International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878, Volume-8 Issue-5, January 2020

A Novel Framework for NIDS Throuh Fast Knn Classifier on CICIDS 2017 Dataset

K. Vamsi Krishna, K. Swathi, B.Basaveswara Rao

Abstract: ***This paper investigates the performance of a Fast k-Nearest Neighbor Classifier (FkNN) for Network Intrusion Detection System (NIDS) on Cloud Environment. For this study Variance Index based Partial Distance Search (VIPDS) kNN [7] is adopted as an FkNN classifier. A benchmark dataset CICIDS2017[16] is considered for the evaluation process because it is a 78 featured dataset with most updated cloud related attacks. To achieve this objective a frame work is proposed for implementing FkNN and compared with kNN classifier by considering two performance measures Accuracy and computational time. This study explores the gain in the computational time without compromising the Accuracy while using FkNN instead of kNN over a large featured dataset. The conclusions are drawn as per the results obtained from the experiments conducted on CICIDS2017 dataset.***

Keywords: Fast kNN Classifier, Network Intrusion Detection System, Variance Indexing, and Cloud.

I. INTRODUCTION

Now a days, a wide number of could computing servers across globe are being attacked with Distributed Denial of Service (DDoS) attacks, for financial, grudge and other political reasons. In order to put an end to this is seriouscyber threats, this segment has been chosen by several researchers [10].It is strenuous for the attackers to attack the server physically, so the attackers to choose the malicious activities (like resource consumption, IP addresses spoofing etc.,) for unavailability of services to legitimate users and to damage the Cloud Service Providers(CSP) economically. However, the attackers widely use traditional DDoS attacks against customer chain of cloud networks, named as Economic Denial of Sustainability (EDoS)[20] because the attackers concentrated the financial losses of the CSPs. Many of the CSPs took a leap on pay-as-you-use schemes in order to provide service only to paid customers, obeying Service Level Agreement (SLA), in order to allocate customers required amount of resources. This payment method drasticallyresulted in slackening of customers count. Researchers and scholars across the globe came up with many significant measures to detect and mitigate DDoS attackers and provide service only to the legitimate customers. Statistical and Machine Learning (ML) techniques through systematic, simulation and integration methods are implemented by several researchers as key solutions to mitigate these attacks [1ande proposed frame

work sing FkNN over a large featured dataset in terms of computational time with out compermising the accu][4][8][9].An ML approach is a highly reliable and feasible platform compared to that of statistical modelling because of ML works on a set of predefined algorithms which iterate in parallel with customer execution predicting whether user is a legitimate or malware [3][5].

Several researchers from the last two decades implemented ML algorithms for NIDS, out of these studies kNN classifier is also widely used. To overcome the lazy learning nature of the kNN classifier, a fast kNN classification approach based on Partial Distance Search (PDS) is implemented[8][9] and the experiments were conducted on Kyoto and NSLKDD datasets were implanted[14][15].

After review of these studies the following observations are identified. (i) Many of these works are not implemented on cloud environmentand also not concentrated on new types of attacks. (ii) Most of the studies are adopted with traditional attack types and consists of less number of features like KDDCUP 99 and KYOTO-2006+ datasets [14][15]. (iii) These benchmark datasets are not generated on cloud environment and they are obsolete.In this connection to address this problem, with the adoption of a benchmark dataset CICIDS2017 and to perform experiments using fast kNN classifier adopted by [7][8] for NIDS. The main objectives of this paper areof two fold i) to givethe prior knowledge to the defenders about the different types of malicious activities and how to implement new type of mitigation algorithms against to these activities effectively within a short span of time.ii) To investigate the impact of the fast kNN classifier in spite of traditional kNN. The motivation of this study is provide a bird-eye-view on proper prior knowledge on cloud based NIDS using ML approaches with the help of recent bench mark dataset. The rest of the paper is organized as follows:

In the next section the related work is presented. The basics and dataset description are given in section 3. The methodology for conducting experiments also explained in section 4. Finally results and conclusions are mentioned in section 5.

Manuscript published on January 30, 2020.
* Correspondence Author
K. Vamsi Krishna*, Ph. D. scholar in the Dept. of Computer Science and Engineering, Acharya Nagarjuna University, Guntur, India
K. Swathi, Professor in the Department of CSE, NRI Institute of Technology, Vijayawada.
B.Basaveswara Rao, Ph.D. scholars in the Dept. of Computer Science and Engineering, Acharya Nagarjuna University, Guntur, India.

