Analysis of Intrusion Detection in KDD '99 Dataset

Jenifar Dayana.A[1]
[1] Department of Computer Science
[1] Bharathidasan Univesity, Trichy

Abstract- *The incredible growth of the web-based applications has increased information security vulnerabilities over the Internet. Security administrator use Intrusion-Detection System (IDS) to monitoring the network traffic and host activities to detect the attacks against host and network resources. It based on Discretization of Entropy which is a pre-processing algorithm. It split the data at points and it is given as input to data mining algorithms. The proposed representation is tested and compared with the other methods using KDD CUP 1999 dataset. The results specify that this new method achieves accuracy rates better than previous methods.*

Keywords- Detection, Preprocessing, future selections, KDD Dataset.

I. INTRODUCTION

An intrusion detection system (IDS) monitors network traffic and monitors for suspicious activity and alerts the system or network administrator. In some cases the IDS may also respond to anomalous or malicious traffic by taking action such as blocking the user or source IP address from accessing the network.Generally, there are four categories of attacks

They are: 1.**DoS (denial-of-service)**: DoS is a class of attack where an attacker makes some computing or memory resource too busy or too full to handle legitimate requests. A denial of service attack can be argued to have a distinct set of features and patterns that manifest themselves when examine packets on the network. Eg: ping-of-death, teardrop, smurf, SYN flood, and the like.

2. R2L: A remote to user (R2L) attack is a class of attack where an attacker sends packets to a machine over a network, then exploits machine's vulnerability to illegally gain local access. Eg: guessing password.

3. U2R (User to root): U2R exploit is a class of attack where an attacker starts out with access to a normal user account on the system and is able to exploit vulnerability to gain root access to the system. Most common exploits in this class of attacks are regular buffer overflows, which are caused by regular programming mistakes and environment assumptions. Eg: various "buffer overflow" attacks.

4. Probing: Probing is a class of attack where an attacker scans a network to gather information or find known vulnerabilities. An attacker with a map of machines and services that are available on a network can use the information to look for exploits. Eg: port- scan, ping-sweep, etc.

This study is based on the combination of a min max normalization and Detection Rate.

II. REVIEW OF LITERATURE

1. Entropy based Anomaly Detection System to Prevent DoS Attacks in Cloud [A.S.Syed Navaz, V.Sangeetha, C.Prabhadev]2014

Users are allowed to pass through router in network site in that it incorporates Detection Algorithm and detects for legitimate user. Second, again it pass through router placed in cloud site in that it incorporates confirmation Algorithm and checks for threshold value, if it's beyond the threshold value it considered as legitimate user, else it's an intruder found in environment.

2. A Proposed HTTP based IDS Production and hosting by Elsevier B.V. on behalf of Faculty of Computers and Information, Cairo University[2014]

The main objective is to enhance IDS performance through preparing the training data set allowing to detect malicious connections that exploit the http service.

3. An Improved Packet size Entropy Based DoS Attack Detection Scheme Kumar T, Aswani (2013).

This paper introduces a new parameter to the packet size entropy based DoS attack detection scheme so that it can improve the detection accuracy. The new parameter is the entropy of the source and destination IP address combination. I.e. a concatenation of both addresses will give a hash like value, which can uniquely identify a particular path.

III. INTRUSION DETECTION DATASET

In this study, we will use the KDD CUP 1999 intrusion detection contest data. This data was prepared by the

1998 DARPA Intrusion Detection Evaluation program by MIT Lincoln Labs [MIT]. The program acquired 9 weeks' of raw transmission control protocol (TCP) dump data. The raw data was processed into approximately 5 million connection records. The data set contains 24 attack types. All of these attacks fall into 4 main categories as described in the Introduction of this paper. Table I summarizes the recorded attacks. Every record in the dataset has 41 features that are shown in Table II. Features 2, 3 and 4 are converted into numbers; for example, the second feature 'Protocol Type' is replaced by 1, 2 or 3 instead of the values TCP, UDP (user datagram protocol) or ICMP (internet control message protocol), respectively.

