Implementation of Machine Learning Algorithms on CICIDS-2017 Dataset for Intrusion Detection using WEKA

Article *in* International Journal of Recent Technology and Engineering (IJRTE) · September 2019

DOI: 10.35940/ijrte.C4587.098319

Lokesh Panwar
THDC Institute of Hydropower Engineering Technology

Shailesh Panwar
Hemwati Nandan Bahuguna Garhwal University

Implementation of Machine Learning Algorithms on CICIDS-2017 Dataset for Intrusion Detection Using WEKA

Shailesh Singh Panwar, Pritam Singh Negi, Lokesh Singh Panwar, Y. P. Raiwani

Abstract: For protecting and securing the network, with Intrusion Detection Systems through hidden intrusion has become a popular and important issue in the network security domain. Detection of attacks is the first step to secure any system. In this paper, the main focus is on seven different attacks, including Brute Force attack, Heartbleed/Denial-of-service (DoS), Web Attack, Infiltration, Botnet, Port Scan and Distributed Denial of Service (DDoS). We rely on features derived from CICIDS-2017 Dataset for these attacks. By using various subset based feature selection techniques performance of attack has been identified for many features. Using these techniques, it has been determined the appropriate group of attributes for finding every attack with related classification algorithms. Simulations of these techniques present that unwanted feature can be removed from attack detection techniques and find the most valuable set of attributes for a definite classification algorithm with discretization and without discretization, which improve the performance of IDS.

Keywords: IDS, CICIDS-2017, Classification Algorithms, Features Selection, WEKA.

I. INTRODUCTION

In the era of the continuously growing internet, securing the network becomes an important issue and challenge in our life. The role of intrusion detection techniques for network security is important for detecting any intrusion. Intrusion detection techniques are not only used to detect the attacks but also take an alarm on any unusual behaviour in a network. The behaviour of the intruded network is uncommon in general; it can be attack or deviation [1].

There are three ways for detecting any intrusion: - (i) Misuse-based or Signature Based Method; which detects familiar attacks, using their signatures without producing a huge number of false alarms. (ii) Anomaly based methods analyses the conventional system and network performance and recognize inconsistency as deviations from general network behaviour. New or fresh (zero-day) attacks can be recognized by this method. (iii) Hybrid methods are the combination of Signature and Anomaly Based methods. This method is used to enhance the familiar intrusion detection rate and for hidden attacks to decrease the false positive rate [2].

The feature selection techniques focus and maintain the significant attributes and enhance the entire performance of IDS. For this, an effective model has been developed with the help of the WEKA tool [3, 4]. Different selected attributes and classification algorithms on the WEKA tool are used to create a useful and powerful framework to detect intrusion. WEKA is defined as a collection of machine learning and a group of data mining algorithms [5]. Section-II of this paper describes the CICIDS-2017 dataset; the section-III describes the outline of feature selection and classifier algorithms and section-IV introduces performance evaluation. Simulation results are analyses in section-V followed by conclusion and future scope in section-VI.

II. RELATED WORKS

Previous research efforts for providing a detailed analysis of the supervised machine learning techniques used for intrusion detection are summarized in Table 1. These studies focused on training and testing different machine learning approaches using standard intrusion detection on datasets. Therefore, we have two possible choices: either using a subset of these standard datasets or extracting new features based on network traces of standard datasets or statistics provided by the controller. In this study, we have used subset of features selection from the CICIDS-2017 dataset based on Correlation Feature Selection and Classifier Subset Evaluator approach and have considered J48, decision trees (DTs) and Naive Bayes (NB) with discretization and without discretization supervised machine learning approaches. For performance measurement, we are computing accuracy, precision, recall, F-measure and time taken to build a model (i.e. 90% training dataset and 10 % testing dataset).

Table 1: Previous Intrusion Detection Systems (IDS) Based On Machine Learning Algorithms

Author/ Year	**Algorithms/ Methods**	**Dataset**
Belavagi 2016 [6]	LR, Gaussian NB, SVM and RF	NSL-KDD
Yin et al. 2017 [7]	J48, ANN, RF,SVM, RNN-IDS	NSL-KDD
Khaled *et al. 2017* [8]	RBM together with the DBNs	KDDCUP'99
Zhao *et al.2017* [9]	DBNs with PNN	KDDCUP'99
Roy *et al. 2017*[10]	DNN	KDDCUP'99
Xu et al. 2018 [11]	GRU and LSTM	KDD99 and NSL-KDD
Wu et al.2018 [12]	NB, NBT, J48, RF, RFT,MLP, SVM,RNN,CNN	NSL-KDD

Revised Manuscript Received on August 22, 2019.
***** Correspondence Author

Shailesh Singh Panwar, Department of Computer Science and Engineering, H.N.B. Garhwal University Srinagar Garhwal, Uttarakhand, India, Email: shaileshpanwar23@gmail.com

Pritam Singh Negi, Department of Computer Science and Engineering, H.N.B. Garhwal University Srinagar Garhwal, Uttarakhand, India.
Email: negipritam@gmail.com

Lokesh Singh Panwar, Department of Electronics and Communication Engineering, H.N.B. Garhwal University Srinagar Garhwal, Uttarakhand, India. Email: lokesh31j@gmail.com.

Y. P. Raiwani*, Department of Computer Science and Engineering, H.N.B. Garhwal University Srinagar Garhwal, Uttarakhand, India.
Email: yp_raiwani@yahoo.com

Wang et al.2018 [13]	FGSM, JSMA	NSL-KDD
Shone et al. 2018 [14]	DBN, S-NDAE	KDD99 and NSL-KDD
Lia et al.2018 [15]	C4.5, RF, k-NN, BPNN, NB,	NSL-KDD
Latah et al.2018 [16]	DT, NB, LDA, NN, SVM, RF, k-NN, AdaBoost, RUSBoost, LogitBoost and BT	NSL-KDD
Kamarudin et al2017 [17].	NB, SVM, MLP, DT,	NSL-KDD
Jia et al.2019 [18]	NDNN	KDD99 and NSL-KDD
Gogoi et al. 2013 [19]	Multi-level hybrid intrusion detection method that	KDD99, NSL-KDD and TUIDS
Gao et al. 2018 [20]	J48, NB, NBT, RF, RT, MLP, SVM etc.	NSL-KDD
Ambusaidi et al2016 [21]	LSSVM + FMIFS, LSSVM + MIFS	KDD Cup 99, NSL-KDD and Kyoto 2006+
Panwar et al. 2014 [22]	NB, J48	NSL-KDD
Raiwani et al. 2015 [23]	MLP, SGD, VP, LR	NSL-KDD
Raiwani et al. 2014 [24]	KM, EM, DB, COBWEB	NSL-KDD
Panwar et al. 2019 [25]	OneR, REPTree	CICIDS-2017
Baluni et al. 2014[26]	FFNN	UKD Road Accident

III. DATASET

Since the inception of CICIDS-2017 dataset, the dataset started attracting researchers for analysis and developments of new models and algorithms [27, 28, 29]. According to the author [29] of CICIDS-2017, the dataset spanned over eight different files containing five days normal and attacks traffic data of the Canadian Institute of Cyber security. A short description of all those files is presented in the Table 2.

Table 2: Description of files containing CICIDS-2017 dataset

Name of Files	Day Activity	Attacks Found
Monday-WorkingHours.pcap_ISCX.csv	Monday	Benign (Normal human activities)
Tuesday-WorkingHours.pcap_ISCX.csv	Tuesday	Benign, FTP-Patator, SSH-Patator
Wednesday-workingHours.pcap_ISCX.csv	Wednesday	Benign, DoS GoldenEye, DoS Hulk, DoS Slowhttptest, DoS slowloris, Heartbleed
Thursday-WorkingHours-Morning-WebAttacks.pcap_ISCX.csv	Thursday	Benign, Web Attack – Brute Force, Web Attack – Sql Injection, Web Attack – XSS
Thursday-WorkingHours-Afternoon-Infilteration.pcap_ISCX.csv	Thursday	Benign, Infiltration
Friday-WorkingHours-Morning.pcap_ISCX.csv	Friday	Benign, Bot
Friday-WorkingHours-Afternoon-PortScan.pcap_ISCX.csv	Friday	Benign, PortScan
Friday-WorkingHours-Afternoon-DDos.pcap_ISCX.csv	Friday	Benign, DDoS

It can be seen from Table 2 that the dataset contains attack information as five days traffic data. Thursday working hour afternoon and Friday data are well suited for binary classification. Similarly, Tuesday, Wednesday and Thursday morning data are best for designing the multi-class detection model. However, it should be noted that a best detection model should be able to detect attacks of any type. Therefore, to design such a typical IDS, the traffic data of all the day should be merged together to form a single dataset to be used by IDS. This is exactly followed to merge these files in this paper.

By merging the files presented in Table 2, found the whole shape of a dataset that contains 3119345 instances and 79 features containing 15 class labels (1 normal + 14 attack labels). Further, examining the instances of the combined files, it has been found that the dataset contains 288602 instances having the missing class label and 203 instances having missing information. By removing such missing instances, found a combined dataset of CICIDS-2017 having 2830540 instances. At this moment checked for any possible redundant instances. Surprisingly, no redundant instances were found. The characteristics of the combined dataset and the detailed class wise occurrence have been presented in the Table 3, Table 4 and Table 5 respectively.