II. RELATED WORK

Several researchers have been working in the field of NIDS techniquesthrough ML approaches from last few decades[2]. In view of the wide usage of cloud, the malicious activities are also increased in the cloud environment so the researchers also developed defense mechanisms from the last decade. This section discusses some of them briefly.

Table I presents a brief description of various researchers along with the datasets considered, methodologies adopted/proposed.

Table I: Related work

Ref. No.	Description
[1]	This paper proposed Spark-Chi-SVM model for Big Data environment. A ChiSqSelector algorithm is used for feature selection approach and applied on KDD Cup 99 dataset and achieved 99% of Accuracy.
[4]	This paper investigated a Back Propagation Neural Network (BPNN) approach based on genetic algorithm for intrusion detection system in cloud environment. KDD cup'99 data set is used in this paper for experiment evaluation and achieved 98.82% of Accuracy.
[5]	The authors of this article introduced a time-based sliding window algorithm as a data preprocessing technique and an ensemble approach on Random Forest algorithm for classification. The experiment is tested on CIDDS-01 dataset that has given an Accuracy of 97%
[8] [9]	This paper adopts a Fast k Nearest Neighbor Classification algorithm for NIDS. The authors applied a partial distance search method as a distance metric, Variance based feature indexing methods. The proposed models are applied on NSL-KDD as well as Kyoto 2006+ dataset and achieved upto 99% of Accuracy.
[11]	This paper presented a detailed study on Cloud challenges and security risks.
[12]	This paper proposed a meta-heuristic optimized approach based on Particle Swap Optimization (PSO). This proposed work is implemented on NSL-KDD. They achieved 98.64% of Accuracy
[13]	The authors of this paper presented flow-based intrusion detection system for high-speed network environment.
[14]	The author studied on the effect of deep Learning approach on Cloud Intrusion detection system. The experiment was implemented on KDD CUP 99 dataset.

From the related work, it is observed that most of the authors proposed various machine learning techniques like kNN, SVM, Random Forest, Deep learning etc. Very few authors concentrated on the reduction of the computational time which is one of the most important parameters of NIDS. All these works were tested on benchmark datasets like KDDCUP 99, NSL-KDD, CIDDS-01 among them most of the datasets are older and might not addressing the latest attacks. From the above observations the work is carriedout to minimize the detection time of the classifier without compromising the detection rate.

III. BASICS AND DATASET DESCRIPTION

This section discusses the basic notations and concepts that are using in the rest of this paper.

A. Basics

The work in this paper is based on the kNN classifier and its variations. ThekNN classifier is a distance based algorithm and Euclidian distance is a common distance metric used by most of the researchers. The Euclidian distance formula used in this paper is given below.

$$d(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \quad (1)$$

Where d is the distance metric, x and y are data samples for which distance is to be measured, n is the number of features in data samples.

- **k- Nearest Neighbor Classifier:**

k-NN classifier is a distance based learner. In this approach no prior model is built before testing due to this, the model is named as lazy learning. Even though the kNN classifier gives best accuracies due to its high computational time for testing limits the most of the researchers to use this approach.

- **Partial Distance Search based kNN:**

A PDS based kNN classifier is a variation of kNN classifier that minimizes the computation time of the classifier by applying a partial distance search (PDS) as a distance metric instead of Euclidian distance.The PDS algorithm will discards an instance (Sr) from the sample set (S)that is having high distance value when compared with the current k nearest distances at the earliest without completely computing the distance.

The following is the formula for PDS adopted in this methodology.

$$D(x, S_r) = \sum_{i=1}^{l}(x_i - y_i)^2 \quad (2)$$

Here $D(x, S_r)$ is a squared distance function, x is the data sample for which need to find the k nearest neighbors from the available sample set S. $0< l\leq n$ where n is the total number features. If l =n then then the sample S_r is considered as new nearest to x otherwise if the $D(x,S_r)$ is greater than the maximum k distances computed upto r-1 then the sample S_r is discarded as it not in k nearest neighbors of x in this case the value of l is less than *n*.

- **Fast kNN classifier (FkNN):**

The basic idea behind FkNN is to reorder the features in the vector in such a way that the featurewhich contributes major part in the distance measure to be placed first. i.e., before performing the VIPDS, allthe features in the vector are indexed based on their contribution in the distance measure. The variances offeatures of these vectors determine their contributions to the distance to speed up searching the k closest vectors.

