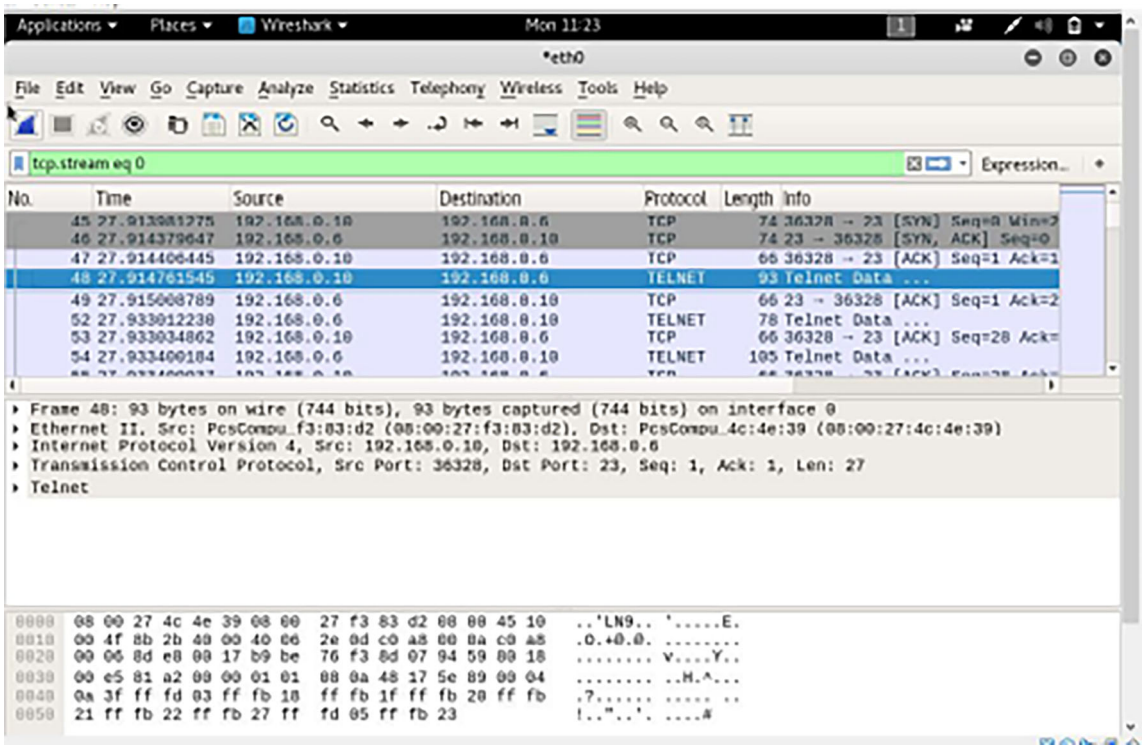

Fig. 14 Feature extraction applying Wireshark

Table 15 RTNITP18 data set for each category of attack

	DoS	Exploit	Normal	Generic	Probe
Number of Instances	200	200	200	200	200

Table 16 Confusion matrix for different categories using rtnitp18 dataset on the proposed ids

	DoS	Exploit	Normal	Probe	Generic
DoS	139	37	0	0	0
Exploit	0	197	3	0	18
Normal	2	5	182	1	0
Probe	0	7	7	177	1
Generic	6	10	19	0	143

Table 17 Precision & Recall values for RTNITP18 on proposed model is as follows

	Recall	Precision	F-measure (%)
DoS	78.98	94.55	86.10
Exploit	90.36	76.95	83.12
Normal	95.79	86.25	90.77
Generic	80.33	88.27	84.11
Probe	92.19	99.43	95.67

Table 18 Performance metrics of the proposed model on RTNITP18 dataset

Accuracy(%)	MFM (%)	AvgAcc (%)	AttAcc (%)	ADR(%)	FAR (%)
83.8	87.95	87.53	85.46	88.29	4.0