Table 3: Records in the CICIDS-2017 Dataset

Type of Attack	Day	Total Records
Benign	Monday	529918
Brute Force Attack	Tuesday	445909
Heartbleed Attack/ DoS Attack	Wednesday	692703
Web Attack	Thursday (Morning)	170366
Infiltration Attack	Thursday(Afternoon)	288602
Botnet Attack	Friday(Morning)	191033
Port Scan Attack	Friday(Afternoon)	286467
DDoS Attack	Friday(Afternoon)	225745

Table 4: Overall characteristics of CICIDS-2017 dataset

Dataset Name	CICIDS2018
Dataset Type	Multi class
Year of release	2017
Total number of distinct instances(After removing missing instances)	2830540
Number of features	79
Number of distinct classes	15

Table 5: The Class wise instance occurrence of CICIDS-2017 dataset

Class Labels	Number of instances
BENIGN	2359087
DoS Hulk	231072
PortScan	158930
DDoS	41835
DoS GoldenEye	10293
FTP-Patator	7938
SSH-Patator	5897
DoS slowloris	5796
DoS Slowhttptest	5499
Botnet	1966
Web Attack – Brute Force	1507
Web Attack – XSS	652
Infiltration	36
Web Attack – Sql Injection	21
Heartbleed	11

Another interesting point we observed that the dataset fulfils all the criteria[30, 31] of a true intrusion detection dataset, such as complete network configuration, complete traffic, labeled dataset, complete interaction, complete capture, available protocols, attack diversity, heterogeneity, the feature set and meta data.

Table 6: Features of CICIDS-2017 Dataset

Feature no.	Features	Feature no.	Features
1.	Destination Port	41.	Packet Length Mean
2.	Flow Duration	42.	Packet Length Std
3.	Total Fwd Packets	43.	Packet Length Variance
4.	Total Backward Packets	44.	FIN Flag Count
5.	Total Length of Fwd Packets	45.	SYN Flag Count
6.	Total Length of Bwd Packets	46.	RST Flag Count
7.	Fwd Packet Length Max	47.	PSH Flag Count
8.	Fwd Packet Length Min	48.	ACK Flag Count
9.	Fwd Packet Length Mean	49.	URG Flag Count
10.	Fwd Packet Length Std	50.	CWE Flag Count
11.	Bwd Packet Length Max	51.	ECE Flag Count
12.	Bwd Packet Length Min	52.	Down/Up Ratio
13.	Bwd Packet Length Mean	53.	Average Packet Size
14.	Bwd Packet Length Std	54.	AvgFwd Segment Size
15.	Flow Bytes/s	55.	AvgBwd Segment Size
16.	Flow Packets/s	56.	Fwd Header Length
17.	Flow IAT Mean	57.	FwdAvg Bytes/Bulk
18.	Flow IAT Std	58.	FwdAvg Packets/Bulk
19.	Flow IAT Max	59.	FwdAvg Bulk Rate
20.	Flow IAT Min	60.	BwdAvg Bytes/Bulk
21.	Fwd IAT Total	61.	BwdAvg Packets/Bulk
22.	Fwd IAT Mean	62.	BwdAvg Bulk Rate
23.	Fwd IAT Std	63.	SubflowFwd Packets
24.	Fwd IAT Max	64.	SubflowFwd Bytes
25.	Fwd IAT Min	65.	SubflowBwd Packets
26.	Bwd IAT Total	66.	SubflowBwd Bytes
27.	Bwd IAT Mean	67.	Init_Win_bytes_forward
28.	Bwd IAT Std	68.	Init_Win_bytes_backward
29.	Bwd IAT Max	69.	act_data_pkt_fwd
30.	Bwd IAT Min	70.	min_seg_size_forward
31.	Fwd PSH Flags	71.	Active Mean
32.	Bwd PSH Flags	72.	Active Std
33.	Fwd URG Flags	73.	Active Max
34.	Bwd URG Flags	74.	Active Min
35.	Fwd Header Len	75.	Idle Mean
36.	Bwd Header Length	76.	Idle Std
37.	Fwd Packets/s	77.	Idle Max
38.	Bwd Packets/s	78.	Idle Min
39.	Min Packet Length	79.	Label
40.	Max Packet Length		

Table 7: Confusion Matrix

Type of Class	Predicted Class	
	Benign	Attack
Benign	True Positive	False Negative
Attack	False Positive	True Negative

IV. METHODOLOGY AND PROPOSED FRAMEWORK

A. PROPOSED FRAMEWORK

This work aims to propose an intrusion detection model in which the collected dataset will be analysed using machine learning algorithms. The key emphasis will be given on data pre-processing, feature selection, discretization, Machine learning algorithms and CICIDS-2017 Dataset.

Steps for the proposed model are as follows:

STEP 1: [Data Pre-processing]. This step adjusts the initial data format (CSV file).

STEP 2: [Feature Selection]. We select the optimal feature for classification. This reciprocal information based on feature selection algorithm can handle linearly and non-linearly dependent data features. Its effectiveness is analysed in instances of network intrusion detection.

STEP 3: [Analyse the results without discretization and with discretization]. In order to enhance the performance of machine learning algorithms, discretization converts, continuous data attribute values into the finite set of intervals with minimal loss of information. Then apply the machine learning algorithms on CICIDS-2017 dataset and analyse the performance of CICIDS-2017 dataset.

STEP 4: [Check Measures]. If the results are as per measures, consider the classification algorithms for further processing. For example, if the accuracy could satisfy the requirement (e.g. accuracy > 95%), consider the classification algorithms for further processing, otherwise, the model would repeat the STEP 3.

STEP 5: [Evaluation]. This step is used to evaluate the model performance. Evaluation metrics include the Time to build a model, Accuracy, Precision, Recall and F- Measure.

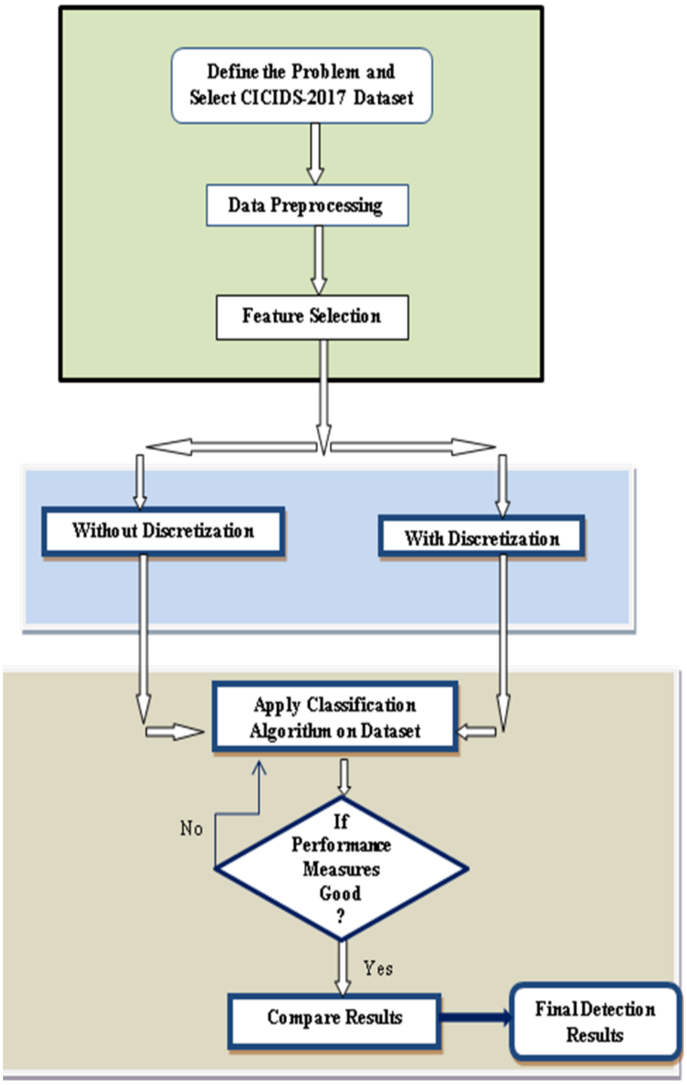
Define the Problem and Select CICIDS-2017 Dataset
Data Preprocessing
Feature Selection
Without Discretization
With Discretization
Apply Classification Algorithm on Dataset
No
If Performance Measures Good ?
Yes
Compare Results
Final Detection Results

Fig.1. Proposed Framework

B. Pre-Processing

Data pre-processing is preparing the data in a particular way or format that is suitable for actually doing the data mining algorithms. Data pre-processing is one of the major step of data mining i.e.

what has been done before the data mining.

In pre-processing sometime data is not always present in a single place. It might have to be gathered form different places and converted into a single & suitable format, followed by data cleaning and Normalization/ Discretization, before Data Mining.

C. Discretization

Discretization is a method for execution, which changes dataset record or properties for some machine learning calculations. The basic advantage of discretization is that some classification techniques can only work on ostensible properties, not on numerical characteristics. Further favourable position is that it will increment the order exactness of the tree and decide calculations that rely upon ostensible information.

D. Feature Selection Classification Algorithms

If one considers correlations between network traffic records to be linear associations, then a linear measure of dependence such as linear correlation coefficient can be used to measure the dependence between two random variables. However, considering the real world communication, the correlation between variables can be nonlinear as well. Apparently, a linear measure cannot reveal the relation between two nonlinearly dependent variables. Thus, measure the capability of analysing the relation between two variables no matter whether they are linearly or nonlinearly dependent. For these reasons, this work intends to explore a means of selecting optimal features from a feature space regardless of the type of correlation between them.

D.1 CFS (Correlation Feature Selection):

Feature selection is a process of selecting a subset of the relevant attribute selected from the large number of basic attributes of a particular dataset by applying unique assessment standards to enhance the pleasant of classification, while the dimension of the data reduces. It is used to evaluate the subset of features based on the well- suited subsets, which have highly correlated facilities with classification, are still unrelated to each other [32].

D.2 Classifier Subset Evaluator (CSE):

- It evaluates the specialty subset on training data or a separate hold-out test set.
- It uses a classification to estimate the eligibility of a set of attributes.

E. Classification Algorithms

E.1 Naive Bayes:

Naive Bayes Classifiers is a family of “probabilistic classification", which implements with the powerful independence hypothesis between the features [33]. It is particularly scalable; require a number of linear parameters inside the wide variety of irregular capabilities in a learning hassle. Naive Bayes is an easy technique for building classifiers: sample and pattern that gives magnificence labels to difficult situation time, which are shown as feature value vectors, where the labels are exhausted from some limited set. This is not the only algorithm for such classification, but the group of algorithm is entirely based on a general preaching. All Naive Bayes classifiers anticipate that the significance of a specific attribute is independent of the significance of any other attribute, given the class variable [34].