B. DATSET:

CICIDS2017 dataset contains benign and the most up-to-date common attacks, which resembles the true real-world data (PCAPs). It also includes the results of the network traffic analysis using CICFlowMeter with labeled flows based on the time stamp, source, and destination IPs, source and destination ports, protocols and attack (CSV files). Also available is the extracted features definition.

Generating realistic background traffic was top priority in building this dataset. The data capturing period started at 9 a.m., Monday, July 3, 2017 and ended at 5 p.m. on Friday July 7, 2017, for a total of 5 days. Monday is the normal day and only includes the benign traffic.Different attack types - FTP, Brute Force SSH, DoS, Heartbleed, Web Attack, Infiltration, Botnet and DDoS. They have been executed both morning and afternoon on Tuesday, Wednesday, Thursday and Friday[16].

In the current study only Friday morning period of data is considered. This data set consists of 1,91,033 instances and 78 features including decision attribute having two class labels namely benign and Bot. A detailed description of the features in the dataset is skipped in this paper because it is available in [6].

C. PERFORMANCE METRICS:

Accuracy, Precision and Recall are considered as the performance measures for identifying the impact of classifiers efficiency. A confusion matrix is calculated with the entries of True Positive (TP), True Negative (TN), False Positive (FP)and False Negative (FN) values.

Where TP is the total number of correctly predicted normal samples, TN is the total number of correctly predicted attack samples, FP and FN the number of normal samples predicted as attacks and number of attack samples predicted as normal respectively.

Accuracy: Accuracy is considered as the ratio of total number of testing samples correctly classified out of the total number of samples.

$$\text{ACCURACY} = \frac{TP+TN}{TP+TN+FP+FN} \quad (3)$$

Accuracy can only be considered when all the samples equally distributed as per the class labels. But when the data distribution is skewed, i.e., not distributed equally for positive and negative samples then precision and recall were also considered.

Precision: Precision is the ratio of actual positive samples out of total samples that are predicted as positive samples.

$$\text{Precision} = \frac{TP}{TP+FP} \quad (4)$$

Recall : Recall is also known as sensitivity is the ratio of total number of correctly classified positive samples over the total number of actual positive samples. This will helpful when false negative values are high.

$$\text{Recall} = \frac{TP}{TP+FN} \quad (5)$$

IV. METHODOLOGY

The methodology of the proposed frame work comprised into three phases, they are i) data preprocessing, ii) variance based feature indexing iii) classification. Figure 1 depicts the proposed methodology. In the first phase only normalization is carried out because all the features in the dataset are numeric and there is no need of transformation.

Normalization: Feature normalization is an important and necessary step in the domain of NIDS. By nature CICIDS17 data set features describe various characteristics of the data and the values are quantitative with different ranges.

These feature values influenced the data analysis or classification process. For example features with higher values can dominate the features with less value. Now the features are need to be normalized to eliminate such dominance by scaling them all within a specific range. In this paper for normalization process the min-max normalization technique is used. The formula for the normalization is as follows:

$$x' = \left(\frac{x-\min_X}{\text{mzax}_X-\min_X} \times (\text{new\_max}_X - \text{new\_min}_X) + \text{new\_minX} \right) \quad (6)$$

Here X is the feature vector to which normalization is to be applied. x is a value in X and x' is the normalized value of x. $\min_X$and$\max_X$ are minimum and maximum values of the feature X. new_min_X and $\text{new_max}_X$are new scale into which the X is be to normalized with a scale of (0,1).The CICIDS 2017 dataset is converted into normalized CICIDS 2017N dataset. An example of normalization for three records are given in Appendix -I.

In the second phase the preprocessed dataset isreordered based on feature ranks, the ranks are computed according to thedescending order of each feature variance value.

The variance formula is given below.

$$v(X) = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2 \quad (7)$$

Where v is the variance function, X is a feature vector for whichvariance needs to be calculated and xi is the i^{th} instance valueand $\bar{x}$ is the mean of the feature vector X. Appendix –II provides listof feature number, feature names and their variance index ranks in descending order of their variance value. The CICIDS2017N dataset is reordered into CICIDS2017V.

In the classification phase three classification techniques discussed in section 3 were implemented. kNN, PDS-kNN were applied on the CICIDS2017N dataset and FkNN is applied on the CICIDS2017V dataset with a 10 fold cross validation.

In this model the learning ability of three classifiers identified with the implementation of different values of k (ie., 3, 5 and 7) because the number of features as well instances in the dataset are high and it will consume more computational time if the value of k is large.The following figure exhibits the flow of the methodology.