TABLE I
KDD DATASET CATEGORIES

Attack	Category	# Samples
normal	Normal	97277
smurf		280790
neptune		107201
back	Dos	2203
teardrop		979
pod		264
land		21
satan		1589
ipsweep	Probe	1247
portsweep		1040
nmap		231
warezclient		1020
guess_passwd		53
warezmaster		20
imap	R2L	12
ftp_write		8
multihop		7
phf		4
spy		2
buffer_overflow		30
rootkit	U2R	10
loadmodule		9
perl		3

TABLE II
KDD CUP'99 FEATURES

No.	Features	No.	Features
1	duration	22	is_guest_login
2	protocol_type	23	count
3	service	24	srv_count
4	flag	25	serror_rate
5	src_bytes	26	srv_serror_rate
6	dst_bytes	27	rerror_rate
7	land	28	srv_rerror_rate
8	wrong_fragt.	29	same_srv_rate
9	urgent	30	diff_srv_rate
10	hot	31	srv_diff_h_rate
11	num_fail_login	32	host_count
12	logged_in	33	host_srv_count
13	nu_comprom	34	h_same_sr_rate
14	root_shell	35	h_diff_srv_rate
15	su_attempted	36	h_src_port_rate
16	num_root	37	h_srv_d_h_rate
17	nu_file_creat	38	h_serror_rate
18	nu_shells	39	h_sr_serro_rate
19	nu_access_files	40	h_rerror_rate
20	nu_out_cmd	41	h_sr_rerro_rate
21	is_host_login		

IV. NORMALIZATION

In this process we convert the data instances to a standard form based on the training datasets distribution, ie, we make the assumption that the training data set accurately reflects the range and deviation of the feature values of the entire distribution. Then we can normalize all data instances to a fixed range of our choosing and hard code the cluster width based on this fixed range. From a given training data set the average and standard deviation feature vectors can be calculated

Calculation consists of the following two stages:

- Preprocessing
- Classification

A. Preprocessing:

Preprocessing refers to the process of extracting information about packets from network traffic for the construction of new statistical features. In preprocessing module as shown in Figure 1, different feature subsets are selected for the identification of different attacks. The input to the preprocessing module is the network traffic consisting of both labeled normal and labeled attack dataset. The output of this module is normalized dataset. The preprocessing module consists of feature selection, feature values extraction, and normalization. This stage prepares the data for training the different Machine learning algorithms during training phase

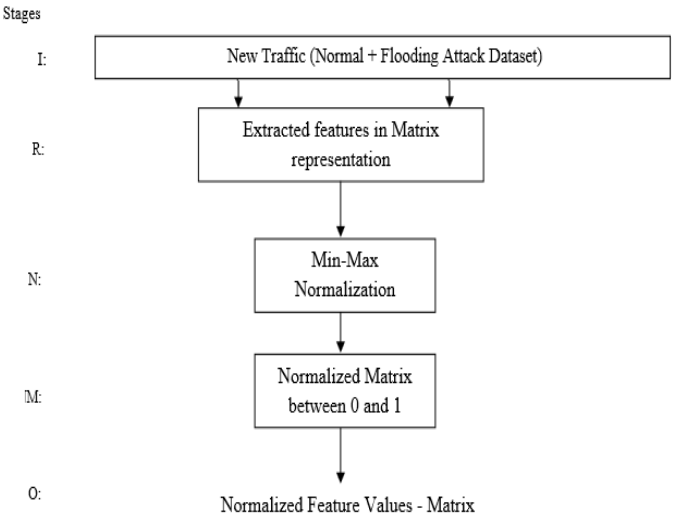

Fig 1.1 Preprocessing module for training phase

The preprocessing steps are explained as follows:

- The captured network traffic consisting of labeled normal and labeled flooding attacks is split into different attack type datasets such as SYN flood, UDP flood, and HTTP flood as shown in Figure 1, Stage I. The splitting of

datasets into three subsets has been implemented in this paper as the scope of attack type classification is restricted to SYN flood, UDP flood, and HTTP flood.

- The split datasets are given as an input to the feature extraction module to extract the feature values, as shown in Figure 1, Stage E. These features quantify the behavioral characteristics of a connection in terms of ratio and number of various data items with respect to time.