Bayes theorem is stated as the probability of the event B given A is equal to the probability of the event A given B multiplied by the probability of A upon the probability of B.

$$P(A/B) = P(B/A)\, P(A) / P(A)$$

Where A is called proposition, B is called evidence, P(A) is the prior probability of proposition, and P(B) is prior evidence of proposition.

E.2 J48:

The J48 is a type of C4.5 decision tree deployed for classification purposes in WEKA [35]. It has been produced by greedy top-down construction technique and uses reduced-error pruning. A detail hypothesis measure is used to determine attributes, which gives an indication of the classification power of each attribute. Once an attribute is selected, the training data are split into subsets, according to different values of the selected attribute and the procedure is repeated for each subset, unless a huge proportion of instances in each subset are related to a single class [36, 37].

E.3 Decision Tree:

Decision Tree is basically a technique or the data structure that builds or helps classification models in the form of a tree. It is a graphical representation of all the possible solution to a decision. Decisions are based on some condition and can be easily explained. There are two types of nodes: (i) Decision nodes, which specify the choice and test. (ii) Leaf nodes, indicates the classification of an example or the value of an example [38].

The decision table of algorithm is generated using classifiers under rules. Most commonly used algorithms in this category are DT, ID3, C4.5 (the extension of ID3) [35, 39] and CART [40].

V. PERFORMANCE EVALUATION

All classifiers are performed on the basis of Accuracy, Precision, Recall, F- Measure and Time to build a model. The performance was calculated by True Positive (TP), False Positive (FP), False Negative (FN) and True Negative (TN) (Table 7). All above values are calculated from the confusion matrices.

- Accuracy is defined as accurately prediction of positive and negative instances and is calculated as:
  Accuracy= (TP + TN)/ (TP+TN+FP+FN)
- Precision calculates the probability of accurate positive prediction.
  Precision = TP/ (TP + FP)
- The recall measures the number of correct classifications penalized by the number of missed entries.
  Recall= (TP/TP+FN)
- The harmonic mean of precision and recall calculates the F-measures, which serves as a derived effectiveness measurement.
  F- Measure= 2* (Precision * Recall)/ (Precision + Recall)
- Time to build is the time to build a model by a classification algorithm on the dataset.

VI. EXPERIMENTAL RESULTS

Our experiments were implemented under the following hardware and software platforms:

Hardware:

Intel Xeon E5-2650 v4 (12 core, 2.2 GHz, 30MB L3 cache, 16 GB RAM, NVIDIA Quadro P400 2GB)

Software:

Windows 10 Professional,
WEKA 3.8.2 (64bit).

To evaluate the system performance objectively, the following experiments were performed 10-fold for cross-validation on CICIDS-2017 dataset. CICIDS-2017 dataset has both benign and attack information data for network link. Each linked data is composed of 79 different attributes which are listed in the Table 6. In feature selection, the proposed methods are used for various matters of the selected feature.

We are checking the feature selection algorithms (i.e., correlation feature selection and classifier subset evaluator) and classification algorithms with feature subset algorithms using discretization and without discretization on the WEKA tool [41]. Table 8 expresses the predefined selected feature numbers for individual attacks and their numbers for each attack. Fig 2 shows that the number of selected feature when using different feature selection algorithm.

Table 8: Predefined selected feature numbers for individual attack using CfsSubset Attribute Evaluator, Classifier Subset Evaluator with Naive Bayes, Classifier Subset Evaluator with J48 Algorithms and Classifier Subset Evaluator with Decision Tree

Attacks	Algorithms	Predefined Selected Feature no.	Total Selected Features
Brute Force Attack	CFS	1,10,70	3
	CSE With Naive Bayes	1,10,18,38,49,67	6
	CSE With J48	1,35,38,49,67	5
	CSE With Decision Tree	68	1
DoS/ Heartbleed Attack	CFS	1,6,67,77	4
	CSE With Naive Bayes	1,2,6,14,24,35,41,67,68,70,74	11
	CSE With J48	1,3,6,7,20,24,40,67	8
	CSE With Decision Tree	1	1
Web Attack	CFS	9,25,68	3
	CSE With Naive Bayes	25,67,68	3
	CSE With J48	1,2,4,16,21,25,67,68	8
	CSE With Decision Tree	68	1
Infiltration Attack	CFS	5,72,74,76	4
	CSE With Naive Bayes	1,13,25,67,68,70	6
	CSE With J48	1,8,68	3
	CSE With Decision Tree	68	1
Botnet Attack	CFS	1, 13, 14, 70	4
	CSE With Naive Bayes	1, 6, 31, 35, 44, 67	6
	CSE With J48	1, 53, 67, 68	4
	CSE With Decision Tree	1	1
Port Scan Attack	CFS	13,47,68, 69, 70	5
	CSE With Naive Bayes	5,7,8,10,23,41,47,67,68	9
	CSE With J48	25,35,38,41,68,71	6
	CSE With Decision Tree	68	1
DDoS Attack	CFS	1, 7, 46, 72	4
	CSE With Naive Bayes	5, 8, 26, 67, 68, 75, 77	7
	CSE With J48	1, 5, 6, 8, 48, 67, 68, 78	8
	CSE With Decision Tree	1	1

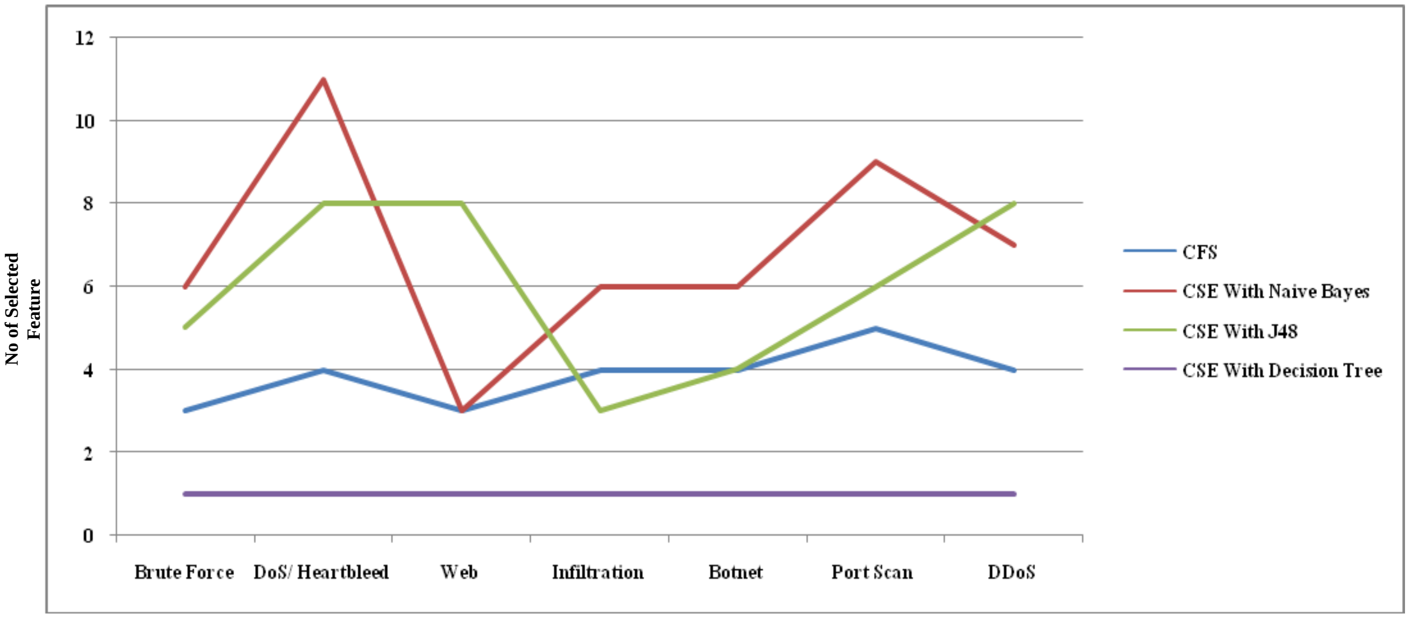

Fig.2. Predefined selected feature numbers for individual attack using Feature Selection Algorithms

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

Table 9: Performance evaluation for *Brute Force* Attack using classifier with the selected feature

Classifier	Features Selection Algorithm	Without Discretization					With Discretization				
		Time	Accuracy	Precision	Recall	F-Measure	Time	Accuracy	Precision	Recall	F-Measure
Naïve Bayes	CFS	0.39	99.2696	1.000	0.993	0.996	0.09	99.8058	1.000	0.998	0.999
	CSE With Naive Bayes	0.48	99.2770	1.000	0.993	0.996	0.05	99.9899	1.000	0.993	0.996
	CSE With J48	0.52	98.5596	1.000	0.993	0.996	0.06	99.9670	1.000	1.000	1.000
	CSE With Decision Tree	0.11	90.5985	0.990	0.921	0.995	0.04	98.8670	0.990	0.998	0.994
J48	CFS	1.14	99.8706	1.000	0.999	0.999	0.27	99.8695	1.000	0.999	0.999
	CSE With Naive Bayes	2.84	99.9951	1.000	1.000	1.000	0.04	99.9937	1.000	1.000	1.000
	CSE With J48	2.00	99.9919	1.000	1.000	1.000	0.39	99.9973	1.000	1.000	1.000
	CSE With Decision Tree	1.05	98.8670	0.990	0.998	0.994	0.16	98.8670	0.990	0.998	0.994
Decision Tree	CFS	0.31	97.8899	1.000	0.992	0.996	0.09	97.8899	1.000	0.992	0.996
	CSE With Naive Bayes	0.98	97.8899	1.000	0.992	0.996	0.11	97.8899	1.000	0.992	0.996
	CSE With J48	0.73	97.8899	1.000	0.992	0.996	1.50	97.8899	1.000	0.992	0.996
	CSE With Decision Tree	0.13	96.8973	0.969	1.000	0.984	0.03	98.1140	0.981	1.000	0.990

