To Decrease the Issue of False Alarm Rate by Providing Authentication & Thus Improving the Efficiency of Intrusion Detection System by Comparing the Result of Filtered Clusterer Algorithm & Make-Density Based Clustering Algorithm without Attribute Count

Pratik Jain, Ravikant Kholwal, Muskan Patidar

Abstract*: The Intrusion Detection System sends alerts when it detects doubtful activities while monitoring the network traffic and other known threats. In today's time in the field of Cyber security Intrusion Detection is considered a brilliant topic that could be objective. But it might not remain objectionable for a longer period. For understanding Intrusion Detection, the meaning of Intrusion must be clear at first. According to the oxford's learners dictionary "Intrusion is the act of entering a place that is private or where you may not be wanted". For this article, here it defines intrusion as any un-possessed system or network festivity on one (or more) computer(s) or network(s). Here is the example of a faithful user trying to access the system taking more than the usual trial counts to complete his access to the particular account or trying to connect to an unauthorized remote port of a server. The ex-employee who was being fired lately can provoke intrusion or any authentic worker can also provoke intrusion or any other person from the outside world could perform it. In this clause, the average data is found as the attack which is considered as the case of false positive. In this paper, the main focus is on the illustration and a solution offered for the same problem. Here we are using the KDD CUP 1999 data set. According to the outcome, the anomaly class is the one that has a higher number of counts than this class. Even if it is the true user trying to get access but the outcome is an anomaly due to the high number of counts in the class. This paper introduces a solution for the detection of a true person and eradicates the false positive.

***Keywords: Data Mining, Anomaly Detection System (ADS), K-Means, Ensemble, Detection Rate, False Alarm Rate, False Positive, Clustering*.**

Manuscript received on April 28, 2021.
Revised Manuscript received on May 03, 2021.
Manuscript published on May 30, 2021.

* Correspondence Author

Pratik Jain*, Department of Computer Science, IPS Academy, Institute of Engineering and Science, Indore, India. Email: pratikjain@ipsacademy.org

Ravikant Kholwal, Department of Computer Science, Indian Institute of Information Technology, Design and Manufacturing, Jabalpur, India. Email: rkant4112@gmail.com

Muskan Patidar, Department of Computer Science, IPS Academy, Institute of Engineering and Science, Indore, India. Email: alkapatidar19@gmail.com

I. INTRODUCTION

In 20 years, the technology has evolved a lot and with it, the threat of intrusion is increased too, so the safety of network systems has become a more critical issue. As this took place several times that people have abused the computer technology in various part of the world, this led to network penetration almost every day in previous some years. It has become essential to find the central way to secure the data as it has very crucial information. Nowadays several ways to secure data like VPN, firewall, and data encryption. But they are not much effective when it comes to data intrusion, they lack in identifying the intrusion by an attacker. But the intrusion detection is more dynamic in protecting the network and also inspects the intrusion and slug/counter-attack. Network Intrusion Detection System (NIDS) generally sticks to one of the three design models which are: Signature-based Detection System, Anomaly-based, and Protocol modeling. Each of the models has its capabilities and weaknesses and various devices have a combination of the three models.

Signature-Based NIDS

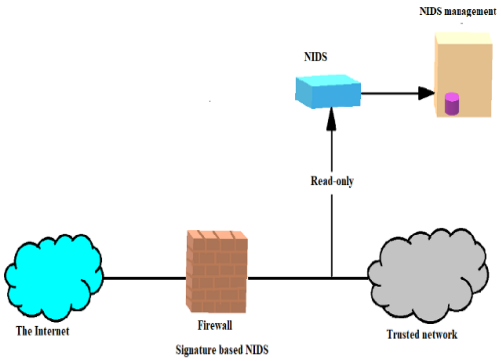

Figure I. (a) Signature-based NIDS

Retrieval Number: 100.1/ijrte.A57550510121
DOI: 10.35940/ijrte.A5755.0510121

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication

To some degree roughly all NIDS devices have a stable dependency on signature-based detection, this a very general design which is shown in figure I.(a). These technology exposition packets for limited patterns related to familiar attacks. It is relatively suitable to unzip, observe and update, and also appropriate for correctly identifying known attacks. Although it is much effective have one drawback that it may not detect the unknown or the modified attack.