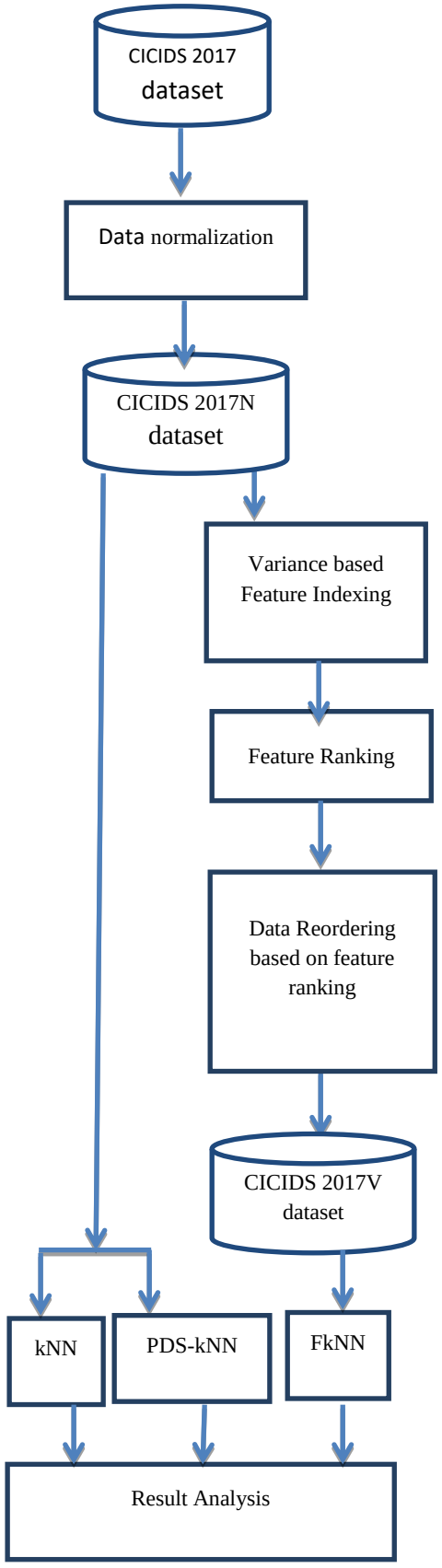

Figure 1: Proposed Methodology

V. RESULTS AND DISCUSSIONS

The methodology is implemented in Intel core i3 processor with Windows 10 operating system, with the JDK1.8 as java compiler.After implementing three classifiers kNN, PDS-kNN and FkNNfor different values of k (3,5 and 7) with 10 fold cross validation the following results obtainedregarding to the Accuracy, Precision, Recall and computational time. The Accuracy, Precision and Recall are calculated form the confusion matrices. The computational time is measured based on execution time of the classifier. In the following subsections the significance of the performance metrics are explained.

A. Accuracy, Precision and Recall:

The following confusion matrix is obtained after execution of three classifiers.

Table II: Confusion matrix for kNN, PDS-kNN and Fast kNN classifiers with different k values (3, 5 and 7)

Classifier Name	*K* value	TP	TN	FP	FN
FkNN	7	188749	1762	206	194
PDS-kNN	7	188749	1762	206	194
kNN	7	188749	1762	206	194
FkNN	5	188759	1789	196	167
PDS-kNN	5	188759	1789	196	167
kNN	5	188759	1789	196	167
FkNN	3	188786	1826	169	130
PDS-kNN	3	188786	1826	169	130
kNN	3	188786	1826	169	130

By observing the confusion matrix

i) All the entries are equal irrespective of type of classification. So there is no impact on detection rate by using PDS-kNN, FkNN instead of kNN.

ii) There is no significant difference between the various values of k.

The Accuracy, Precision and Recall are calculated as per the formulas defined in section III.C and from the above confusion matrix.

Table III presents Accuracy, Precision and Recall measures of the classifiers at different values of k.

Table III: Accuracy, Precision and Recall measures for kNN classifier at different k values (3, 5 and 7) of different classifiers.

Classifier &*K* Value	Accuracy	Precision	Recall
3	99.8434	99.9106	99.9312
5	99.8099	99.8963	99.9116
7	99.7905	99.8909	99.8973

The Accuracy, Precision and Recall values of all three classifiers are equal because the confusion matrix entries are equal. The variation in the k value is not much influencing on these three metrics.