Table 10: Performance evaluation for *Heartbleed Attack/ DoS Attack* using classifier with the selected features

Classifier	Features Selection Algorithm	Without Discretization					With Discretization				
		Time	Accuracy	Precision	Recall	F-Measure	Time	Accuracy	Precision	Recall	F-Measure
Naïve Bayes	CFS	0.59	88.8262	1.000	0.889	0.941	0.11	98.7126	0.999	0.987	0.993
	CSE With Naive Bayes	1.24	90.3751	1.000	0.899	0.947	0.11	99.8145	0.999	0.999	0.999
	CSE With J48	1.84	91.3872	1.000	0.893	0.943	0.04	99.2516	0.999	0.990	0.994
	CSE With Decision Tree	0.14	89.8427	1.000	0.889	0.941	0.09	89.8427	1.000	0.889	0.941
J48	CFS	8.91	99.4422	0.995	0.998	0.997	0.47	99.4347	0.995	0.998	0.997
	CSE With Naive Bayes	29.60	99.9554	1.000	1.000	1.000	0.94	99.9373	1.000	1.000	1.000
	CSE With J48	34.58	99.9568	1.000	1.000	1.000	0.86	99.9460	1.000	1.000	1.000
	CSE With Decision Tree	0.59	89.8427	1.000	0.889	0.941	0.14	89.8427	1.000	0.889	0.941
Decision Tree	CFS	0.73	83.5489	0.821	0.963	0.886	0.09	89.8427	1.000	.889	0.941
	CSE With Naive Bayes	3.02	85.7092	0.828	0.999	0.905	0.15	89.8427	0.828	0.999	0.905
	CSE With J48	1.95	84.3523	0.821	0.980	0.893	0.23	89.8427	1.000	0.889	0.941
	CSE With Decision Tree	0.11	63.5238	0.635	1.000	0.777	0.06	89.8427	1.000	0.889	0.941

Table 11: Performance evaluation for *Web Attack* using classifier with the selected features

Classifier	Features Selection Algorithm	Without Discretization					With Discretization				
		Time	Accuracy	Precision	Recall	F-Measure	Time	Accuracy	Precision	Recall	F-Measure
Naïve Bayes	CFS	0.20	13.4264	0.996	0.130	0.230	0.03	99.4981	0.999	0.999	0.999
	CSE With Naive Bayes	0.08	83.4996	0.999	0.842	0.914	0.01	99.4066	1.000	0.998	0.999
	CSE With J48	0.17	94.9614	1.000	0.954	0.976	0.02	98.9094	1.000	0.995	0.997
	CSE With Decision Tree	0.03	96.4165	0.987	0.977	0.982	0.02	99.2469	0.999	0.997	0.998
J48	CFS	1.01	99.5833	0.999	1.000	1.000	0.11	99.5821	0.999	1.000	1.000
	CSE With Naive Bayes	0.77	99.5985	1.000	1.000	1.000	0.08	99.5985	1.000	1.000	1.000
	CSE With J48	2.86	99.6138	1.000	1.000	1.000	0.09	99.6243	1.000	1.000	1.000
	CSE With Decision Tree	0.20	99.2463	0.999	0.997	0.998	0.03	99.2469	0.999	0.997	0.998
Decision Tree	CFS	0.16	98.7204	0.987	1.000	0.994	0.02	99.1319	0.998	0.997	0.997
	CSE With Naive Bayes	0.09	98.7204	0.987	1.000	0.994	0.01	99.1319	0.998	0.997	0.997
	CSE With J48	0.41	98.7204	0.987	1.000	0.994	0.05	99.1319	0.998	0.997	0.997
	CSE With Decision Tree	0.04	98.7204	0.987	1.000	0.994	0.02	99.1319	0.998	0.997	0.997

Table 12: Performance evaluation for Infiltration Attack using classifier with the selected features

Classifier	Features Selection Algorithm	Without Discretization					With Discretization				
		Time	Accuracy	Precision	Recall	F-Measure	Time	Accuracy	Precision	Recall	F-Measure
Naïve Bayes	CFS	0.30	99.0440	1.000	0.990	0.995	0.08	99.9598	1.000	0.990	0.995
	CSE With Naive Bayes	0.30	97.5603	1.000	0.976	0.988	0.01	99.9993	1.000	0.976	0.988
	CSE With J48	0.17	84.9606	1.000	0.850	0.919	0.02	99.9965	1.000	0.850	0.919
	CSE With Decision Tree	0.08	99.9875	1.000	1.000	1.000	0.02	99.9938	1.000	1.000	1.000
J48	CFS	1.61	99.9900	1.000	1.000	1.000	0.25	99.9913	1.000	1.000	1.000
	CSE With Naive Bayes	2.03	99.9958	1.000	1.000	1.000	0.20	99.9945	1.000	1.000	1.000
	CSE With J48	0.67	99.9955	1.000	1.000	1.000	0.19	99.9972	1.000	1.000	1.000
	CSE With Decision Tree	0.17	99.9931	1.000	1.000	1.000	0.03	99.9938	1.000	1.000	1.000
Decision Tree	CFS	0.27	99.9875	1.000	1.000	1.000	0.11	99.9875	1.000	1.000	1.000
	CSE With Naive Bayes	0.53	99.9875	1.000	1.000	1.000	0.05	99.9875	1.000	1.000	1.000
	CSE With J48	0.20	99.9875	1.000	1.000	1.000	0.03	99.9875	1.000	1.000	1.000
	CSE With Decision Tree	0.05	99.9875	1.000	1.000	1.000	0.05	99.9900	1.000	1.000	1.000

Table 13: Performance evaluation for *Botnet Attack* using classifier with the selected features

Classifier	Features Selection Algorithm	Without Discretization					With Discretization				
		Time	Accuracy	Precision	Recall	F-Measure	Time	Accuracy	Precision	Recall	F-Measure
Naïve Bayes	CFS	0.22	86.7180	0.993	0.872	0.929	0.06	99.5095	0.998	0.997	0.998
	CSE With Naive Bayes	0.16	14.899	1.000	0.140	0.246	0.01	99.9717	1.000	1.000	1.000
	CSE With J48	0.14	98.4704	0.990	0.995	0.992	0.03	99.9707	1.000	1.000	1.000
	CSE With Decision Tree	0.03	98.9709	0.990	1.000	0.995	0.02	99.6304	0.996	1.000	0.998
J48	CFS	0.80	99.6357	0.997	0.999	0.998	0.09	99.6362	0.997	1.000	0.998
	CSE With Naive Bayes	0.64	99.9639	1.000	1.000	1.000	0.05	99.9602	1.000	1.000	1.000
	CSE With J48	0.55	99.9817	1.000	1.000	1.000	0.05	99.9628	1.000	1.000	1.000
	CSE With Decision Tree	0.17	99.6283	0.996	1.000	0.998	0.03	99.6304	0.996	1.000	0.998
Decision Tree	CFS	0.20	98.9709	0.990	1.000	0.995	0.06	99.6310	0.996	1.000	0.998
	CSE With Naive Bayes	0.27	98.9709	0.990	1.000	0.995	0.02	99.6310	0.996	1.000	0.998
	CSE With J48	0.16	98.9709	0.990	1.000	0.995	0.02	99.6310	0.996	1.000	0.998
	CSE With Decision Tree	0.04	98.9709	0.990	1.000	0.995	0.02	99.6310	0.996	1.000	0.998

Table 14: Performance evaluation for *PortScan Attack* using classifier with the selected features

Classifier	Features Selection Algorithm	Without Discretization					With Discretization				
		Time	Accuracy	Precision	Recall	F-Measure	Time	Accuracy	Precision	Recall	F-Measure
Naïve Bayes	CFS	0.28	97.0974	0.991	0.943	0.967	0.06	99.3165	0.991	0.994	0.992
	CSE With Naive Bayes	0.31	90.6733	0.989	0.799	0.884	0.03	99.9417	0.999	0.999	0.999
	CSE With J48	0.25	69.674	0.968	0.330	0.492	0.03	99.5839	0.992	0.999	0.995
	CSE With Decision Tree	0.06	58.5160	0.909	0.076	0.140	0.02	98.4243	0.994	0.971	0.982
J48	CFS	2.55	99.8265	0.999	0.997	0.998	0.28	99.8391	0.999	0.997	0.998
	CSE With Naive Bayes	5.30	99.9623	1.000	0.999	1.000	0.39	99.9623	1.000	0.999	1.000
	CSE With J48	4.22	99.9853	1.000	1.000	1.000	0.30	99.9822	1.000	1.000	1.000
	CSE With Decision Tree	0.42	98.4239	0.994	0.971	0.982	0.05	98.4243	0.994	0.971	0.982
Decision Tree	CFS	0.31	89.3625	0.999	0.762	0.864	0.06	98.3680	0.990	0.973	0.982
	CSE With Naive Bayes	0.55	95.7744	0.995	0.910	0.950	0.06	98.3680	0.990	0.973	0.982
	CSE With J48	0.38	95.0870	0.999	0.891	0.942	0.05	98.3680	0.990	0.973	0.982
	CSE With Decision Tree	0.04	82.2936	0.999	0.603	0.752	0.03	98.3680	0.990	0.973	0.982

Table 15: Performance evaluation for *DDoS Attack* using classifier with the selected features