Categories of Intrusion Detection System

The below-given figure I.(b) describes the categories in which we can bifurcate IDS (Intrusion Detection System): Protocol modeling, signature-based detection system, Anomaly-based detection system. The Intrusion Detection System sends alerts when it detects doubtful activities while monitoring the network traffic and other known threats.

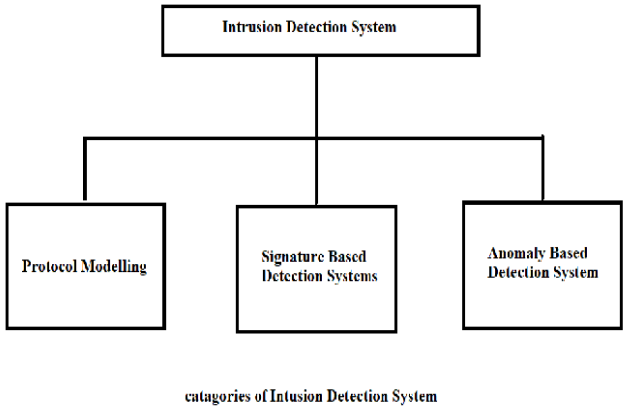

Figure I.(b) shows categories of the intrusion detection system.

A. Protocol Modeling

It is implemented by observing network burden for unusual protocol trafficking and alarming traffic with conclusive appointed protocols that are unknown to the system. For the depiction of normal activity, the protocol modeling depends on multiple data sources. The entire analysis of normal network traffic, protocol specification RFCs, and logical applications that use the same protocol can be considered as the general sources for data.

B. Signature-Based Detection Systems

Signature-Based Detection Systems (SBIDS) based on the recognized signature. This kind of detection is concerned with a regular update of signature as it is much forceful as oppose to known invasion. And also, it's not suitable to detect unusual intrusions and new attacks it is a general drawback. It has a suitable sublime exalted rate than anomaly intrusion detection [3].

C. Anomaly-*Based Detection System*

An anomaly-based intrusion detection system (ABIDS) has drawn numerous researchers because of its competence in discovering novel/fiction attacks. For some of the unnoticed attack which is not considered in the machine-learning area is not conscious while training. ABIDS has two major advantages over SBIDS, the first one is the ability to identify extrinsically and "zero-day" invasion. The process involves the comparison of similar activities and discrete activities from similar ones. The second advantage is the normal activity profile which is customized for a system, network, and hereupon building it strong for the intruder to know with assurance what activities it can take away without getting noticed [8]. The capability of the system can be determined by two factors, those are how well they are instrumented and whether all protocols are tested or not. Delimiting the rule-set is the major drawback of anomaly detection.

II. LITERATURE TRACERY

The paper is comparing the annexation of entropy of network features and resources vector machine both with individual methods are defined as hybrid technique. Anomaly traffic detection system based on comparison of entropy of network features and Support Vector Machine (SVM) [1]. The paper shows that Symbolic dynamic filtering (SDF) can be described as the algorithm which exploits the reduction algorithm [2]. The paper describes about the intrusion data classification, the feature election and illustrated swarn optimization uses a hybrid intrusion detection system which is using intelligence dynamic swarn based rough set (IDR-IR) [3]. B. The paper describes, the data reduction and Fussy Adaptive Resonance Theory (Fussy ART) for the classification, the Anomaly network intrusion detection method which is based on the Principal Component Analysis (PCA) is used [4]. In SDF, to become a feature for pattern taxonomy fabrication of probabilistic finite-state automata (PFSA) is done after time series data are differentiated for generating symbol sequences [5]. The author explains about cluster analysis is used for anomaly detection which uses Simple K-mean clustering Procedure 2. K-mean clustering is considered to be a simple, flagrant, and well-known algorithm that is mostly used for the huge data set as its computer-profound [6]. Here the author by the use of 1999 KDD cup data and focusing on the data mining techniques combined with Embattle tree and the countenance direct machines in his research, represented the outcome of his experiment which displays SVM in detecting anomaly and false alarm rate using 1999 KDD cup data has less ability than the C4.5 Algorithm [7]. In this paper the author uses reduction in false alarm rate and amended detection rate are the objectives with the major method in bunding analysis, he uses hybrid view for IDS rooted on data mining [8]. The paper explains about the two reasons why the high dimension dataset containing the definitive number of clusters provided by the user are not up to the mark evaluated because they lead to impractical data and various irregular data which makes deviation in the evaluation process [9]. For the convenience of the unsupervised anomaly detection the paper offers a new density-based clustering algorithm and a grid-based clustering algorithm [10]. The paper justifies that bunching techniques and data mining taxonomy are the two factors on which hybrid detection framework depends [11]. This paper illuminates, the ability to produce desired output is measured with three matrices: recall, precision, and max rule confidence. The change in the ability to extract accurate rules could be due to the behavior of the leaving method when the noise enhances [12]. The paper provides how the identification of different intrusion can be done; they provide works on spacious comparatively study of several anomaly detection programs [13].