- The features present in Table I detect TCP SYN flood, TCP SYN+ACK flood, TCP Spoofed SYN flood, TCP ACK Flood, HTTP flood, HTTPS flood, and sUDP flood attack.

- Extracted feature values from each dataset are represented in matrix form consisting of first two columns as feature values and the last column as class label, Figure 1, Stage MR. From figure 1, F1 and F2 are extracted from SYN flood dataset. Similarly, F3, F4, & F5, F6 are extracted from UDP Flood and HTTP flood dataset respectively.

V. METHODOLOGY

Classification Module:

In this study, we will use the KDD CUP 1999 intrusion detection contest data. This data was prepared by the 1998 DARPA Intrusion Detection Evaluation program by MIT Lincoln Labs [MIT]. The program acquired 9 weeks' of raw transmission control protocol (TCP) dump data. The raw data was processed into approximately 5 million connection records. The data set contains 24 attack types. All of these attacks fall into 4 main categories as described in the Introduction of this paper.

Feature Selection Module

In Feature Selection Module as shown in Figure 1, different feature subsets are selected for the identification of different attacks. The input to the preprocessing module is the network traffic consisting of both labeled normal and labeled attack dataset.

Min Max Optimization Module

Normalization is a process of ensuring that each attribute value in a database is suitable for further querying, which is free from certain undesirable characteristics and eliminates the effect of scale difference.

- The extracted features as shown in Table I and its values are input to the normalization module as shown in Figure 1, Stage N.

- The feature values are scaled to the range [0, 1] using (1), where 'i (t)' denotes the value of the feature, 'min (i)' denotes the minimum value, and 'max (i)' denotes the maximum value. Thus, data available for the classifier are real numbers between 0 and 1, Stage NM.

$$inorm\ (t) = \frac{i(t) - min\ (i)}{max\ (i) - min\ (i)}$$

- These normalized three matrix files shown in Figure 1.1 and given as input to the machine learning algorithm. Thus, all the data consist of normalized values between 0 and 1, Stage O.

Algorithm to Detect Intrusion:

1. Calculate Entropy for KDD data.
2. For each potential split in KDD data.
   Calculate Entropy in each potential bin.
   Find the net entropy for the split.
   Calculate entropy gain.
3. Select the split with the highest entropy gain.
4. Recursively perform the partition on each split until a termination criteria.

   Terminate when specified number of bins.

   Terminate once entropy gain falls below a certain threshold.

