# Comparing the Result of KDD Cup 1999 Data by using K-mean Algorithm and make Density based Cluster in Intrusion Detection System by Removing the Count Attribute

Pratik Jain
Department of Computer Science and Engineering
IPS Academy, Indore, India

Divyansh Kumrawat
Department of Computer Science and Engineering
IPS Academy, Indore, India

## ABSTRACT

An IDS monitors network traffic searching for suspicious activity and known threats, sending up to alerts when it finds such items. In the recent avocation, Intrusion detection as a magnificence still remains censorial in cyber safety. But maybe not as a lasting resolution. To understand intrusion detection firstly understand what is intrusion. Cambridge dictionary defines an intrusion as "an occasion when someone goes into a place or situation where they are not wanted or expected to be". For the purpose of this article, here it defines intrusion as any un-possessed system or network festivity on one (or more) computer(s) or network(s). This is an illustration of a lawful user of a system trying to intensify his privileges to gain greater entrance to the system that he is currently entrusted, or the same user trying to connect to an unauthorized remote port of a server. These are the intrusions that can engender from the outside world, a aggrieved ex-employee who was fired lately, or from your faithful staff. In this clause, the mediocre data is discovered as invasion when the case is false positive. Here they are focusing on this problem with an illustration & offering one solution for the same problem. The KDD CUP 1999 data set is used. In the outcome of this experiment it can be seen that if a class has higher number of counts then this class is opined as an anomaly class. But it will be count as anomaly if the true person is passing the threshold value. One solution is proposed to detect the true person and to remove false positive.

## Keywords

Data mining, Anomaly Detection System (ADS), K-Means, Ensemble, Detection rate, False alarm rate, false positive, Clustering

## 1. INTRODUCTION

In the last two decades, with the growth of computer technology, safety of network system is become a crucial issue, as computer technology have been exploited by many people all over the world in several areas, this leads network invasion day by day over the past some years. It is very necessary to find a dominant way to guard the data as it contains highly susceptive information. Today, there are very eternal security such as data encryption, VPN and fire wall. They were good within them. Still they have worth to use but they are lacking to detect the attacks by a freak. In spite of, intrusion detection is a moveable one which can give dynamic protection to the network security in invigilating attacks and slug/counter attacks. Network intrusion detection systems (NIDS) usually adhere one of the three design models. These regular IDS design categories are signature-based, anomaly-based and protocol modelling. Each and every design model has its own strengths and inability and many devices are a conjunction of the three models.

**Signature-based NIDS**

This is the very generic design: roughly all NIDS devices have a firm dependence on signature-based detection at some degree. This technology explication packets for exclusive patterns related to conversant attacks. Signature-based detection is comparatively convenient to unzip, perceive, and update, and also it is suitable at positively identifying known attacks. In spite of, it has one drawback that they may not find out unknown or modified invasions

**Categories of Intrusion Detection System**

### 1. Signature based Detection Systems

Signature based intrusion detection system (SBIDS) based on the known signature. This detection affairs on the continual up to dating signature as it is much emphatic averse known invasions. Also it is unapt to detect the unfamiliar intrusions and novels attacks, as it's general flaw. The only convenience is that it has sublime detection rate than the anomaly intrusion detection[9].

### 2. Anomaly based Detection System

Anomaly based intrusion detection system (ABIDS) has pulled many researchers due to its abilities of detecting novel/fiction attack. There are some sort of unrecognized attack that the machine learning tract is not conscious during exercitation. For this, the Fiction invasion detection system of working is proposed, ABIDS has two prime benefits over SBIDS, the very first is the capability to detect extrinsic and “zero day” invasion. This is done by likening the modest activity with that of deviation from them. Second one is the ordinary activity profile are customized for system, network and here upon building it much stiff for an attacker to know with certitude what activities it can take away without getting find out [11]. The competency of the system depends on how nicely it is instrumented and tested on all protocols. The general drawback of anomaly detection is delimiting its rule set.

### 3. Protocol Modeling

Protocol modeling is executed by examining network burden for uncommon protocol bustling and alarming on traffic with definitive deputed protocols or protocols that are unknown to the system. Protocol modeling relies on various multiple data sources to depict what normal protocol activity is. Generic sources for this data can include protocol specification RFCs, plausible applications that exercise that protocol, and entire analysis of normal network traffic.