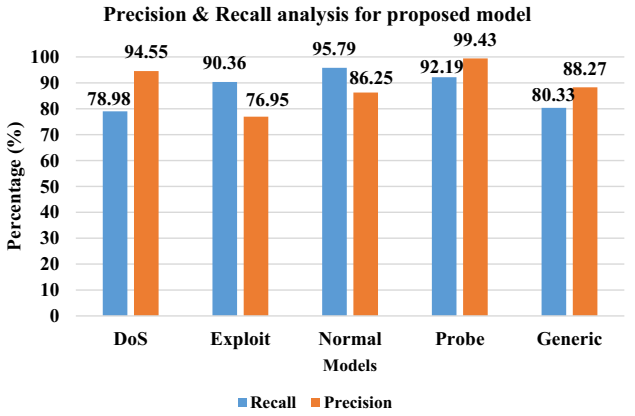

Fig. 15 Precision and Recall of RTNITP18 dataset on proposed integrated model

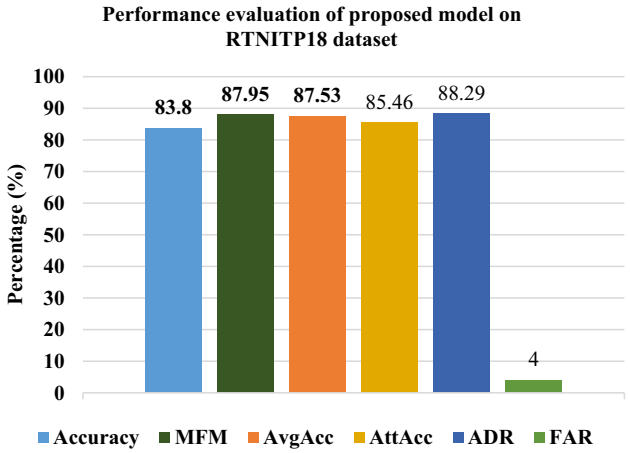

Fig. 16 Performance analysis of proposed model using RTNITP18

5 Conclusions

This paper proposes an integrated classification based IDS and evaluates its performance on offline traditional data set and on line real time data set. This paper evaluates the performance of proposed model on a new data set (UNSW-NB15) which covers the most recent attacks (DoS, Exploit, Normal, Probe, Generic) compared to KDD99 data set. It is observed that the value of several evaluation metrics (e.g. MFM = 84.5%, ADR = 90.32, FAR = 2.01% etc.) have higher performance compared to other existing traditional decision tree based models. Since the proposed approach is based on the misuse-based technique so it is not able to detect any zero day attacks which are publicly unknown and hence there is no signature found for that attack. But once the attack is performed the signature is available to the proposed IDS model. Now our IDS model is updated with the signature to prevent the attacks of these categories. This paper generates a real time data set at NIT Patna CSE lab (RTNITP18) and it acts as the testing data set to evaluate the performance of our proposed model. Accuracy of proposed model is 83.8%. We can conclude that our proposed integrated model acts as the dog watcher in the network to prevent the systems of the organisation from malicious attacks. In future we will try to improve following drawbacks of our proposed integrated model, such as: (1) Enhance the detection rate on real time data set and (2) Develop the ability to classify new unknown attacks.