VI. CONCLUSION

In this paper, the IDS based on Entropy classifier is analyzed, KDD data set is used to train and test the IDS. The proposal to enhance the IDS performance is preparing the training data set such that it could achieve 100% IDS performance. The target of the second IDS proposal is to improve the performance and to reduce the number of features by selecting only the most important features that characterize each attack type and normal connections; in addition it proposes to classify the data set based on services. As a future work, the proposed IDS can be used in the IDS running phase by installing it on a network to protect this network against real time attacks.

```
 Intrusion Detection System
**********************************************************************
List of Protocols to be Normalized
**********************************************************************
          TCP
          UDP
          CMP
          ARP
**********************************************************************
Flag in Data Set
**********************************************************************
    'OTH'
    'REJ'
    'RSTO'
    'RSTOS0'
    'RSTR'
    'RSTRH'
    'S0'
    'S1'
    'S2'
    'S3'
    'SF'
    'SH'
    'SHR'

**********************************************************************



**********************************************************************
Services in KDD Dataset
**********************************************************************
    'aol'
    'http_443'
    'http_8001'
    'http_2784'
    'domain_u'
    'ftp_data'
    'auth'
    'bgp'
    'courier'
    'tftp_u'
    'uucp_path'
    'csnet_ns'
    'ctf'
    'daytime'
    'time'
    'discard'
    'domain'
    'echo'
    'eco_i'
    'ecr_i'
    'efs'
    'exec'
    'finger'
```

```
**********************************************************
Start processing the File : sample_dataset.xlsx
**********************************************************************************
Data Set
**********************************************************************************
  Columns 1 through 17

    'tcp'    'http'     '181'    '0'    'SF'     '0'    '1'    '0'    '8'    '0.00'    '0.00'    '0.00'    '0.00'    '1.00'    '0.00'    '9'    '0.00'
    'tcp'    'http'     '239'    '0'    'SF'     '0'    '1'    '0'    '8'    '0.00'    '0.00'    '0.00'    '0.00'    '1.00'    '0.00'    '19'    '0.00'
    'tcp'    'http'     '235'    '0'    'SF'     '0'    '1'    '0'    '8'    '0.00'    '0.00'    '0.00'    '0.00'    '1.00'    '0.00'    '29'    '0.00'
    'tcp'    'daytime'    '219'    '0'    'SF'     '0'    '1'    '0'    '6'    '0.00'    '0.00'    '0.00'    '0.00'    '1.00'    '0.00'    '39'    '0.00'
    'tcp'    'http'     '217'    '0'    'SF'     '0'    '1'    '0'    '6'    '0.00'    '0.00'    '0.00'    '0.00'    '1.00'    '0.00'    '49'    '0.00'
    'tcp'    'http'     '217'    '0'    'SF'     '0'    '1'    '0'    '6'    '0.00'    '0.00'    '0.00'    '0.00'    '1.00'    '0.00'    '59'    '0.00'
    'tcp'    'imap4'    '212'    '0'    'SF'     '0'    '1'    '0'    '1'    '0.00'    '0.00'    '0.00'    '0.00'    '1.00'    '0.00'    '69'    '0.00'