## 2. LITERATURE TRACERY

V. chandola et al, They used Hybrid detection framework, Hybrid detection framework is that which depends on data mining taxonomy and bunching techniques [1]. Francesco

Mercaldo, in his research it says it should concern to focus on the use of data mining techniques together with Embattle tree and countenance direct machines for anomaly detection. In the outcome of experiments shows that the algorithm C4.5 has greater ability than SVM in detecting network anomaly and false alarm rate by using 1999 KDD cup data [2]. D. Denning, Algorithm exploits a feature eduction algorithm called symbolic dynamic filtering (SDF)[3]. In SDF, time-series data are saparated for generating symbol sequences that then fabricate probabilistic finite state automata (PFSA) to attend as features for pattern taxonomy [4]. Ugo Fiore et al, in this paper, when noise enhances it is firstly deem the behavior of the leaning method because it could change the ability of extracting accurate rules. Effectiveness is evaluated with 3 metrics: Max rule confidence, Precision and Recall [5]. T. Bhavani et al, they uses Cluster Analysis for Anomaly Detection. Here it has been used a simple K-mean clustering procedure2. K-mean clustering is a simple, flagrant algorithm. It is less computer-profound than many other algorithms, and therefore it is a better choice when the dataset is large [6].T. Bhavani et al, uses Cluster Analysis for Anomaly Detection. The use of a simple K-mean clustering procedure2. K-mean clustering is a simple, well-known algorithm. It is a preferable alternative when the dataset is huge since It is less computer-intensive than many other algorithms [6]. S. Lina et al, for High dimensional dataset these definitive number of cluster given by user are not good calculation, since it leads to non-viable data dealing or its leads to various outlier [7]. B. Thuraisingham, Network intrusion detection systems employ signature-based methods or data mining-based methods which generally rely on labeled training data. Anomaly network intrusion detection method based on Principal Component Analysis (PCA) for data reduction and Fuzzy Adaptive Resonance Theory (Fuzzy ART) for classifier is presented [8]. S. wu et al, New hybrid intrusion detection system using intelligent dynamic swarm based rough set (IDS-IR) for feature election and illuminated swarm optimization for intrusion data classification [9]. B. Singh et al, The approach is studied through simulation and applied to an industrial case study. The outcome propound feasible use for decision making in production management. It praxis Algorithm for the building of a astir network based on work order data [10]. M. Xue et al, used hybrid views for IDS rooted on data mining. The major method is bunching analysis with objective of amended detection rate and reduction in false alarm rate [11]. K.Wankhade et al, in this paper, Anomaly traffic detection system based on the Entropy of network features and Support Vector Machine (SVM) are compared. Afore, a hybrid technique that is annexation of both entropy of network features and recourse vector machine is compared with individual methods [12]. A.Samad, works on spacious comparative study of several anomaly detection programs for identifying different network intrusion [13]. J. Jonathan, They offered a new density-based and grid-based clustering algorithm that is convenient for unsupervised anomaly detection [14].

# 3. PROBLEM RECOGNIZANCE

The term Intrusion detection can be featured by the Intrusion detection system in which the the term Intrusion are unwanted access and all. In the detection system, it detects uneven activity automatically and it secure the network and guard it. The techniques for the detection of the anomalous activity are systematized into two groups:-

## 3.1 Predefined Intrusion Behavior

First it stores the pattern of intrusion or the malevolent behavior and then it judges the intrusion according to the acquired pattern. it can find predestined patterns intrusions and also it has higher detection precision and having low false alarm rate.

## 3.2 Predefined normal behavior

It judges the normal behavior by storing the pattern of user's normal behavior into the database and if the deviation is intense enough, It can say that there is anomalous activity [2], [3], [4].

An Intrusion Detection System (IDS) desires sublimate chastity and detection rate as well as inferior false alarm rate. In general, the performance of IDS is evaluated in term of accuracy (AC), detection rate (DR), and false alarm rate (FAR) as in the following formula:

(1)Accuracy = (TP+TN) / (TP+TN+FP+FN)

(2) Detection Rate = (TP) / (TP+FP)

(3) False Alarm Rate = (FP) / (FP+TN)

**TABLE 1: General Behavior of Intrusion Detection Data**

| Actual | Predicted Normal | Predicted Attack |
|---|---|---|
| **Normal** | TN | FP |
| **Intrusions** | FN | TP |