References

1. Agarwal, M., Pasumarthi, D., Biswas, S., Nandi, S.: Machine learning approach for detection of flooding DoS attacks in 802.11 networks and attacker localization. Int. J. Mach. Learn. Cybern. (2016). https://doi.org/10.1007/s13042-014-0309-2
2. Aghdam, M.H., Kabiri, P.: Feature selection for intrusion detection system using ant colony optimization. IJ Netw. Secur. **18**(3), 420–432 (2016)
3. Akshaya, P.: Intrusion detection system using machine learning approach. Int. J. Eng. Comput. Sci. **5**(10), 18249–18254 (2016)
4. Alomari, E., Manickam, S., Gupta, B.B., Karuppayah, S., Alfaris, R.: Botnet-based distributed denial of service (DDoS) attacks on web servers: classification and art. arXiv preprint arXiv:1208.0403 (2012)
5. Banerjee, U., Vashishtha, A., Saxena, M.: Evaluation of the capabilities of WireShark as a tool for intrusion detection. Int. J. Comput. Appl. **6**(7), 1–5 (2010)
6. Chowdhury, M.N., Ferens, K., Ferens, M.: Network Intrusion Detection Using Machine Learning. In: Proceedings of the International Conference on Security and Management (SAM), p. 30 (2016)
7. Das, V., Pathak, V., Sharma, S., Srikanth, M.V.V.N.S., Kumar, G., Nadu, T.: Network intrusion detection system based on machine learning algorithms. Int. J. Comput. Sci. Inf. Technol. (2010). https://doi.org/10.5121/ijcsit.2010.2613
8. Fares, A.H., Sharawy, M.I., Zayed, H.H.: Intrusion detection: supervised machine learning. J. Comput. Sci. Eng. (2011). https://doi.org/10.5626/JCSE.2011.5.4.305
9. Garcia-Teodoro, P., Diaz-Verdejo, J., Maciá-Fernández, G., Vázquez, E.: Anomaly-based network intrusion detection: techniques, systems and challenges. Comput. Secur. (2009). https://doi.org/10.1016/j.cose.2008.08.003
10. Goutte, C., Gaussier, E.: A probabilistic interpretation of precision, recall and F-score, with implication for evaluation. European Conference on Information Retrieval, pp. 345–359. Springer, Berlin (2005)
11. Gou, Z., Ahmadon, M.A.B., Yamaguchi, S., Gupta, B.B.: A Petri net-based framework of intrusion detection systems. In: 2015 IEEE 4th Global Conference on Consumer Electronics (GCCE) (pp. 579–583). IEEE (2015, October)
12. Gupta, B., Agrawal, D.P., Yamaguchi, S.: Handbook of research on modern cryptographic solutions for computer and cyber security. IGI Global, Pennsylvania (2016)
13. Gupta, B.B., Misra, M., Joshi, R.C.: FVBA: a combined statistical approach for low rate degrading and high bandwidth disruptive DDoS attacks detection in ISP domain. In: 2008 16th IEEE International Conference on Networks (pp. 1–4). IEEE (2008, December)
14. Hu, J., Yu, X., Qiu, D., Chen, H.H.: A simple and efficient hidden Markov model scheme for host-based anomaly intrusion detection. IEEE Netw. **23**(1), 42–47 (2009)
15. Ibrahim, H.E., Badr, S.M., Shaheen, M.A.: Adaptive layered approach using machine learning techniques with gain ratio for intrusion detection systems. Int. J. Comput. Appl. **56**(7), 10–16 (2012)
16. Jha, J., Ragha, L.: Intrusion detection system using support vector machine. IJAIS. **ICWAC**(3), 25–30 (2013)
17. Kalekar, A., Kshatriya, N., Chakranarayan, S., Wadekar, S.: Real time intrusion detection system using machine learning. Int. J. Eng. Res. Technol. **3**(2), 185–187 (2014)
18. KDD 99 data set. http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html. Accessed Feb 14, 2018
19. Kułakowski, P., Vales-Alonso, J., Egea-López, E., Ludwin, W., García-Haro, J.: Angle-of-arrival localization based on antenna arrays for wireless sensor networks. Comput. Electr. Eng. (2010). https://doi.org/10.1016/j.compeleceng.2010.03.007
20. Mabu, S., Chen, C., Lu, N., Shimada, K., Hirasawa, K.: An intrusion-detection model based on fuzzy class-association-rule mining using genetic network programming. IEEE Trans. Syst. Man Cybern. C **41**(1), 130–139 (2011)

21. Mishra, A., Gupta, B.B., Joshi, R.C.: A comparative study of distributed denial of service attacks, intrusion tolerance and mitigation techniques. In: 2011 European Intelligence and Security Informatics Conference (pp. 286–289). IEEE (2011, September)
22. Modi, U., Jain, A.: An improved method to detect intrusion. Inf. Eng. (2016). https://doi.org/10.5121/ieij.2016.4203
23. Moustafa, N., Slay, J.: UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In: Military Communications and Information Systems Conference (MilCIS), 2015. IEEE, pp. 1–6, (2015)
24. Moustafa, N., Slay, J.: The evaluation of network anomaly detection systems: statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set. Inf. Secur. J. (2016). https://doi.org/10.1080/19393555.2015.1125974
25. Negi, P., Mishra, A., Gupta, B.B.: Enhanced CBF packet filtering method to detect DDoS attack in cloud computing environment. arXiv preprint arXiv:1304.7073 (2013)
26. Papamartzivanos, D., Mármol, F.G., Kambourakis, G.: Dendron: genetic trees driven rule induction for network intrusion detection systems. Futur. Gener. Comput. Syst. **79**, 558–574 (2018)
27. Revathi, S., Malathi, A.: A detailed analysis on NSL-KDD dataset using various machine learning techniques for intrusion detection. Int. J. Eng. Res. Technol. **2**(12), 1848–1853 (2013)
28. Sangkatsanee, P., Wattanapongsakorn, N., Charnsripinyo, C.: Practical real-time intrusion detection using machine learning approaches. Comput. Commun. (2011). https://doi.org/10.1016/j.comcom.2011.07.001
29. Sasan, H.P.S., Sharma, M.: Intrusion detection using feature selection and machine learning algorithm with misuse detection. Int. J. Comput. Sci. Inf. Technol. (2016). https://doi.org/10.5121/ijcsit.2016.8102
30. Sindhu, S.S.S., Geetha, S., Kannan, A.: Decision tree based light weight intrusion detection using a wrapper approach. Expert Syst. Appl. (2012). https://doi.org/10.1016/j.eswa.2011.06.013
31. Subhan, F., Hasbullah, H., Ashraf, K.: Kalman filter-based hybrid indoor position estimation technique in bluetooth networks. Int. J. Navig. Observ. (2013). https://doi.org/10.1155/2013/570964
32. Wang, C., He, Q., Shao, M., Hu, Q.: Feature selection based on maximal neighborhood discernibility. Int. J. Mach. Learn. Cybern. (2017). https://doi.org/10.1007/s13042-017-0712-6
33. Wattanapongsakorn, N., Charnsripinyo, C.: Web-based monitoring approach for network-based intrusion detection and prevention. Multimed. Tools Appl. (2015). https://doi.org/10.1007/s11042-014-2097-9
34. Weka 3.6.0 tools. http://www.cs.waikato.ac.nz/ml/weka/. Accessed 15 January 2017
35. Yasami, Y., Mozaffari, S.P.: A novel unsupervised classification approach for network anomaly detection by k-Means clustering and ID3 decision tree learning methods. J. Supercomput. (2010). https://doi.org/10.1007/s11227-009-0338-x
36. Yin, C., Ma, L., Feng, L.: Towards accurate intrusion detection based on improved clonal selection algorithm. Multimed. Tools Appl. (2017). https://doi.org/10.1007/s11042-015-3117-0
37. Zhan, J., Malik, H.M., Akram, M.: Novel decision-making algorithms based on intuitionistic fuzzy rough environment. Int. J. Mach. Learn. Cybern. (2018). https://doi.org/10.1007/s13042-018-0827-4