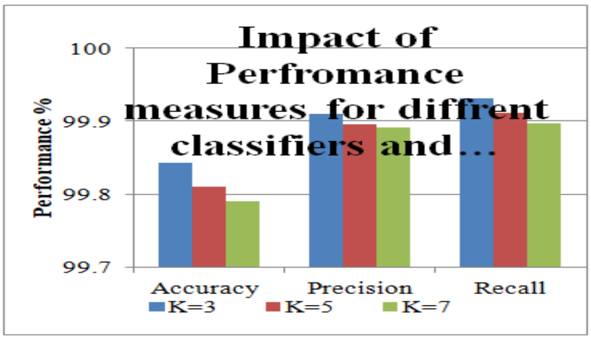

Figure 2: Impact of Accuracy, Precision and Recall for different kNN classifiers at different k values (3, 5 and 7)

B. Computational time:

The execution time of any detection mechanism is very important along with high detection rates. The execution times of these three classifiers are presented in the Table IV and the gain in the computational time is presented in Table V.

Table IV: Computational Time for kNN, PDS-kNN and Fast kNN classifiers with different k values(3, 5 and 7)

K Value	Computational Time in seconds		
	FkNN	PDS-kNN	kNN
3	1,784	2,176	15,534
5	1,835	2,411	15,516
7	1,877	2,593	15,660

Table V: The percentage of gain in computational time for three classifiers at different values of k (i.e., 3, 5 and 7).

K Value	% of Gain in Computational Time		
	kNN – PDS-kNN	kNN-FkNN	PDS-kNNFkNN
3	85.99	88.52	18.01
5	84.46	88.17	23.89
7	83.44	88.01	27.67

It is observed that the gain in the computational time of the FkNN classifier is high when compared with kNN classifier whereas the gain in computational time is low when compared to PDS-kNN classifiers.

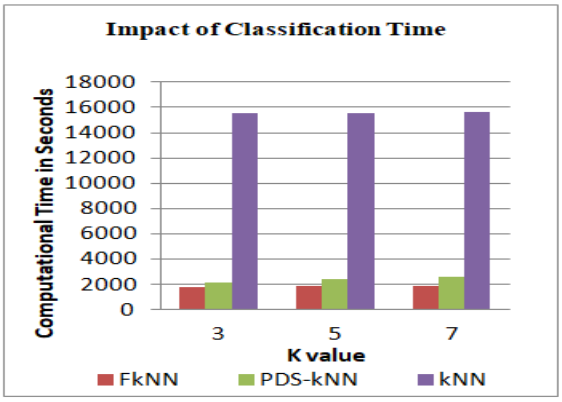

Figure 3: Impact of Computational Time for kNN PDS-kNN and FkNN classifiers with different k values(3, 5 and 7)

For k=3 the gain in computational time among kNN and FkNN is 88.52% which is observed as a highest value.

Among the kNN and FkNN the gain in computational is stabilized at 88% for different values of k.

VI. CONCLUSION

In this paper, aFkNN classifier is introduced as an intrusion detection system for cloud environment onCICIDS2017 benchmark dataset.

The FkNN is compared with kNN and PDS-kNNintermsof Accuracy, Precision, Recall and computational time. From the experimental results it is concluded that the FkNN classifier is a better classifier with less detection time and without loss in Accuracy. The gain in computational time is morethan 88% as compared to traditional kNN. The authors suggested FkNNas a better machine learning algorithm for NIDS with latest attack types and with high dimensional dataset for cloud environment to mitigate attacks and reduce the economic losses of CSPs with less span of time. As a future scope of this study is to implement on a real-time environment and can be compared with other classifiers also.