I. True positive (TP) means attack data detected as attack.

II. True negative (TN) means normal data detected as normal.

III. False positive (FP) means normal data detected as attack.

IV. False negative (FN) means attack data detected as normal.

Let the problem is related to false positive. Here the normal data is detected as intrusion. First it need to perceive how the data is opined as normal or anomaly. It take the data of KDD Cup 1999 data. The 1998 DARPA Intrusion Detection Evaluation Program was prepared and managed by MIT Lincoln Labs. The aim is to survey and evaluate research in intrusion detection. A standard set of data to be audited, which includes a wide variety of intrusions simulated in a military network environment, was provided. The 1999 KDD intrusion detection contest uses a version of this dataset. So after backtracking the data and by comparing the normal classes and anomaly classes it conclude that it takes 41 attributes to check whether the input is of normal class or of anomaly class. The attributes are (duration, protocol_type, service, flag, src_bytes, dst_bytes, land, wrong_fragment, urgent, hot, num_failed_logins, logged_in, num_compromised, root_shell, su_attempted, num_root, num_file_creations, num_shells, num_access_files, num_outbound_cmds, is_host_login, is_guest_login, 'count', srv_count, serror_rate, srv_serror_rate, rerror_rate, srv_rerror_rate, same_srv_rate, diff_srv_rate, srv_diff_ host_rate, dst_host_count, dst_host_srv_count, dst_host_ same_srv_rate, dst_host_diff_srv_rate, dst_host_same _src_port_rate, dst_host_srv_diff_host_rate, dst_host _serror_rate, dst_host_srv_serror_rate, dst_host_ rerror_rate, dst_host_srv_rerror_rate)

'class' {'normal', 'anomaly'}

Now, by this 41 attributes it will be decided whether the data is normal or anomaly.For example: Let us consider four data of KDD Cup 1999 dataset.

Example 2.1
0,udp,other,SF,146,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,13,1,0.00,0.0 0,0.00,0.00,0.08,0.15,0.00,255,1,0.00,0.60,0.88,0.00,0.00,0.00,0.00,0.00, normal

Example 2.2
0,tcp,http,SF,232,8153,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,5,5,0.20,0.20,0.00,0.00,1.00,0.00,0.00,30,255,1.00,0.00,0.03,0.04,0.03,0.01,0.00,0.01, normal

Example 2.3
0,tcp,finger,S0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,24,12,1.00,1.00,0.00,0.00,0.50,0.08,0.00,255,59,0.23,0.04,0.00,0.00,1.00,1.00,0.00,0.00,anomaly

Example 2.4
0,tcp,private,S0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,48,16,1.00,1.00,0.00,0.00,0.14,0.06,0.00,255,15,0.06,0.07,0.00,0.00,1.00,1.00,0.00,0.00, anomaly

The Intrusion detection system is working on 41 attributes to add up the anomalous behavior. There are several values of each attribute & if any of the entry is deviating from the mean value then it is consider as anomaly. There is a problem of false positive in the intrusion detection system. It should be discard count attribute to solve the false positive in IDS. The problem is that, There is a need to find out the attack before time.

The main concern is attributes as there are 41 attributes and it takes time to find the anomalous behavior. There is a need to improve the efficiency by decreasing the number of attributes. The major components mould be kept under mensuration while reshuffling attribute. So, nay than reforming algorithm, for to work on attributes.

## 4. EXPERIMENT AND RESULTS

There is an elevation in the rate of false positive due to count attributes. To evaluate the system the interest is in two general signs of performance: the detection rate and the false positive rate. The false positive rate is defined as the total number of the normal instance that was (incorrectly) classified as intrusions divided by the total number of normal instances. The detection rate is demonstrated as the count of intrusion paradigm detected by the system split up by the total count of intrusion paradigms existent in the test set. These are nice pointers of exccution since they scale what percentage of intrusions the system is able to detect and how many incorrect assortments it make is the process. By calculating these values over the labels data to measure performance.

So, the problem is about authentication. It can improve the authenticity by providing an OTP or one-time password to the user Email address or to the contact number. Since the count attributes is of no use it can remove the count attributes by this. OTP is the best way, hence by using this the problem is able to solve easily.