    'tcp'    'smtp'     '153'    '0'    'SF'     '0'    '1'    '0'    '5'    '0.00'    '0.00'    '0.00'    '0.00'    '1.00'    '0.00'    '79'    '0.00'
    'tcp'    'http'     '210'    '0'    'SF'     '0'    '1'    '0'    '8'    '0.00'    '0.00'    '0.00'    '0.00'    '1.00'    '0.00'    '89'    '0.00'
    'tcp'    'http'     '212'    '0'    'SF'     '0'    '1'    '0'    '8'    '0.00'    '0.00'    '0.00'    '0.00'    '1.00'    '0.00'    '99'    '0.00'
    'tcp'    'ftp_data'    '210'    '0'    'S0'     '0'    '1'    '0'    '18'    '0.00'    '0.00'    '0.00'    '0.00'    '1.00'    '0.00'    '109'    '0.00'
    'tcp'    'http'     '177'    '0'    'S0'     '0'    '1'    '0'    '1'    '0.00'    '0.00'    '0.00'    '0.00'    '1.00'    '0.00'    '119'    '0.00'
    'tcp'    'http'     '222'    '0'    'S0'     '0'    '1'    '0'    '11'    '0.00'    '0.00'    '0.00'    '0.00'    '1.00'    '0.00'    '129'    '0.00'
    'tcp'    'http'     '256'    '0'    'S0'     '0'    '1'    '0'    '4'    '0.00'    '0.00'    '0.00'    '0.00'    '1.00'    '0.00'    '139'    '0.00'
    'tcp'    'http'     '241'    '0'    'SF'     '0'    '1'    '0'    '1'    '0.00'    '0.00'    '0.00'    '0.00'    '1.00'    '0.00'    '149'    '0.00'
    'tcp'    'ftp_data'    '260'    '0'    'SF'     '0'    '1'    '0'    '11'    '0.00'    '0.00'    '0.00'    '0.00'    '1.00'    '0.00'    '159'    '0.00'
    'tcp'    'ftp_data'    '241'    '0'    'SF'     '0'    '1'    '0'    '2'    '0.00'    '0.00'    '0.00'    '0.00'    '1.00'    '0.00'    '169'    '0.00'
    'tcp'    'ftp_data'    '257'    '0'    'SF'     '0'    '1'    '0'    '12'    '0.00'    '0.00'    '0.00'    '0.00'    '1.00'    '0.00'    '179'    '0.00'
    'tcp'    'ftp_data'    '233'    '0'    'SF'     '0'    '1'    '0'    '2'    '0.00'    '0.00'    '0.00'    '0.00'    '1.00'    '0.00'    '189'    '0.00'
    'tcp'    'ftp_data'    '233'    '0'    'SF'     '0'    '1'    '0'    '7'    '0.00'    '0.00'    '0.00'    '0.00'    '1.00'    '0.00'    '199'    '0.00'
    'tcp'    'http'     '256'    '0'    'S1'     '0'    '1'    '0'    '17'    '0.00'    '0.00'    '0.00'    '0.00'    '1.00'    '0.00'    '209'    '0.00'
    'tcp'    'http'     '234'    '0'    'SF'     '0'    '1'    '0'    '5'    '0.00'    '0.00'    '0.00'    '0.00'    '1.00'    '0.00'    '219'    '0.00'
    'tcp'    'http'     '241'    '0'    'SF'     '0'    '1'    '0'    '12'    '0.00'    '0.00'    '0.00'    '0.00'    '1.00'    '0.00'    '229'    '0.00'
    'tcp'    'http'     '239'    '0'    'SF'     '0'    '1'    '0'    '3'    '0.00'    '0.00'    '0.00'    '0.00'    '1.00'    '0.00'    '239'    '0.00'
    'tcp'    'http'     '245'    '0'    'SF'     '0'    '1'    '0'    '13'    '0.00'    '0.00'    '0.00'    '0.00'    '1.00'    '0.00'    '249'    '0.00'
    'tcp'    'telnet'    '248'    '0'    'SF'     '0'    '1'    '0'    '23'    '0.00'    '0.00'    '0.00'    '0.00'    '1.00'    '0.00'    '255'    '0.00'
```

```
No of Row = 100
No of Column = 18
**************************************************
```

```
Protocols in Dataset
**********************************************************************
    'SF'
    'SF'
    'SF'
    'SF'

**********************************************************************
Probabilities of the Protocol /type