Appendix – I

List of three original and three normalized samples from CICIDS 2017 dataset

S. No	Original data samples from the CICIDS-2007 dataset before normalization
1	3268,112740690,32,16,6448,1152,403,0,201.5,204.7242047,72,72,72,0,67.41133126,0.425755776,2398738.085,5798697.94,16400000,3,113000000, 3636796.452,6848760.823,16400000,3,113000000,7516023.2,8323384.915,16400000,3,1,0,0,0,1024,512,0.283837184,0.141918592,0,403,163.3265306,178.9317127,32016.55782,0,1,0,0,1,0,0,0,0,166.7291667,201.5,72,1024,0,0,0,0,0,0,32,6448,16,1152,377,2079,15,32,359.4285714,11.99801571,380,343,16100000,498804.8203,16400000,15400000,BENIGN
2	389,112740560,32,16,6448,5056,403,0,201.5,204.7242047,316,316,316,0,102.0395854,0.425756267,2398735.319,5798709.67,16400000,2,113000000,3636792.258,6848776.836,16400000,2,113000000,7516016.133,8323376.147,16400000,4,1,0,0,0,1024,512,0.283837512,0.141918756,0,403,243,174.7169139,30526,0,1,0,0,1,0,0,0,0,248.0625,201.5,316,1024,0,0,0,0,0,0,32,6448,16,5056,955,2079,15,32,320.2857143,15.74499165,330,285,16100000,498793.6656,16400000,15400000,BENIGN
3	0,113757377,545,0,0,0,0,0,0,0,0,0,0,0,0,4.790898088,209112.8254,1395543.434,20800000,0,114000000,209112.8254,1395543.434,20800000,0,0,0,0,0,0,0,0,0,0,0,0,4.790898088,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,545,0,0,0,-1,-1,0,0,9361828.6,7324645.883,18900000,19,12200000,6935824.002,20800000,5504997,BENIGN
S. No	**data samples from the CICIDS-2007 dataset after normalization**

1	0.050317177,0.939505811,0.000149065,5.62E-05,0.00522041,1.84E-06,0.016236906,0,0.033917665,0.029041082,0.005524862,0.043929225, 0.019010706,0,0.005763721,0.400000085,0.020157562,0.068380872,0.136666753,1.34E-07,0.941666667,0.030306637,0.082021088, 0.136666667, 2.50E-08, 0.941666667,0.062633527,0.099800778,0.136666667,2.50E-08,1,0,0,0,0.000234353,9.00E-05,9.46E-08,7.10E-08,0,0.016236906,0.072090186,0.037997894,0.001442187,0,1,0,0,1,0,0,0,0,0.071619058,0.033917665,0.019010706,0.000234353,0,0,0,0,0,0,0.000149065,0.00522041,5.62E-05,1.84E-06,0.005767822,0.031738281,7.55E-05,0.571428571,3.39E-06,2.38E-07,3.58E-06,3.24E-06,0.134166667, 0.006511812,0.136666667,0.128333333,BENIGN
2	0.005989407,0.939504728,0.000149065,5.62E-05,0.00522041,8.06E-06,0.016236906,0,0.033917665,0.029041082,0.024248005,0.192800488, 0.083435876,0,0.005763738,0.400000085,0.020157538,0.06838101,0.136666753,1.26E-07,0.941666667,0.030306602,0.082021279, 0.136666667,1.67E-08,0.941666667,0.062633468,0.099800673,0.136666667,3.33E-08,1,0,0,0,0.000234353,9.00E-05,9.46E-08,7.10E-08,0, 0.016236906,0.107257009,0.03710284,0.001375045,0,1,0,0,1,0,0,0,0,0.106556057,0.033917665,0.083435876,0.000234353,0,0,0,0,0,0,0.000149065,0.00522041,5.62E-05,8.06E-06,0.014587402,0.031738281,7.55E-05,0.571428571,3.02E-06,3.12E-07,3.11E-06,2.69E-06,0.134166667, 0.006511667, 0.136666667,0.128333333,BENIGN
3	0,0.947978202,0.00261585,0,0,0,0,0,0,0,0,0,0,0,0,0.005763689,0.400000958,0.001757351,0.01645688,0.173333416,1.09E-07,0.95,0.001742607, 0.016713095,0.173333333,0,0,0,0,0,0,0,0,0,0,0,0,0,1.60E-06,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0.00261585,0,0,0,0,0,0,0,0, 0.088319138, 0.145330275,0.178301887,1.79E-07,0.101666667,0.090546005,0.173333333,0.045874975,BENIGN

Appendix – II

List of features in variance index ranking in descending order from CICIDS 2017 dataset along with #feature, feature name and corresponding ranks