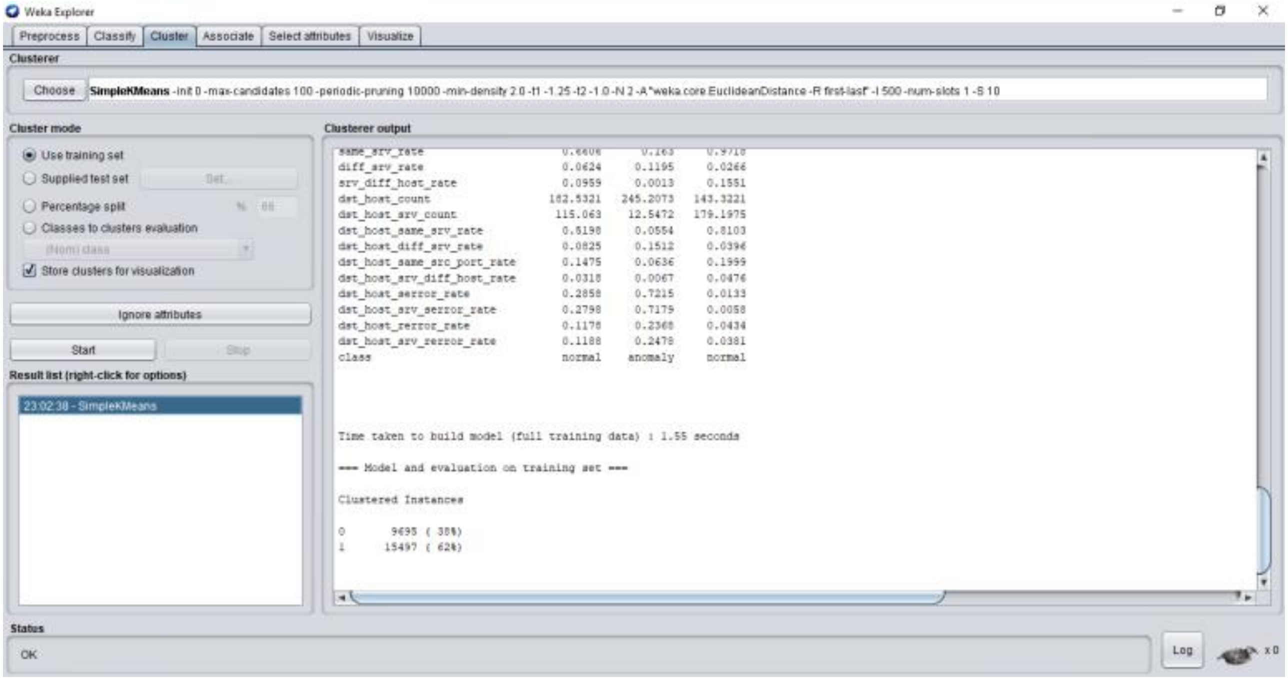
### Algorithm 1: Registration

```
1.  Start
2.  Fill all the required fields in the registration form.
    Including username, email id & passwords.
3.  If the user attempts to submit the incomplete
    registration form
    Show "error message" in dialog box
4.  Else
    Register successful.
5.  Exit.
```

### Algorithm 2: Login

```
1.  Start

2.  Input username & password and fill the
    CAPTCHA/I am not Robot
3.  If username & password are correct then
    Login successfully
4.  Else (for i=1 to i= 10)
        // (Where i is the number of attempts)
    Repeat 1 to 2.
5.  Generate onetime password (OTP) & send it to
    the email id or the provided contact number of
    the user.
6.  If OTP is
    correct
    Repeat 1 to 4
7.  Else
    Show "Wrong OTP".
8.  Exit.
```

Figure 1 depicts the result of system using simple k mean algorithm with count attribute. It can see that it takes 1.55 seconds to complete the clustering.



SimpleKMeans -init 0 -max-candidates 100 -periodic-pruning 10000 -min-density 2.0 -t1 -1.25 -t2 -1.0 -N 2 -A "weka.core.EuclideanDistance -R first-last" -I 500 -num-slots 1 -S 10

● Use training set
○ Supplied test set
○ Percentage split % 66
○ Classes to clusters evaluation
(Nom) class
☑ Store clusters for visualization