The author clarifies that, the present outcome is practically used for decision making in production management. It is the practical algorithm for the building of the astir network based on work order data [14]. In this paper, it shows the description of all the algorithm used in weka tool [15].

III. PROBLEM RECOGNIZANCE

The word intrusion detection can be divided from the intrusion detection system in which the phased intrusion is considered to be unauthorized access which is not at all a common pattern. The security of the network and identification of uncommon activities are monitored by the detection system. There are two categories in which the techniques to identify the anomalous activities can be separated: -

A. Predefined Normal Behaviour

In this technique, the opposite of the given is done. Instead of a malicious pattern, the normal behaviour is recorded and if any recognizable deviation from the normal behaviour is observed then, it identifies it as anomalous activity [7],[2],[5].

B. Predefined intrusion behaviour

In this kind of intrusion behaviour primarily the patterns of intrusion or malicious activities are being observed and are recorded then they are compared with the acquired pattern and the predetermined patterns which also the main cause of higher precision and rate.

The willingness to keep the detection rate and false alarm rate low and transform the false detection as normal one is being used in Intrusion Detection System. Usually, the performance of IOS is measured in terms of accuracy (AC), detection rate (DR), and false alarm rate (FAR) is below in the 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 Behaviour of Intrusion Detection Data

Prediction	**Actual**	
	Normal	***Intrusion***
Predicted Normal	TN	FN
Predicted Intrusions	FP	TP

1. True positive (TP) implies that intrusion is recognized as an intrusion.
2. True negative (TN) implies that normal is recognized as normal.
3. False-positive (FP) implies that normal is recognized as an intrusion.
4. False-negative (FN) implies that intrusion is recognized as normal.

In this paper, work is done on the KDD cup 1999 data. MIT Lincoln Labs prepared and managed a program in 1998 called DARPA Intrusion Detection Evaluation Program. This paper focuses on the survey and the evaluation/results of the research in intrusion detection. Here we are working on the problem where the normal data is treated as the intrusion which is technically called false positive. This data set is provided with the military network environment in which many intrusions are created to examine the results and this is treated as the standard set of data for examining during this research. A version of the same data set is used in the 1999 KDD Intrusion Detection contest. It is found that there are 41 attributes which on comparing the anomaly class and normal classes determines whether the input belongs to a normal class or an anomaly class. 41 attributes are :- Attribute 1:duration represented as A1, Attribute 2:protocol_type represented as A2, Attribute 3:service represented as A3, Attribute 4:flag represented as A4, Attribute 5:src_bytes represented as A5, Attribute 6:dst_bytes represented as A6, Attribute 7:land represented as A7, Attribute 8:wrong_fragment represented as A8, Attribute 9:urgent represented as A9, Attribute 10:hot represented as A10, Attribute 11:num_failed_logins represented as A11, Attribute 12:logged_in represented as A12, Attribute 13:num_compromised represented as A13, Attribute 14:root_shell represented as A14, Attribute 15:su_attempted represented as A15, Attribute 16:num_root represented as A16, Attribute 17:num_file_creations represented as A17, Attribute 18:num_shells represented as A18, Attribute 19:num_access_files represented as A19, Attribute 20:num_outbound_cmds represented as A20, Attribute 21:is_host_login represented as A21, Attribute 22:is_guest_login represented as A22, Attribute 23: 'count' represented as A23, Attribute 24:srv_count represented as A24, Attribute 25:serror_rate represented as A25, Attribute 26:srv_serror_rate represented as A26, Attribute 27:rerror_rate represented as A27, Attribute 28:srv_rerror_rate represented as A28, Attribute 29:same_srv_rate represented as A29, Attribute 30:diff_srv_rate represented as A30, Attribute 31:srv_diff_ host_rate represented as A31, Attribute 32:dst_host_count represented as A32, Attribute 33:dst_host_srv_count represented as A33, Attribute 34:dst_host_ same_srv_rate represented as A34, Attribute 35:dst_host_diff_srv_rate represented as A35, Attribute 36:dst_host_same_src_port_rate represented as A36, Attribute 37:dst_host_srv_diff_host_rate represented as A37, Attribute 38:dst_host _serror_rate represented as A38, Attribute 39:dst_host_srv_serror_rate represented as A39, Attribute 40:dst_host_ rerror_rate represented as A40, Attribute 41:dst_host_srv_rerror_rate represented as A41,