Classifier	Features Selection Algorithm	Without Discretization					With Discretization				
		Time	Accuracy	Precision	Recall	F-Measure	Time	Accuracy	Precision	Recall	F-Measure
Naïve Bayes	CFS	0.19	84.6030	1.000	0.644	0.784	0.06	98.9125	1.000	0.975	0.987
	CSE With Naive Bayes	0.20	86.9990	0.999	0.700	0.823	0.02	99.9699	0.999	1.000	1.000
	CSE With J48	0.22	92.1199	1.000	0.818	0.900	0.03	99.7572	1.000	0.995	0.997
	CSE With Decision Tree	0.05	75.0285	1.000	0.423	0.595	0.02	96.0442	1.000	0.909	0.952
J48	CFS	1.58	98.9125	1.000	0.975	0.987	0.20	98.9125	1.000	0.975	0.987
	CSE With Naive Bayes	2.83	99.9553	0.999	1.000	0.999	0.47	99.9685	0.999	1.000	1.000
	CSE With J48	3.08	99.9836	1.000	1.000	1.000	0.23	99.9849	1.000	1.000	1.000
	CSE With Decision Tree	0.14	96.0442	1.000	0.909	0.952	0.03	96.0442	1.000	0.909	0.952
Decision Tree	CFS	0.16	87.1519	1.000	0.703	0.826	0.06	96.0442	1.000	0.909	0.952
	CSE With Naive Bayes	0.30	85.9851	0.999	0.677	0.807	0.05	79.2049	0.676	0.996	0.806
	CSE With J48	0.36	85.9851	0.999	0.677	0.807	0.03	96.0442	1.000	0.909	0.952
	CSE With Decision Tree	0.03	81.6403	1.000	0.576	0.731	0.01	96.0442	1.000	0.909	0.952