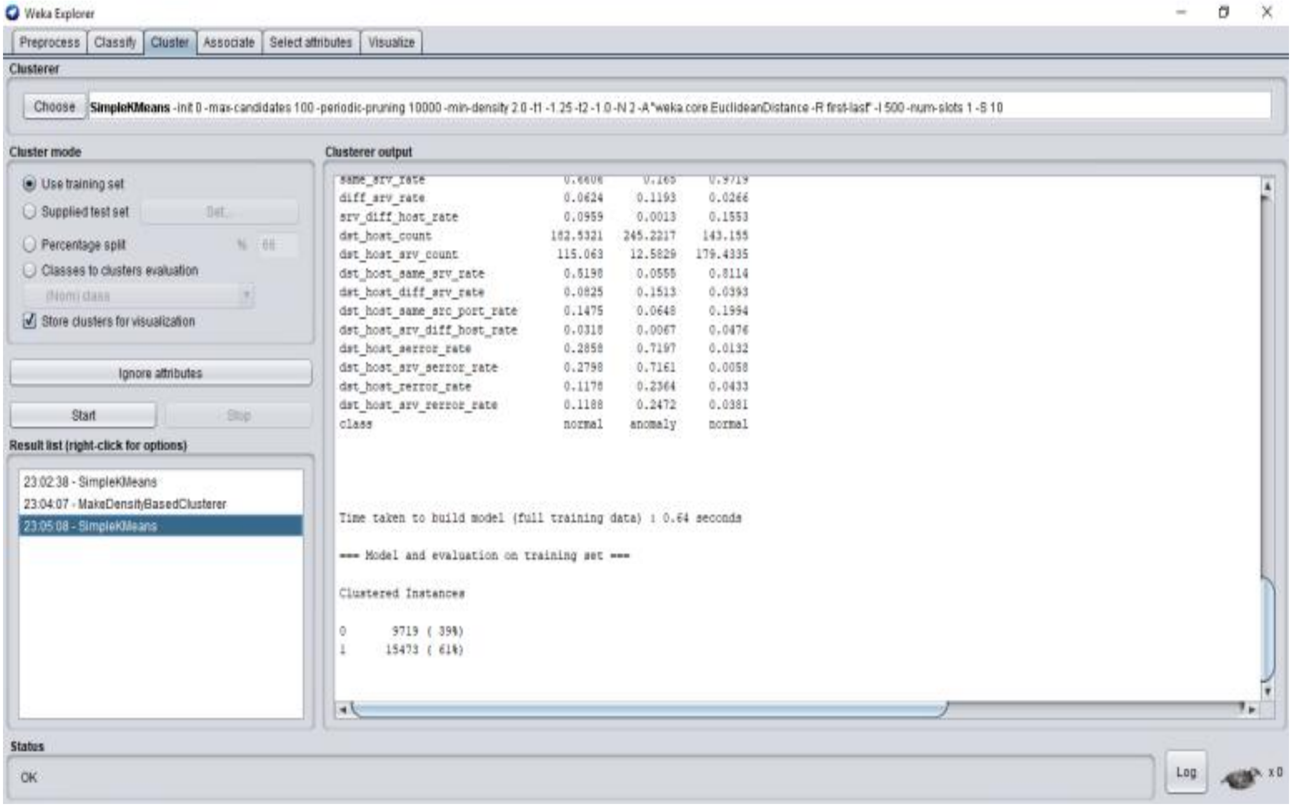
23:02:38 - SimpleKMeans

```
same_srv_rate                  0.6606      0.163     0.9718
diff_srv_rate                  0.0624     0.1195     0.0266
srv_diff_host_rate             0.0959     0.0013     0.1551
dst_host_count               182.5321   245.2073   143.3221
dst_host_srv_count            115.063    12.5472   179.1975
dst_host_same_srv_rate         0.5198     0.0554     0.8103
dst_host_diff_srv_rate         0.0825     0.1512     0.0396
dst_host_same_src_port_rate    0.1475     0.0636     0.1999
dst_host_srv_diff_host_rate    0.0318     0.0067     0.0476
dst_host_serror_rate           0.2858     0.7215     0.0133
dst_host_srv_serror_rate       0.2798     0.7179     0.0058
dst_host_rerror_rate           0.1178     0.2368     0.0434
dst_host_srv_rerror_rate       0.1188     0.2478     0.0381
class                          normal    anomaly     normal



Time taken to build model (full training data) : 1.55 seconds

=== Model and evaluation on training set ===

Clustered Instances

0       9695 ( 38%)
1      15497 ( 62%)
```

Status: OK

**Figure 1: Experiment Result using K mean Algorithm**

Figure 2 depicts the result of system using simple k mean algorithm with count attribute. It can see that it takes 0.64 seconds to complete the clustering.



```
SimpleKMeans -init 0 -max-candidates 100 -periodic-pruning 10000 -min-density 2.0 -t1 -1.25 -t2 -1.0 -N 2 -A "weka.core.EuclideanDistance -R first-last" -I 500 -num-slots 1 -S 10
```

● Use training set
○ Supplied test set
○ Percentage split % 66
○ Classes to clusters evaluation
(Nom) class
☑ Store clusters for visualization