S.No	Feature#	Feature Name	Variance value
1	47	ACK Flag Count	0.193003203
2	46	PSH Flag Count	0.175372367
3	48	URG Flag Count	0.089500999
4	1	Destination Port	0.066020158
5	2	Flow Duration	0.065490096
6	21	Fwd IAT Total	0.064933109
7	26	Bwd IAT Total	0.060896978
8	31	Fwd PSH Flags	0.048302998
9	44	SYN Flag Count	0.048302998
10	66	Init_Win_bytes_forward	0.044419275
11	67	nit_Win_bytes_backward	0.019872749
12	19	Flow IAT Max	0.015077018
13	24	Fwd IAT Max	0.014958496
14	76	Idle Max	0.014092489
15	74	Idle Mean	0.013391791
16	77	Idle Min	0.013050813
17	69	min_seg_size_forward	0.01298375
18	29	Bwd IAT Max	0.012834673
19	43	FIN Flag Count	0.012709407
20	14	Bwd Packet Length Std	0.009655671
21	27	Bwd IAT Mean	0.007498124
22	22	Fwd IAT Mean	0.007398288
23	30	Bwd IAT Min	0.007164029
24	25	Fwd IAT Min	0.007063009
25	18	Flow IAT Std	0.0056745
26	54	AvgBwd Segment Size	0.005293753
27	13	Bwd Packet Length Mean	0.005293753

S.No	Feature#	Feature Name	Variance value
28	52	Average Packet Size	0.00518673
29	40	Packet Length Mean	0.005077867
30	36	Fwd Packets/s	0.004891028
31	51	Down/Up Ratio	0.00410754
32	11	Bwd Packet Length Max	0.003721604
33	41	Packet Length Std	0.002611162
34	23	Fwd IAT Std	0.002485448
35	12	Bwd Packet Length Min	0.002441801
36	28	Bwd IAT Std	0.001904407
37	16	Flow Packets/s	0.001867307
38	39	Max Packet Length	0.001359742
39	17	Flow IAT Mean	0.001260407
40	75	Idle Std	0.000801149
41	10	Fwd Packet Length Std	0.000518359
42	7	Fwd Packet Length Max	0.000499358
43	38	Min Packet Length	0.000442076
44	20	Flow IAT Min	0.00039501
45	9	Fwd Packet Length Mean	0.00038827
46	53	AvgFwd Segment Size	0.00038827
47	8	Fwd Packet Length Min	0.00032509
48	37	Bwd Packets/s	0.000282706
49	45	RST Flag Count	0.000251364
50	50	ECE Flag Count	0.000251364
51	74	Active Max	0.000158412

52	42	Packet Length Variance	0.000157207
53	15	Flow Bytes/s	0.000140068
54	71	Active Std	9.57E-05
S.No	**Feature#**	**Feature Name**	**Variance value**
55	70	Active Mean	6.71E-05
56	73	Active Min	5.18E-05
57	5	Total Length of Fwd Packets	4.12E-05
58	63	SubflowFwd Bytes	4.12E-05
59	34	Fwd Header Length	2.81E-05
60	65	SubflowBwd Bytes	2.80E-05
61	6	Total Length of Bwd Packets	2.80E-05
62	62	SubflowFwd Packets	2.79E-05
63	3	Total Fwd Packets	2.79E-05
64	68	act_data_pkt_fwd	2.77E-05
65	35	Bwd Header Length	2.73E-05
66	4	Total Backward Packets	2.71E-05
67	64	SubflowBwd Packets	2.71E-05
68	56	FwdAvg Bytes/Bulk	0
69	61	BwdAvg Bulk Rate	0
70	33	Fwd URG Flags	0
71	57	FwdAvg Packets/Bulk	0
72	58	FwdAvg Bulk Rate	0
73	49	CWE Flag Count	0
74	34	Bwd URG Flags	0
75	32	Bwd PSH Flags	0
76	60	BwdAvg Packets/Bulk	0
77	59	BwdAvg Bytes/Bulk	0