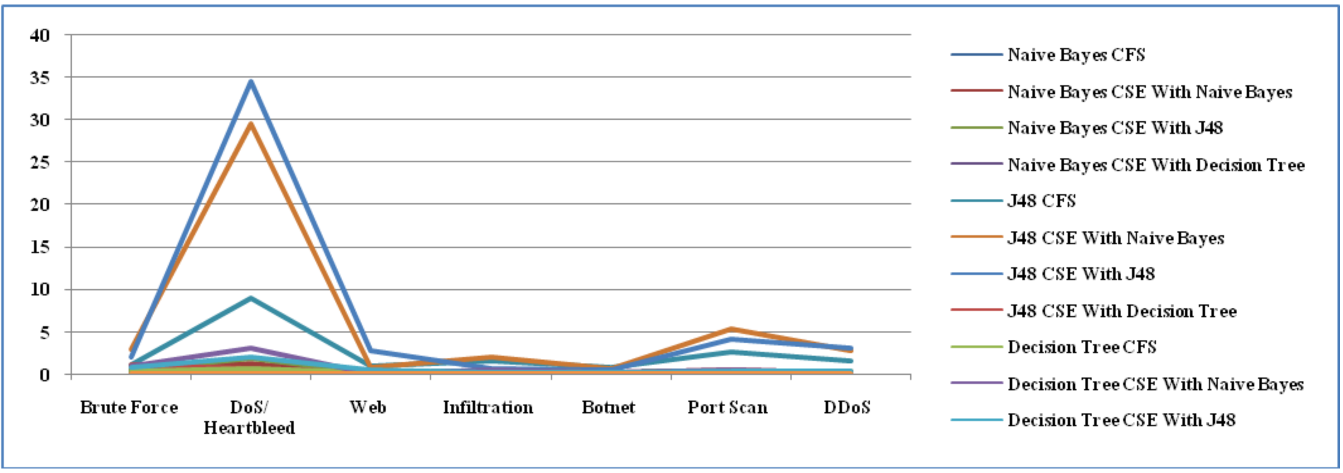

Fig.3. Time Comparison without Discretization

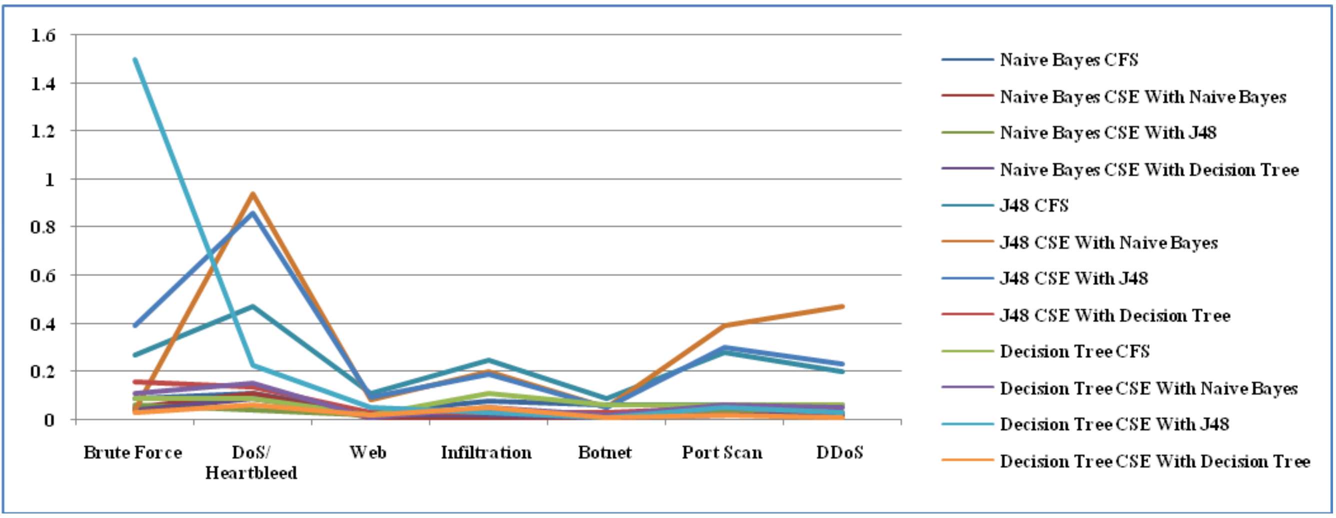

Fig.4. Time Comparison with Discretization

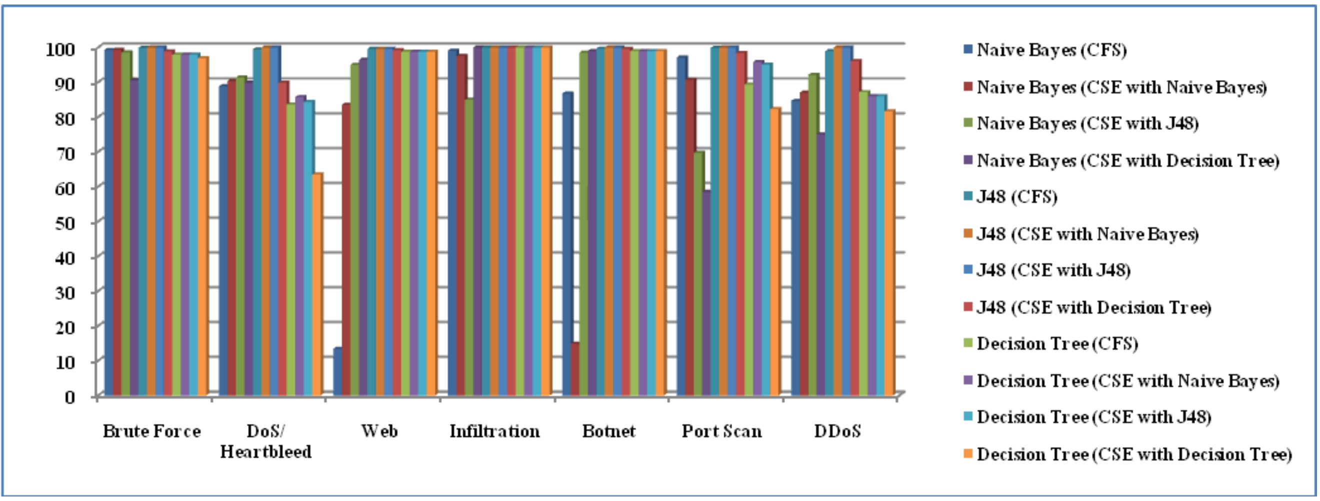

Fig.5. Accuracy Comparison without Discretization

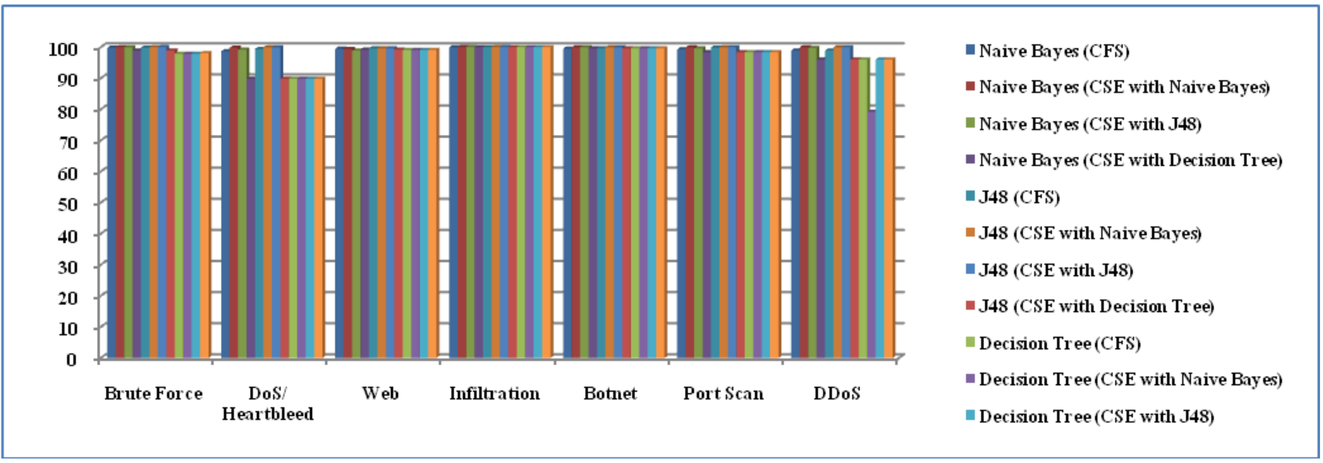

Fig.6. Accuracy Comparison with Discretization

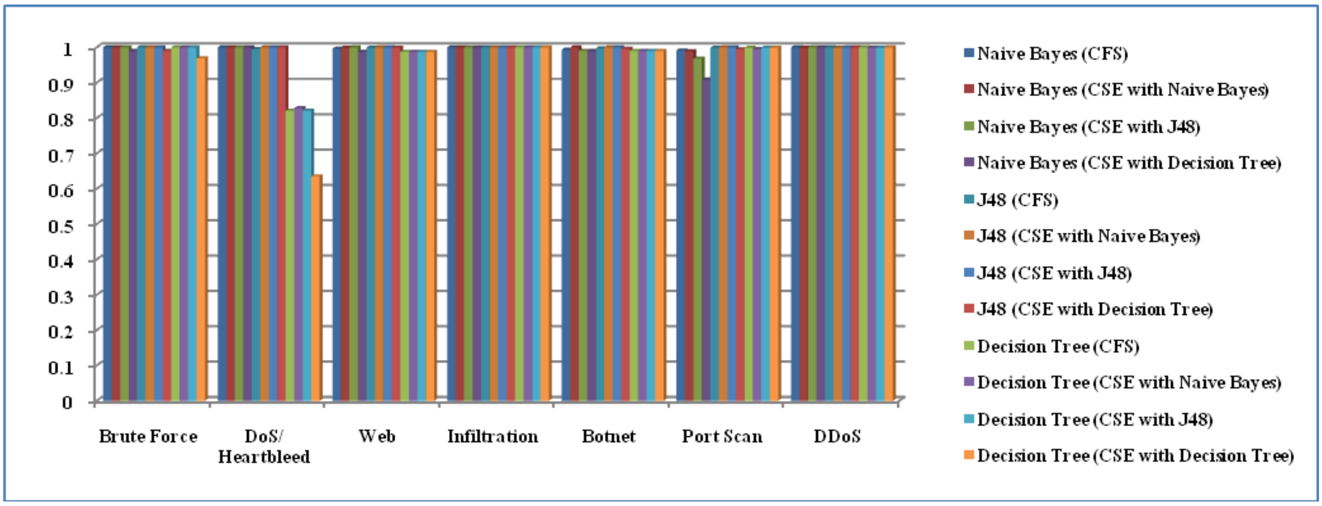

Fig.7.Precision Comparison without Discretization

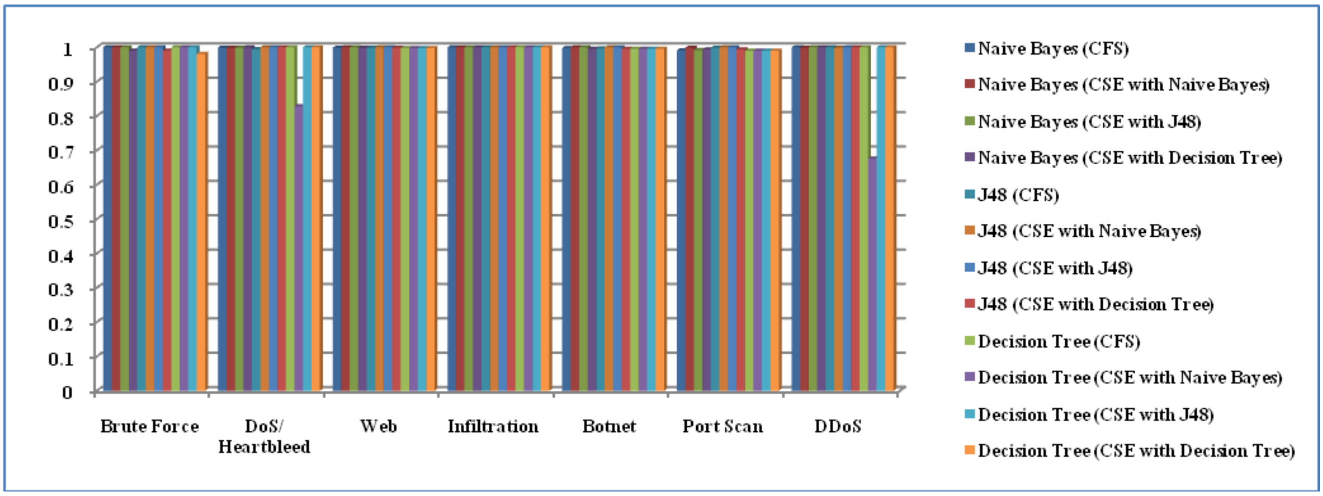

Fig.8.Precision Comparison with Discretization

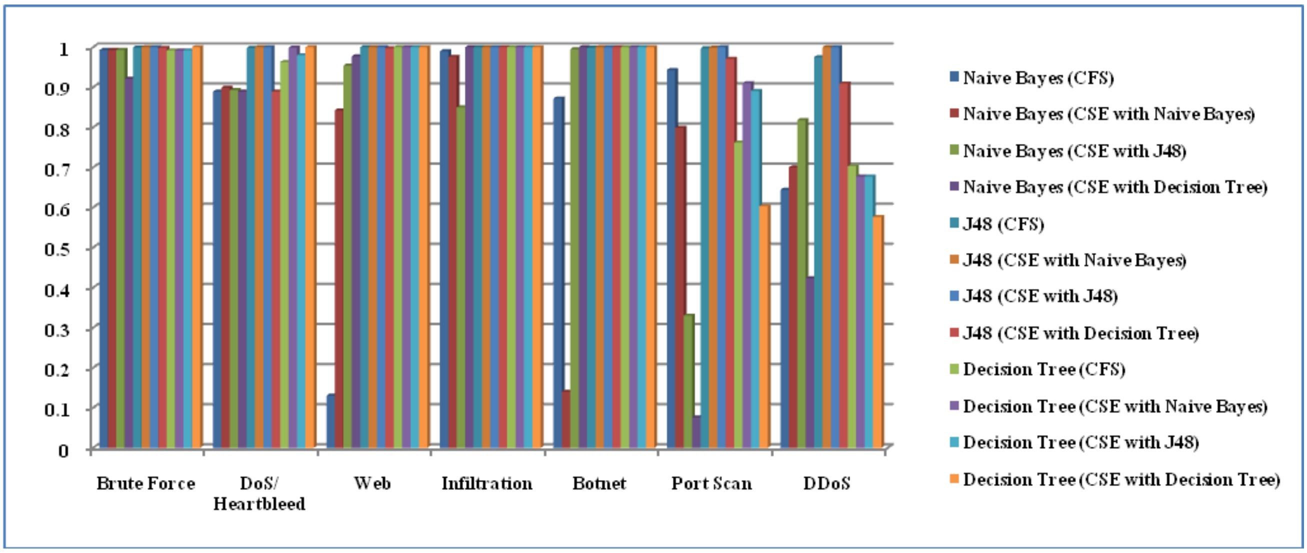

Fig.9.Recall Comparison without Discretization

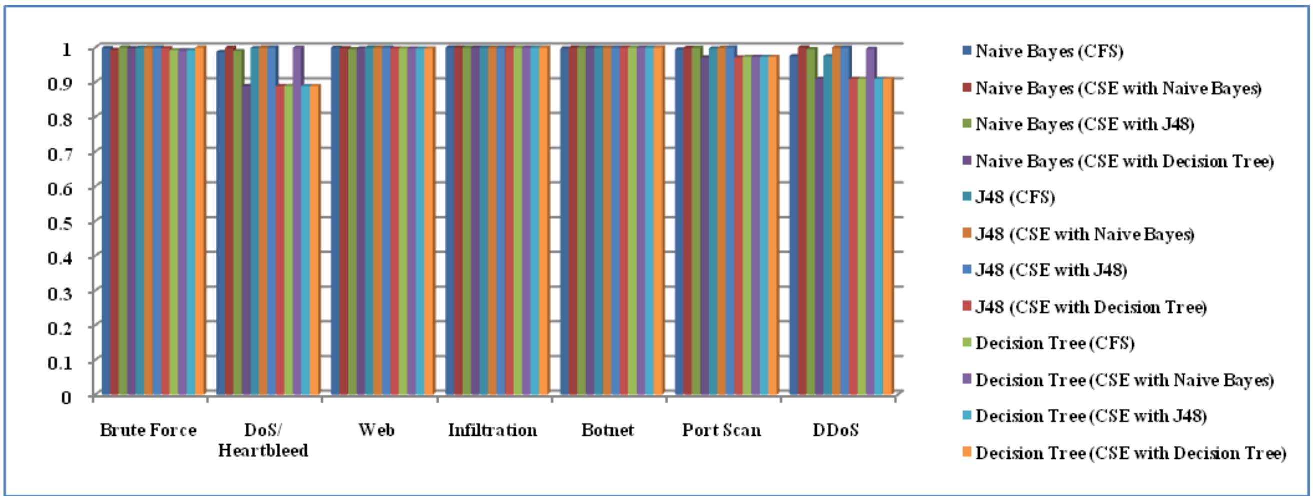

Fig.10.Recall Comparison with Discretization

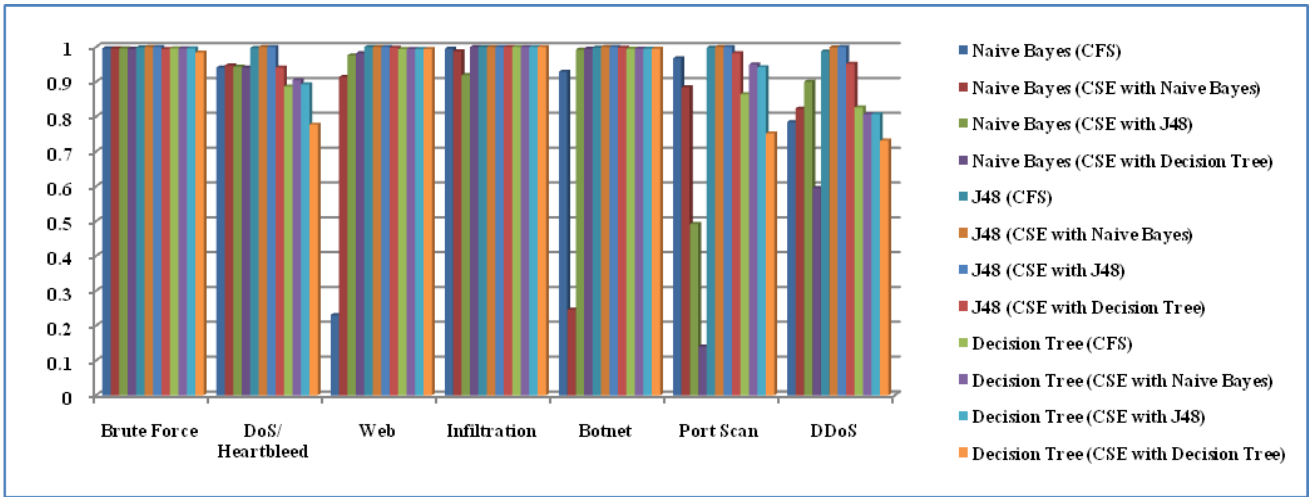

Fig.11. F- Measure Comparison without Discretization

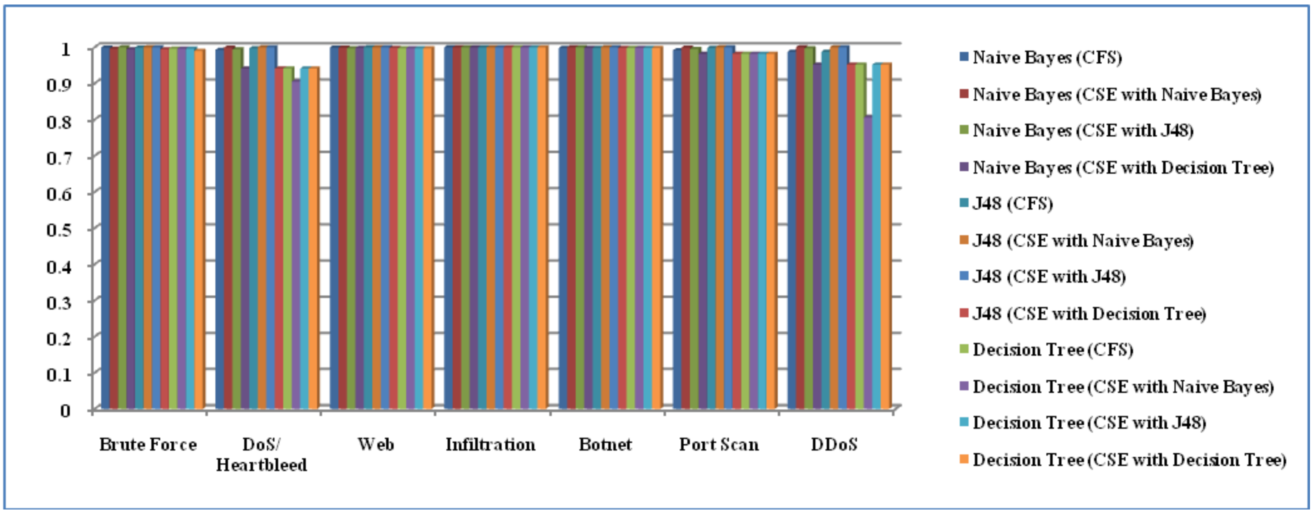

Fig.12. F- Measure Comparison with Discretization

We have applied three separate classified algorithms (i.e. Naïve Bayes, J48 and Decision Tree) selected from the WEKA-Tool using discretization and without discretization. Then compared the performance in terms of accuracy, precision, recall, F1-measure and time taken to build a model. Table 9, 10, 11, 12, 13, 14 and 15 and fig 3, 4, 5, 6, 7, 8, 9, 10, 11, 12 shows the results obtained for both with discretization and without discretization.

Comparison in different tables exhibits the usage of the selected classification algorithms with entire features and the predefined selected feature for the individual attack by CFS and Classifier Subset Evaluator with Naive Bayes, J48 and Decision Tree are given as following:

Accuracy analysis from different tables and Fig 5, 6 shows that:

- Without discretization the *most accurate* classifiers are J48 classifier using CSE with Naive Bayes features selection technique for Brute Force Attack and the J48 classifier using CSE with J48 features selection technique for Web attack, Heartbleed Attack/ DoS, Infiltration, Port Scan, Botnet and DDoS Attack. (CSE :- Classifier Subset Evaluator)
- With discretization, the J48 classifier using CSE with J48 features selection technique for Brute Force, Heartbleed Attack/ DoS, Web, Port Scan and DDoS and Naive Bayes using CSE with Naive Bayes features selection technique for Botnet and Infiltration Attack provides the *best performance* in the term of accuracy.

It is clear from various tables and Fig.3, 4 that time to build a model without discretization is:

- Naive Bayes classifier using CSE with Decision Tree techniques for Brute Force, Port Scan and Botnet Attack, Decision Tree classifier using CSE with Decision Tree techniques for DoS and Infiltration Attack, Naive Bayes classifier using CSE with Naive Bayes techniques for Web Attack and the J48 classifier using CSE with Naive Bayes for DDOS attack *takes less time* to build a model.

It is also clear that:

- With discretization Decision Tree classifier using CSE with Decision Tree techniques for Brute Force, DoS, Port Scan and DDoS Attack, Decision Tree classifier using CSE with Naive Bayes techniques for Web Attack and Naive Bayes classifier using CSE with Naive Bayes techniques for Infiltration and Botnet Attack *takes less time* to build a model.

Interpretation of precision from different tables and Fig 7, 8 indicates that:

- Without discretization, the J48 classifier using CSE with J48 and Naive Bayes techniques for Port Scan and Web Attack, all classifier for Brut Force Attack, the J48 and Naive Bayes classifier for DoS Attack, J48 and Naive Bayes Classifier using CSE with Naive Bayes and CSE with J48 techniques for Botnet Attack, all classifier for Infiltration Attack and Decision Tree classifier using CSE with Decision Tree techniques for DDoS Attack provides the *best performance* in the term of precision.

- With discretization, Most of classifier has the same precision for Brute Force Attack, the J48 and Naive Bayes Classifier using CSE with Naive Bayes and CSE with J48 techniques for Web and Botnet Attack, all classifier for Infiltration and DDoS Attack, J48 classifier for DoS Attack, J48 and Decision Tree classifier using CSE with Naive Bayes and Decision Tree techniques for Port Scan Attack provides the *best performance* in the term of precision.

Recall from different tables and Fig 9, 10 shows that:

- Without discretization, the J48 classifier using CSE with J48 and Naive Bayes techniques for Brute Force and DoS Attack, Decision Tree and the J48 classifier for Web, Infiltration, Botnet and Web Attack, the J48 classifier using CSE with J48 techniques for Port Scan Attack and All classifier for DDoS Attack provide the *best performance* in the term of Recall.
- With discretization, the J48 classifier using CSE with J48 and Naive Bayes techniques for Brute Force and DoS Attack, the J48 classifier for Web Attack, the J48 and Decision Tree classifier for Infiltration Attack, all classifier for Botnet Attack, the J48 and Naive Bayes classifier using CSE with naive Bayes classifier for DDoS Attack and J48 classifier using CSE with Naive Bayes techniques for DoS Attack provide the *best performance* in the term of Recall.

F- Measure calculation from various tables and Fig 11, 12 indicates that:

- Without discretization, the J48 classifier using CSE with J48 and Naive Bayes techniques for Brute Force, DoS, Botnet and Port Scan Attack, the J48 classifier for Web Attack, J48 and Decision Tree for Infiltration Attack and the J48 classifier using CSE with J48 technique for DDoS Attack provide the *best performance* in the term of F- Measure.
- With discretization, the J48 classifier using CSE with J48 and Naive Bayes techniques for Brute Force, DoS, Botnet and Web Attack, J48 and Decision Tree for Infiltration Attack, the J48 classifier using CSE with J48 technique for Port Scan and the J48 classifier using CSE with Naive Bayes techniques for DDoS Attack provide the *best performance* in the term of F- Measure.

VII. CONCLUSION

In this paper, we have examined various classification algorithms and feature selection algorithms with discretization and without discretization for detecting network intrusion and anomaly with the help of WEKA on CICIDS-2017 dataset and have shown the best performance in terms of accuracy, precision, Recall, F-measure and time to build a model. Simulation results shows that the use of feature selection algorithms with discretization reduces the dimension of dataset, time to build a model, false alarms and induces high performance results.

The results also show that classifier with filtered supervised discretization can decrease time taken by algorithms and increase the predicted accuracy, F- Measure, recall and Precision (Table 9 to 15). It also shows that the filtering supervised discretization has a larger effect in the execution of the classification algorithms.

In the future work, we will consider the idea of advanced machine learning, deep learning and hybrid algorithms to detect network intrusion and anomaly with WEKA on CICIDS-2017 dataset to achieve a higher level of performance.