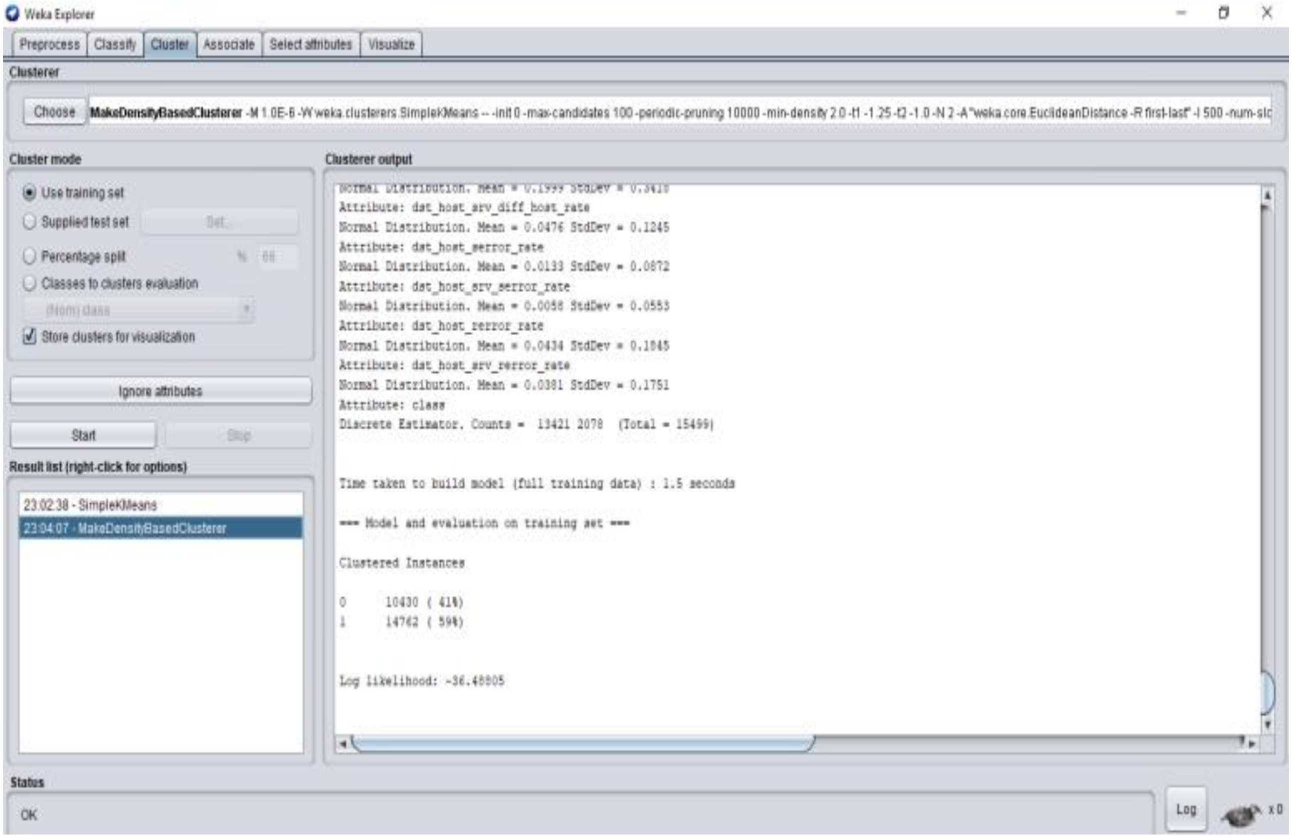
23:02:38 - SimpleKMeans
23:04:07 - MakeDensityBasedClusterer
23:05:08 - SimpleKMeans

```
same_srv_rate                 0.6606     0.165    0.9719
diff_srv_rate                 0.0624    0.1193    0.0266
srv_diff_host_rate            0.0959    0.0013    0.1553
dst_host_count              182.5321  245.2217   143.155
dst_host_srv_count           115.063   12.5829  179.4335
dst_host_same_srv_rate        0.5198    0.0555    0.8114
dst_host_diff_srv_rate        0.0825    0.1513    0.0393
dst_host_same_src_port_rate   0.1475    0.0648    0.1994
dst_host_srv_diff_host_rate   0.0318    0.0067    0.0476
dst_host_serror_rate          0.2858    0.7197    0.0132
dst_host_srv_serror_rate      0.2798    0.7161    0.0058
dst_host_rerror_rate          0.1178    0.2364    0.0433
dst_host_srv_rerror_rate      0.1188    0.2472    0.0381
class                         normal   anomaly    normal




Time taken to build model (full training data) : 0.64 seconds

=== Model and evaluation on training set ===

Clustered Instances

0       9719 ( 39%)
1      15473 ( 61%)
```

Status
OK

**Figure 2: Experiment result using K mean algorithm**

Figure 3 depicts the result of system using Make Density based Clusterer algorithm with count attribute. It can see that it takes 1.5 seconds to complete the clustering.



```
MakeDensityBasedClusterer -M 1.0E-6 -W weka.clusterers.SimpleKMeans -- -init 0 -max-candidates 100 -periodic-pruning 10000 -min-density 2.0 -t1 -1.25 -t2 -1.0 -N 2 -A "weka.core.EuclideanDistance -R first-last" -I 500 -num-slc
```

● Use training set
○ Supplied test set
○ Percentage split % 66
○ Classes to clusters evaluation
(Nom) class
☑ Store clusters for visualization