**********************************************************************
   80.0000    0.8000
    8.0000    0.0800
    6.0000    0.0600
    6.0000    0.0600
```

```
**********************************************************************
robabilities of the Protocol /type
**********************************************************************
   80.0000    0.8000
    8.0000    0.0800
    6.0000    0.0600
    6.0000    0.0600

**********************************************************************
robabilities of the Flag
**********************************************************************
lag Name
    '0.89'
    '0.04'
    '0.01'
    '0H'
    '0.02'
    '0R'

lag Prbabilities
    '0.89'
    '0.89'
    '0.89'
    '0.89'
    '0.04'

**********************************************************************


-->  Currently generated is : PDF_sample.csv
started Normailaization

0.8,0.81,181,0,0.89,0,1,0,8,0.00,0.00,0.00,0.00,1.00,0.00,9,0.00,0
0.8,0.81,239,0,0.89,0,1,0,8,0.00,0.00,0.00,0.00,1.00,0.00,19,0.00,0
0.8,0.81,235,0,0.89,0,1,0,8,0.00,0.00,0.00,0.00,1.00,0.00,29,0.00,0
0.8,0.01,219,0,0.89,0,1,0,6,0.00,0.00,0.00,0.00,1.00,0.00,39,0.00,0
0.8,0.81,217,0,0.89,0,1,0,6,0.00,0.00,0.00,0.00,1.00,0.00,49,0.00,0
0.8,0.81,217,0,0.89,0,1,0,6,0.00,0.00,0.00,0.00,1.00,0.00,59,0.00,0
0.8,0.01,212,0,0.89,0,1,0,1,0.00,0.00,0.00,0.00,1.00,0.00,69,0.00,0
0.8,0.03,159,0,0.89,0,1,0,5,0.00,0.00,0.00,0.00,1.00,0.00,79,0.00,0
0.8,0.81,210,0,0.89,0,1,0,8,0.00,0.00,0.00,0.00,1.00,0.00,89,0.00,0
0.8,0.81,212,0,0.89,0,1,0,8,0.00,0.00,0.00,0.00,1.00,0.00,99,0.00,0
0.8,0.06,210,0,0.04,0,1,0,18,0.00,0.00,0.00,0.00,1.00,0.00,109,0.00,0
0.8,0.81,177,0,0.04,0,1,0,1,0.00,0.00,0.00,0.00,1.00,0.00,119,0.00,0
0.8,0.81,222,0,0.04,0,1,0,11,0.00,0.00,0.00,0.00,1.00,0.00,129,0.00,0
0.8,0.81,256,0,0.04,0,1,0,4,0.00,0.00,0.00,0.00,1.00,0.00,139,0.00,0
0.8,0.81,241,0,0.89,0,1,0,1,0.00,0.00,0.00,0.00,1.00,0.00,149,0.00,0
0.8,0.06,260,0,0.89,0,1,0,11,0.00,0.00,0.00,0.00,1.00,0.00,159,0.00,0
0.8,0.06,241,0,0.89,0,1,0,2,0.00,0.00,0.00,0.00,1.00,0.00,169,0.00,0
0.8,0.06,257,0,0.89,0,1,0,12,0.00,0.00,0.00,0.00,1.00,0.00,179,0.00,0
0.8,0.06,233,0,0.89,0,1,0,2,0.00,0.00,0.00,0.00,1.00,0.00,189,0.00,0
0.8,0.06,233,0,0.89,0,1,0,7,0.00,0.00,0.00,0.00,1.00,0.00,199,0.00,0
0.8,0.81,256,0,0.01,0,1,0,17,0.00,0.00,0.00,0.00,1.00,0.00,209,0.00,0
```

```
Normalization takes place for each feature individually
==> Finished is: NORM_sample.csv

0.8,0.81,0.16,0,0.89,0,1,0,0.25926,0,0,0,0,1,0,0,0,0
0.8,0.81,0.41778,0,0.89,0,1,0,0.25926,0,0,0,0,1,0,0.04065,0,0
0.8,0.81,0.4,0,0.89,0,1,0,0.25926,0,0,0,0,1,0,0.081301,0,0
0.8,0.01,0.32889,0,0.89,0,1,0,0.18519,0,0,0,0,1,0,0.12195,0,0
0.8,0.81,0.32,0,0.89,0,1,0,0.18519,0,0,0,0,1,0,0.1626,0,0
0.8,0.81,0.32,0,0.89,0,1,0,0.18519,0,0,0,0,1,0,0.20325,0,0
0.8,0.01,0.29778,0,0.89,0,1,0,0,0,0,0,0,1,0,0.2439,0,0
0.8,0.03,0.062222,0,0.89,0,1,0,0.14815,0,0,0,0,1,0,0.28455,0,0
0.8,0.81,0.28889,0,0.89,0,1,0,0.25926,0,0,0,0,1,0,0.3252,0,0
0.8,0.81,0.29778,0,0.89,0,1,0,0.25926,0,0,0,0,1,0,0.36585,0,0
0.8,0.06,0.28889,0,0.04,0,1,0,0.62963,0,0,0,0,1,0,0.4065,0,0
0.8,0.81,0.14222,0,0.04,0,1,0,0,0,0,0,0,1,0,0.44715,0,0