Class: Class A: normal, Class B: anomaly

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

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication

Attributes	Example 1	Example 2	Example 3	Example 4
A1	0	0	0	0
A2	Udp	tcp	tcp	Tcp
A3	other	http	finger	Private
A4	SF	SF	S0	S0
A5	146	232	0	0
A6	0	8153	0	0
A7	0	0	0	0
A8	0	0	0	0
A9	0	0	0	0
A10	0	0	0	0
A11	0	0	0	0
A12	0	1	0	0
A13	0	0	0	0
A14	0	0	0	0
A15	0	0	0	0
A16	0	0	0	0
A17	0	0	0	0
A18	0	0	0	0
A19	0	0	0	0
A20	0	0	0	0
A21	0	0	0	0
A22	0	0	0	0
A23	13	5	0	48
A24	1	5	24	16
A25	0.00	0.20	12	1.00
A26	0.00	0.20	1.00	1.00
A27	0.00	0.00	1.00	0.00
A28	0.00	0.00	0.00	0.00
A29	0.08	1.00	0.00	0.14
A30	0.15	0.00	0.50	0.06
A31	0.00	0.00	0.00	0.00
A32	255	255	255	255
A33	1	30	59	15
A34	0.00	1.00	0.23	0.06
A35	0.60	0.00	0.04	0.07
A36	0.88	0.03	0.00	0.00
A37	0.00	0.04	0.00	0.00
A38	0.00	0.03	1.00	1.00
A39	0.00	0.01	1.00	1.00
A40	0.00	0.00	0.00	0.00
A41	0.00	0.01	0.00	0.00
Class	Normal	Normal	Anomaly	Anomaly

The 41 attributes are important to keep in consideration and also the time taken by then to detect the malicious or anomalous behaviour. The number of attributes should be reduced to increase efficiency. But before reshuffling any of the attributes the aim of increasing efficiency and the proper detection is kept in mind. As there are 41. attributes to find out the anomalous behaviour. The anomaly is dependent on the comparison with the mean value, if several values deviate from the mean value then the data is anomalous. The false-positive problem in the IDS can be solved y removing the count attribute. The problem is that it is needed to find out an attack before time.

IV. EXPERIMENT AND RESULTS

The count attribute causes an increment in the false-positive rates. For the up-gradation of the system, the main focus is on the two factors for performance, which are detection rate and false alarm rate. The total number of normal instances identified as intrusion divided by the total number of normal instances is termed as false the rate the number of divided intrusions pattern identified by the system down into the total number of intrusion patterns existing in the data set is termed as detection rate. How many mistakes does it does in detection and measurement of what percentage of intrusion the system can detect properly are the two points that indicate good execution? To scale the performance, we can calculate these values on the sample data.

The count attribute has not been used so we can remove the count attribute and replace it with OTP or a one-time password. So, it can improve problems of authentication by giving OTP on the E-mail address contact number of the user. OTP is the top way to which could easily solve this problem.

Algorithm 1: Registration

1. Begin
2. Enter details in a registration form and complete all mandatory fields.
3. If all mandatory information is not filled in the registration form.

display “error message” in a dialogue box

4. Else

Registered successfully.