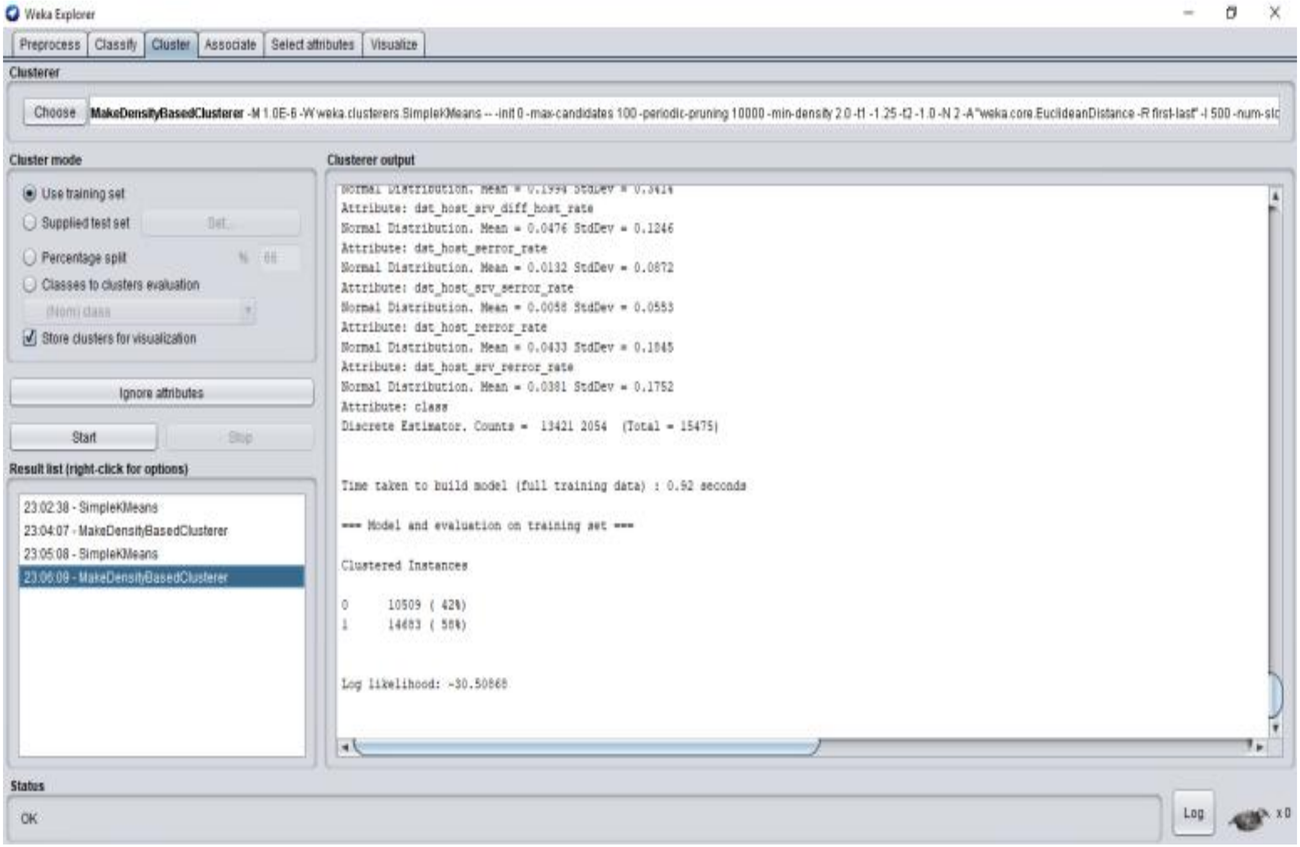
23:02:38 - SimpleKMeans
23:04:07 - MakeDensityBasedClusterer

```
Normal Distribution. Mean = 0.1999 StdDev = 0.3410
Attribute: dst_host_srv_diff_host_rate
Normal Distribution. Mean = 0.0476 StdDev = 0.1245
Attribute: dst_host_serror_rate
Normal Distribution. Mean = 0.0133 StdDev = 0.0872
Attribute: dst_host_srv_serror_rate
Normal Distribution. Mean = 0.0058 StdDev = 0.0553
Attribute: dst_host_rerror_rate
Normal Distribution. Mean = 0.0434 StdDev = 0.1845
Attribute: dst_host_srv_rerror_rate
Normal Distribution. Mean = 0.0381 StdDev = 0.1751
Attribute: class
Discrete Estimator. Counts =  13421 2078  (Total = 15499)


Time taken to build model (full training data) : 1.5 seconds

=== Model and evaluation on training set ===

Clustered Instances

0      10430 ( 41%)
1      14762 ( 59%)


Log likelihood: -36.48805
```

Status
OK

**Figure 3: Experiment result using Make Density based Clusterer algorithm**

Figure 4 depicts the result of system using Make Density Based Clusterer algorithm with count attribute. It can see that it takes 0.92 seconds to complete the clustering.



```
MakeDensityBasedClusterer -M 1.0E-6 -W weka.clusterers.SimpleKMeans -- -init 0 -max-candidates 100 -periodic-pruning 10000 -min-density 2.0 -t1 -1.25 -t2 -1.0 -N 2 -A "weka.core.EuclideanDistance -R first-last" -I 500 -num-slc
```

● Use training set
○ Supplied test set
○ Percentage split % 66
○ Classes to clusters evaluation
(Nom) class
☑ Store clusters for visualization