0.8,0.81,0.34222,0,0.04,0,1,0,0.37037,0,0,0,0,1,0,0.4878,0,0
0.8,0.81,0.49333,0,0.04,0,1,0,0.11111,0,0,0,0,1,0,0.52846,0,0
0.8,0.81,0.42667,0,0.89,0,1,0,0,0,0,0,0,1,0,0.56911,0,0
0.8,0.06,0.51111,0,0.89,0,1,0,0.37037,0,0,0,0,1,0,0.60976,0,0
0.8,0.06,0.42667,0,0.89,0,1,0,0.037037,0,0,0,0,1,0,0.65041,0,0
0.8,0.06,0.49778,0,0.89,0,1,0,0.40741,0,0,0,0,1,0,0.69106,0,0
0.8,0.06,0.39111,0,0.89,0,1,0,0.037037,0,0,0,0,1,0,0.73171,0,0
0.8,0.06,0.39111,0,0.89,0,1,0,0.22222,0,0,0,0,1,0,0.77236,0,0
0.8,0.81,0.49333,0,0.01,0,1,0,0.59259,0,0,0,0,1,0,0.81301,0,0
0.8,0.81,0.39556,0,0.89,0,1,0,0.14815,0,0,0,0,1,0,0.85366,0,0


0.8,0.81,0.084444,0,0.89,0,1,0,0.59259,0,0,0,0,1,0,1,0,0
0.8,0.81,0.43111,0,0.89,0,1,0,0.96296,0,0,0,0,1,0,1,0,0
0.06,0.81,0.41333,0,0.89,0,1,0,0.11111,0,0,0,0,1,0,1,0,0
0.8,0.81,0.41333,0,0.89,0,1,0,0.48148,0,0,0,0,1,0,1,0,0
0.08,0.81,0.28,0,0.89,0,1,0,0.14815,0,0,0,0,1,0,1,0,0
0.8,0.81,0.30222,0,0.89,0,1,0,0.51852,0,0,0,0,1,0,1,0,0
0.8,0.81,0.67556,0,0.89,0,1,0,0.18519,0,0,0,0,1,0,1,0,0
0.8,0.81,0.69778,0,0.89,0,1,0,0.55556,0,0,0,0,1,0,1,0,0
0.06,0.81,0.73333,0,0.89,0,1,0,0.22222,0,0,0,0,1,0,1,0,0
0.8,0.81,0.75111,0,0.89,0,1,0,0.074074,0,0,0,0,1,0,1,0,0
0.06,0.81,0.72889,0,0,0,1,0,0.44444,0,0,0,0,1,0,1,0,0
0.08,0.81,0.72,0,0.89,0,1,0,0.2963,0,0,0,0,1,0,1,0,0
0.8,0.81,0.73333,0,0.89,0,1,0,0.14815,0,0,0,0,1,0,1,0,0
0.06,0.81,0.18667,0,0.89,0,1,0,0,0,0,0,0,1,0,1,0,0
0.8,0.81,0.24889,0,0.89,0,1,0,0.37037,0,0,0,0,1,0,1,0,0
0.8,0.81,0.044444,0,0.89,0,1,0,0.037037,0,0,0,0,1,0,1,0,0
0.8,0.81,0.34222,0,0.89,0,1,0,0.40741,0,0,0,0,1,0,1,0,0
0.06,0.81,0.32889,0,0.89,0,1,0,0.25926,0,0,0,0,1,0,1,0,0
0.8,0.81,0.33333,0,0.89,0,1,0,0.11111,0,0,0,0,1,0,1,0,0
0.8,0.81,0.37778,0,0.89,0,1,0,0.48148,0,0,0,0,1,0,1,0,0
0.06,0.81,0.33778,0,0.89,0,1,0,0.33333,0,0,0,0,1,0,1,0,0
0.8,0.81,0.81778,0,0.89,0,1,0,0.037037,0,0,0,0,1,0,1,0,0
0.06,0.81,0.85333,0,0.89,0,1,0,0.40741,0,0,0,0,1,0,1,0,0
0.8,0.81,0.65778,0,0.89,0,1,0,0.77778,0,0,0,0,1,0,1,0,0
0.8,0.81,0.11556,0,0.89,0,1,0,0,0,0,0,0,1,0,1,0,0
0.06,0.81,0.14667,0,0.89,0,1,0,0.037037,0,0,0,0,1,0,1,0,0
0.08,0.81,0.82667,0,0.89,0,1,0,0.037037,0,0,0,0,1,0,1,0,0

**********************************************************************
Total execution time is: 1.045207
**********************************************************************
```

REFERENCES

[1] A.S.Syed Navaz, V.Sangeetha, C.Prabhadev Entropy based Anomaly Detection System to Prevent DDoS Attacks in Cloud. In Proceedings of the International Journal of Computer Applications (0975 – 8887) Volume 62– No.15, January 2013

[2] Mohamed M. Abd-Eldayem IT Department, Faculty of Computers and Information, Cairo University, Egypt Egyptian Informatics Journal 03/2014; 15(1). DOI: 10.1016/j.eij.2014.01.001

[3] A.P. Engelbrecht. Computational intelligence: An introduction. Wiley, 2007.

[4] P Divya et al Clustering Based Feature Selection and Outlier Analysis P Divya et al International Journal of Computer Science & Communication Networks

[5] Kumar T,Aswani An Improved Packet size Entropy Based DoS Attack Detection Scheme,NIT.

[6] The NSL–KDD dataset.<http://nsl.cs.unb.ca/NSL-KDD/>

[7] KDD Cup 1999 Data.<http://kdd.ics.uci.edu/databases /kdd-cup99/kddcup99.html>