REFERENCES

1. Othman, Suad Mohammed, et al. "Intrusion detection model using machine learning algorithm on Big Data environment." Journal of Big Data 5.1 (2018): 34.
2. Salo, Fadi, Ali Bou Nassif, and Aleksander Essex. "Dimensionality reduction with IG-PCA and ensemble classifier for network intrusion detection." Computer Networks 148 (2019): 164-175.
3. Wagh, SharmilaKishor, Vinod K. Pachghare, and Satish R. Kolhe. "Survey on intrusion detection system using machine learning techniques." International Journal of Computer Applications 78.16 (2013).
4. Chiba, Z., Abghour, N., Moussaid, K., El Omri, A., &Rida, M. (2019, June). An Efficient Network IDS for Cloud Environments Based on a Combination of Deep Learning and an Optimized Self-adaptive Heuristic Search Algorithm. In International Conference on Networked Systems (pp. 235-249). Springer, Cham.
5. Idhammad, Mohamed, Karim Afdel, and Mustapha Belouch. "Distributed intrusion detection system for cloud environments based on data mining techniques." Procedia Computer Science 127 (2018): 35-41.
6. Abdulhammed, Razan, et al. "Features Dimensionality Reduction Approaches for Machine Learning Based Network Intrusion Detection." Electronics 8.3 (2019): 322.
7. Basaveswara Rao B &Swathi K (2016) Variance-Index Based Feature Selection Algorithm for Network Intrusion Detection, IOSR Journal of Computer Engineering (IOSR-JCE) e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 18, Issue 4, Ver. V (Jul.-Aug. 2016), PP 01-11.
8. Basaveswara Rao B &Swathi, K. (2017). Fast kNN Classifiers for Network Intrusion Detection System. Indian Journal of Science and Technology, 10(14).
9. Swathi, Kailasam, and BobbaBasaveswara Rao. "Impact of PDS Based kNN Classifiers on Kyoto Dataset." International Journal of Rough Sets and Data Analysis (IJRSDA) 6.2 (2019): 61-72.
10. Ali, Mohammed Hasan, and Mohamad FadliZolkipli. "Intrusion-Detection System Based on Fast Learning Network in Cloud Computing." Advanced Science Letters 24.10 (2018): 7360-7363.
11. Ahmed, H. A. S., Ali, M. H., Kadhum, L. M., Zolkipli, M. F., &Alsariera, Y. A. (2017). A review of challenges and security risks of cloud computing. Journal of Telecommunication, Electronic and Computer Engineering (JTEC), 9(1-2), 87-91.
12. Ali, Mohammed Hasan, et al. "A hybrid Particle swarm optimization-Extreme Learning Machine approach for Intrusion Detection System." 2018 IEEE Student Conference on Research and Development (SCOReD). IEEE, 2018.
13. Umer, Muhammad Fahad, Muhammad Sher, and Yaxin Bi. "Flow-based intrusion detection: Techniques and challenges." Computers & Security 70 (2017): 238-254.
14. "Nsl-kdd data set for network-based intrusion detection systems." Available on: http://nsl.cs.unb.ca/KDD/NSLKDD.html, March 2009.
15. Kyoto 2006+ dataset is available on : http://www.takakura.com/Kyoto_data/
16. https://www.unb.ca/cic/datasets/ids-2017.html
17. Basaveswara Rao B, et al., "A Fast KNN Based Intrusion Detection System for Cloud Environment", Jour of Adv Research in Dynamical & Control Systems, Vol. 10, Issue 7, 2018 PP 1509 -1515.
18. Ring, Markus, Sarah Wunderlich, DenizScheuring, Dieter Landes, and Andreas Hotho. "A Survey of Network-based Intrusion Detection Data Sets." Computers & Security (2019).
19. Ahmim, A., Maglaras, L., Ferrag, M. A., Derdour, M., &Janicke, H. (2019, May). A novel hierarchical intrusion detection system based on decision tree and rules-based models. In 2019 15th International Conference on Distributed Computing in Sensor Systems (DCOSS) (pp. 228-233). IEEE.
20. Bulla, Suneetha, et al. "An experimental evaluation of the impact of the EDoS attacks against cloud computing services using AWS." International Journal of Engineering & Technology 7.1.5 (2018): 202-208.

AUTHORS PROFILE

K. Vamsi Krishna is a Ph. D. scholar in the Dept. of Computer Science and Engineering, Acharya Nagarjuna University, Guntur, India. He has received his M.Tech (2009) degree in CSE from JNTU Kakinada and published about 6 research papers in international journals in the fields of Network & Cloud Security, Bigdata analytics, Datamining, etc.

K. Swathireceived her Ph.D. in 2019 in Computer Science and Engineering from AcharyaNagarjuna University, Guntur, India. Currently Professor in the Department of CSE, NRI Institute of Technology, Vijayawada. She has published about ten international journals and about ten national/international conferences in the fields of Network Security, Data Analytics, Machine Learning, Cloud Security, Technology in Education, etc.

Bobba B Rao received his Ph.D. in 2004, and about 7 Ph.D. scholars were awarded under his guidance and guiding about 5 Ph.D. scholars in the Dept. of Computer Science and Engineering, Acharya Nagarjuna University, Guntur, India. Currently working in areas like Network Security, Cloud Computing, Mobile Ad-hoc Networks, Big data Analytics, game theory, etc and published 30+ papers in various national/international journals and Conferences.