23:02:38 - SimpleKMeans
23:04:07 - MakeDensityBasedClusterer
23:05:08 - SimpleKMeans
23:06:09 - MakeDensityBasedClusterer

```
Normal Distribution. Mean = 0.1994 StdDev = 0.3414
Attribute: dst_host_srv_diff_host_rate
Normal Distribution. Mean = 0.0476 StdDev = 0.1246
Attribute: dst_host_serror_rate
Normal Distribution. Mean = 0.0132 StdDev = 0.0872
Attribute: dst_host_srv_serror_rate
Normal Distribution. Mean = 0.0058 StdDev = 0.0553
Attribute: dst_host_rerror_rate
Normal Distribution. Mean = 0.0433 StdDev = 0.1845
Attribute: dst_host_srv_rerror_rate
Normal Distribution. Mean = 0.0381 StdDev = 0.1752
Attribute: class
Discrete Estimator. Counts =  13421 2054  (Total = 15475)


Time taken to build model (full training data) : 0.92 seconds

=== Model and evaluation on training set ===

Clustered Instances

0      10509 ( 42%)
1      14683 ( 58%)


Log likelihood: -30.50868
```

OK

**Figure 4: Experiment Result using Make Density based Clusterer Algorithm**

Table 2 depicts the comparison of results between Simlpe K-mean algorithm & Make Density Based Clusterer algorithm.

| Algorithm | Time taken with count attribute | Time taken without count attribute |
|---|---|---|
| Simple k mean value | 1.55 seconds | 0.64 seconds |
| Make Density Based clusterer | 1.5 seconds | 0.92 seconds |

## 5. CONCLUSION

In the current scenario, so many people have suffered a lot from these when they have to open an account online or in internet banking and also because of having more accounts it is very difficult to manage so many passwords in their memory. In case of encountering with three wrong attempts they are blocked by that bank's website for next 24 hours. In this paper, the solution is given for the particular problem. So if this solution is followed by system the problem of false positive can be reduced. By removing the count attribute it can see that the performance of algorithms is improving in a good manner. While comparing the rows of table 2 It can clearly compare that the performance of algorithm is being improving.

## 6. REFERENCES

[1] V. Chandola,A.Banerjee,V.Kumar, "Anomaly detection as a survey" ACM Comput. Surv.41(3)(2009)15:1–15:58.

[2] Francesco Mercaldo, "Identification of anomalies in processes of database alteration" IEEE 2013.

[3] Dorothy E. Denning. "An Intrusion- Detection Model" 1986 IEEE Computer Society Symposium on Research in Security and Privacy, pp 118-31.

[4] S. K. Chaturvedi1 , Prof. Vineet R. , Prof. Nirupama T. "Anomaly Detection in Network using Data mining Techniques" International Journal ISSN 2250-2459 Volume 2, Issue 5, May 2012.

[5] UgoFiore, Francesco, Aniello "Network anomaly detection with the restricted Boltzmann machine" Neurocomputing 122 (2013) 13–23.

[6] T. Bhavani et al., "Data Mining for Security Applications," Proceedings of the 2008 IEEE/IFIP International Conference on Embedded and Ubiquitous Computing Volume 02, IEEE Computer Society, 2008.

[7] Shih-Wei Lina, Kuo-Ching Yingb, Chou-Yuan Leec, Zne-Jung Leed "An intelligent algorithm with feature selection and decision rules applied toanomaly detection" Elsevier 2011.

[8] Bhavani Thuraisingham "Data Mining for Malicious Code Detection and Security Applications" 2009 IEEE/WIC/ACM 2009.

[9] Shu Wu, Member, and Shengrui Wang "Information-Theoretic Outlier Detection for Large-Scale Categorical Data" VOL. 25, NO. 3, MARCH 2013.

[10] Bharat singh,Nidhi Kushwaha and OP vyas "Exploiting Anomaly Detections for high Dimensional data using Descriptive Approach of Data mining" IEEE(ICCT) 2013.

[11] M. Xue , C. Zhu, "Applied Research on Data Mining Algorithm in Network Intrusion Detection," jcai , pp.275-277, 2009 International Joint Conference Artificial Intelligence, 2009.

[12] Kapil Wankhade, Mrudula Gudadhe, Prakash Prasad, "A New Data Mining Based network Intrusion Detection Model", In Proceedings of ICCCT 2010, IEEE, 2010, pp.731-735.

[13] Abdul Samad bin Haji Ismail "A Novel Method for Unsupervised Anomaly Detection using Unlabeled Data" IEEE 2008.

[14] Jonathan J, Davis, Andrew J. Clark "Data preprocessing for anomaly based network intrusion detection: A review" Elsevier 2011.

[15]