REFERENCES

1. Lee, W., Stolfo, S. J., Mok, K. W.: ‘A Framework for Constructing Features and Models for Intrusion Detection Systems’, ACM Transaction on Information and System Security, Nov. 2000, 3, (4), pp 227-261.
2. Xin, Y.,Kong,L., Liu, Z., et al.: ‘Machine Learning and Deep Learning Methods for Cyber security’, IEEE Access, 2017, 6, pp 35365-35381.
3. Zhu, D., Kumar, P., Zhang, X., Chu, C. H.: ‘Data Mining for Network Intrusion Detection: A Comparison of Alternative Methods’, Decision Sciences, June 2001, 32, (4), pp 635-660.
4. Kim, T., Yeo,S. S., Liu,Z.,Lai, Y.: ‘A Data Mining Framework for Building Intrusion Detection Models Based on IPv6’, Advances in Information Security and Assurance, Springer, Berlin, Heidelberg, 2009, 5576, pp 608-618.
5. WEKA User Manual, [Online], Available:www.gtbit.org/ downloads/dwdmsem6/dwdmsem6lm an.pdf, 2013.
6. Belavagi, M.C., Muniyal, B.: ‘Performance evaluation of supervised machine learning algorithms for intrusion detection’, Proc. Twelfth Int. Multi-Conf. on Information Processing, Bangalore, India, Dec. 2016, pp. 117–123.
7. Yin, C.L., Zhu, Y.F., Fei, J.L., et al.: ‘A deep learning approach for intrusion detection using recurrent neural networks’, IEEE. Access, Oct. 2017, 5, pp. 21954–21961.
8. Alrawashdeh, K., Purdy, C.: ‘Toward an online anomaly intrusion detection system based on deep learning’, IEEE Int. Conf. Machine Learning and Applications, Anaheim, CA, USA, 2017, pp 195–200.
9. Zhao, G., Zhang, C., Zheng, L.: ‘Intrusion detection using deep belief network and probabilistic neural network’, 2017 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC), Guangzhou, 2017, pp 639-642.
10. Roy, S.S., Mallik, A., Gulati, R., et al.: ‘A deep learning based artificial neural network approach for intrusion detection’, International Conference on Mathematics and Computing, April 2017, pp 44–53.
11. Xu. C, Shen, J., Du, X.,Zhang, F.: ‘An Intrusion Detection System Using a Deep Neural Network With Gated Recurrent Units’, *IEEE Access*, August 2018, 6, pp 48697-48707.
12. Wu K., Chen, Z., Li, W.: ‘A Novel Intrusion Detection Model for a Massive Network Using Convolutional Neural Networks’, IEEE Access, Sept. 2018, 6, pp. 50850-50859.
13. Wang, Z.:‘deep learning-based intrusion detection with adversaries’, Challenges and Opportunities of Big Data Against Cyber Crime (IEEE Access), July 2018, 6, pp 38367 – 38384.
14. Shone, N., Ngoc, T. N., Phai, V. D., Shi, Q.: ‘A Deep Learning Approach to Network Intrusion Detection’, IEEE Transactions on Emerging Topics in Computational Intelligence, Jan. 2018, 2, (1), pp 41-50.
15. Li, L., Yu, Y., Bai, S., Hou, Y., Chen, X.:‘An Effective Two-Step Intrusion Detection Approach Based on Binary Classification and k-NN’, Dec. 2018, *IEEE Access, 6*, pp 12060-12073.
16. Kamarudin, M.H., Maple, C., Watson, T., Safa, N. S.: ‘A LogitBoost-Based Algorithm for Detecting Known and Unknown Web Attacks’, *IEEE Access*, Nov. 2017, 5, pp 26190-26200.
17. Jia, Y., Wang, M., Wang, Y.: ‘Network intrusion detection algorithm based on deep neural network’, *IET* Information Security, Jan. 2019, 13, (1), pp 48-53.
18. Gogoi, P., Bhattacharyya, D. K., Jugal, B. B.,Kalita, K.: ‘MLH-IDS: A Multi-Level Hybrid Intrusion Detection Method’, the Computer Journal, April 2014, 57, (4), pp 602–623.

19. Gao, Y., Liu, Y., Jin, Y., Chen, J., Wu, H.: ‘A Novel Semi-Supervised Learning Approach for Network Intrusion Detection on Cloud-Based Robotic System’, *IEEE Access*, Sept. 2018, 6, pp50927-50938.
20. Ambusaidi, M. A., He, X., Nanda, P., Tan, Z.: ‘Building an Intrusion Detection System Using a Filter-Based Feature Selection Algorithm’, IEEE Transactions on Computers, Oct. 2016, 65, (10), pp 2986-2998.
21. Latah M., and Toker, L.: ‘Towards efficient anomaly-based intrusion detection for software-defined networks’, IET Networks, Nov. 2018, 7, (6), pp 453-459.
22. Panwar, S. S., Raiwani, Y. P.: ‘Data Reduction Techniques to Analyze NSL-KDD Dataset’, International Journal of Computer Engineering &Technology’, 2014, 5,(10), pp 21-31.
23. Raiwani, Y. P., Panwar, S. S.: ‘Research Challenges and Performance of Clustering Techniques to Analyze NSL-KDD Dataset’, International Journal of Emerging Trends & Technology in Computer Science (IJETTCS), 2014, 3, (6), pp 172-177.
24. Raiwani, Y. P., Panwar, S. S.: ‘Data Reduction and Neural Networking Algorithms to Improve Intrusion Detection System with NSL-KDD Dataset’, International Journal of Emerging Trends & Technology in Computer Science (IJETTCS), 2015, 4, (1),pp 219-225.
25. Panwar, S. S., Raiwani, Y. P., Panwar, L. S.: ‘Evaluation of Network Intrusion Detection with Features Selection and Machine Learning Algorithms on CICIDS-2017 Dataset’, International Conference on Advanced Engineering, Science, Management and Technology (ICAESMT-19), March 2019.
26. Baluni, P., Raiwani, Y.P.: ‘Vehicular Accident Analysis Using Neural Network’, International journal of Emerging Technologies and advanced Engineering (IJETAE), 2014, Vol.4 (9), pp. 161-164.
27. Radford, B. J., Richardson, B.D., Davis, S. E.: ‘Sequence Aggregation Rules for Anomaly Detection in Computer Network Traffic’, American Statistical Association 2018 Symposium on Data Science and Statistics, May 2018, pp. 1-**13.**
28. Vijayanand, R., Devaraj, D., Kannapiran, B.: ‘Intrusion detection system for wireless mesh network using multiple support vector machine classifiers with genetic-algorithm-based feature selection. Computers & Security’, Computers & Security, 2018, 77, pp 304-314.
29. Nicholas, L., Ooi, S. Y., Pang, Y. H., Hwang, S.O., Tan, S.: ‘Study of long short-term memory in flow-based network intrusion detection system’, Journal of Intelligent & Fuzzy Systems, Dec. 2018, 35, (6), pp 1-11, 2018.
30. Gharib, A., Sharafaldin, I., Lashkari, A. H., Ghorbani, A. A.: ‘An Evaluation Framework for Intrusion Detection Dataset’, International Conference on Information Science and Security (ICISS), Pattaya (Thailand), March 2016, pp 1-6.
31. Sharafaldin, I., Lashkari, A. H., Ghorbani, A. A.: ‘Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization’, 4th International Conference on In-formation Systems Security and Privacy (ICISSP), Purtogal, Jan. 2018, pp 108-116.
32. Aher, S. B., LOBO, Mr.: ‘Data Mining in Educational System using WEKA’ International Conference on Emerging Technology Trends (ICETT’11), 2019, pp 20-25.
33. Bakshi K.,Bakshi, K.: ‘Considerations for Artificial Intelligence and Machine Learning: Approaches and Use Cases, *IEEE Aerospace Conference*, Big Sky, MT, 2018, pp 1-9.
34. John, G. H., Langley, P.: ‘Estimating continuous distributions in Bayesian classifiers’, Int. Conf. Uncertainty in artificial intelligence, San Mateo, Feb. 2013, pp. 338–345.
35. Quinlan, J. R.: ‘C4.5: Programs for Machine Learning’, Morgan Kaufmann Publishers Inc. San Francisco, CA, USA, 1993.
36. WEKA Data Mining Machine Learning Software [Online]Availablehttp://www.cs.waikato.ac.nz/ml/weka/
37. Garner, S.: ‘WEKA: The Waikato environment for knowledge’, Available:https://www.cs.waikato.ac.nz/~ml/publications/1995/Garner95-WEKA.pdf, 1995.
38. Burges, C., ‘A tutorial on support vector machines for pattern recognition’, Data Min Knowl Disc, June 1998, 2, (2), pp 1–47.
39. Breiman, L., Friedman, J. H., Olshen, R. A., Stone, C. J.: ‘Classification and regression trees’, Wadsworth and Brooks/ Cole Advanced Books & Software, Monterey, April 1984.
40. Murthy, S. K.: ‘Automatic construction of decision trees from data: A multi-disciplinary survey’, Data Mining and Knowledge Discovery, Dec. 1998, 2, (4), pp 345–389.
41. Garge, T., Kumar, Y.: ‘Combinational feature selection approach for network intrusion detection system’, International *Conference on Parallel, Distributed and Grid Computing*, Solan, 2014, pp 82-87.

AUTHORS PROFILE

Shailesh Singh Panwar received his B.Tech. degree from Uttarakhand Technical University Dehradun, Uttarakhand, India and M.Tech. degree from Graphic Era University Dehradun India. He has Four Years of teaching experience as a guest faculty in the Department of Computer Science and Engineering, H.N.B. Garhwal University (A Central University) Srinagar Garhwal, Uttarakhand, India. He has published more than ten research papers in conferences and reputed journals. He is currently perusing his Ph.D. in data mining fields from H.N.B. Garhwal University (A Central University) Srinagar Garhwal.

Dr. Pritam Singh Negi is Assistant Professor in the Department of Computer Science and Engineering, H.N.B. Garhwal University (A Central University) Srinagar Garhwal, Uttarakhand, India. Having teaching experience of sixteen years in the field of Computer Science & Applications in University, he is actively engaged in research work and has published number of research papers in reputed journals. His areas of interests include Information Retrieval, NLP, Data Mining and Network Security.

Lokesh Singh Panwar received his B.Tech. degree from HNB Garhwal University (A Central University), Uttarakhand, India and M.Tech. degree from Graphic Era University Dehradun India. He has five year industrial experience and one Years of teaching experience as a guest faculty in the Department of Electronics and Communication Engineering, H.N.B. Garhwal University (A Central University) Srinagar Garhwal, Uttarakhand, India. He has published more than six research papers in conferences and reputed journals. His areas of interests include VLSI Design, Semiconductor Devices, Machine Learning, MEMS and Sensor Design.

Dr. Y. P. Raiwani is Professor in the Department of Computer Science and Engineering, H.N.B. Garhwal University (A Central University) Srinagar Garhwal, Uttarakhand, India. Having teaching experience of more than twenty eight years in the field of Computer Science & Applications in University, He is actively engaged in research work and has published number of research papers in reputed national and international journals. His areas of interests include Data Mining, Machine Learning, Network Security, Cloud Computing and E-M Commerce.