Vikash Kumar is pursuing his M.Tech.–Ph.D. dual degree in Computer Science and Engineering Department from National Institute of Technology Patna India. His research interest includes WSN, Network security, and Intrusion Detection System.

Ditipriya Sinha Assistant Professor CSE Department, National Institute of Technology Patna. She was an Assistant Professor in the Department of CSE, Birla Institute of Technology, Mesra. She has received her Ph.D. degree in the department of Computer Science and Technology, Indian Institute of Engineering Science and Technology (IIEST), Shibpur and Master of Technology from West Bengal University of Technology in the department of Software Engineering. She was a silver medalist in the degree of Master of Technology. Her area of research is Mobile Adhoc Network, Wireless Sensor Network and Scheduling algorithms.

Ayan Kumar Das has received Ph.D. degree in the Department of Computer Science and Engineering, University of Calcutta and received Master of Technology from West Bengal University of Technology in the department of Software Engineering. He is presently serving as an Assistant Professor in the department of Computer Science and Engineering, Birla Institute of Technology, Mesra. He was an Assistant Professor in the department of Information Technology, Calcutta Institute of Engineering and Management. His area of research is Wireless Sensor Network, Internet of Things and Cloud Computing.

Subhash Chandra Pandey completed his Ph.D. from Dr. APJ Abdul Kalam Technical University, Lucknow, India and M.Tech. in computer engineering from Motilal Nehru National Institute of Technology, Allahabad, UP, India. Presently, he is associated with Birla Institute of Technology, Mesra, Ranchi (Patna Campus) as an Assistant Professor. He is member of many professional bodies including "International Association of Engineers – Hong Kong", "Computer Science Teachers Association – USA" and "International Association of Artificial Intelligence and Law". He is regularly publishing the research papers in reputed international journals and international conference proceedings. His current research interests include soft computing, evolutionary computing, bio-inspired computing, machine intelligence, data mining, Philosophical aspects of machine cognition. Nowadays, he is working on network security.

Radha Tamal Goswami Director Techno India College of Technology, Professor in the Department of Computer Science and Engineering, Newtown, Kolkata India. Dr. Radha Tamal Goswami has received his Ph.D. in Technology from Birla Institute of Technology Mesra Ranchi India. He is having 23 years of experience in the field of academics and research. He was the professor in Computer Science and Engineering and also the Director of BIT Mesra Kolkata Campus since 1995. He joined Techno India College of Technology Kolkata as a Director in September 1, 2016 on 2 years' lien from BIT Mesra. His research interest in the field of Network Security and BigData. He has conducted almost 30 MDP and FDP program. He has guided more than 100 students in UG and PG projects. He is the visiting faculty of ten Institutions and member of ACM, IEEE, CSI and NIPM. Published almost 30 research papers. He chaired many National and International conferences.