5. End

Algorithm 2: Login

1. Begin
2. Fill the username and password field and also complete CAPTCHA or I am not a Robot checkbox
3. If both the details are valid then login successful.
4. Else (for i=1 to i=10)

// (i represent how many times the user can attempt to login)

Repeat step 1and 2

5. One-time password is generated and sent to the user's email address or mobile number.
6. If OTP matches

Repeat steps 1 and 4.

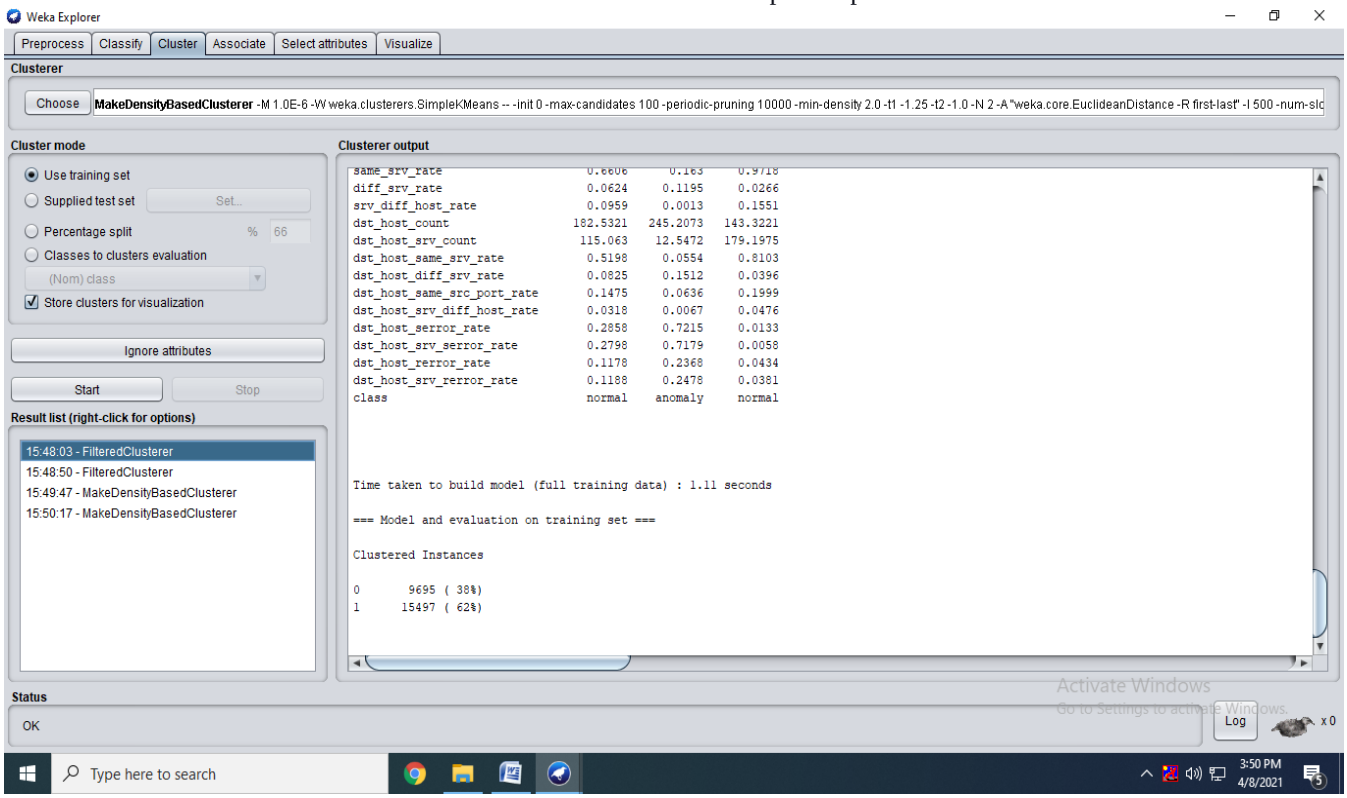

Figure 1. The outcome of an experiment using filter clusterer algorithm (including count attribute)

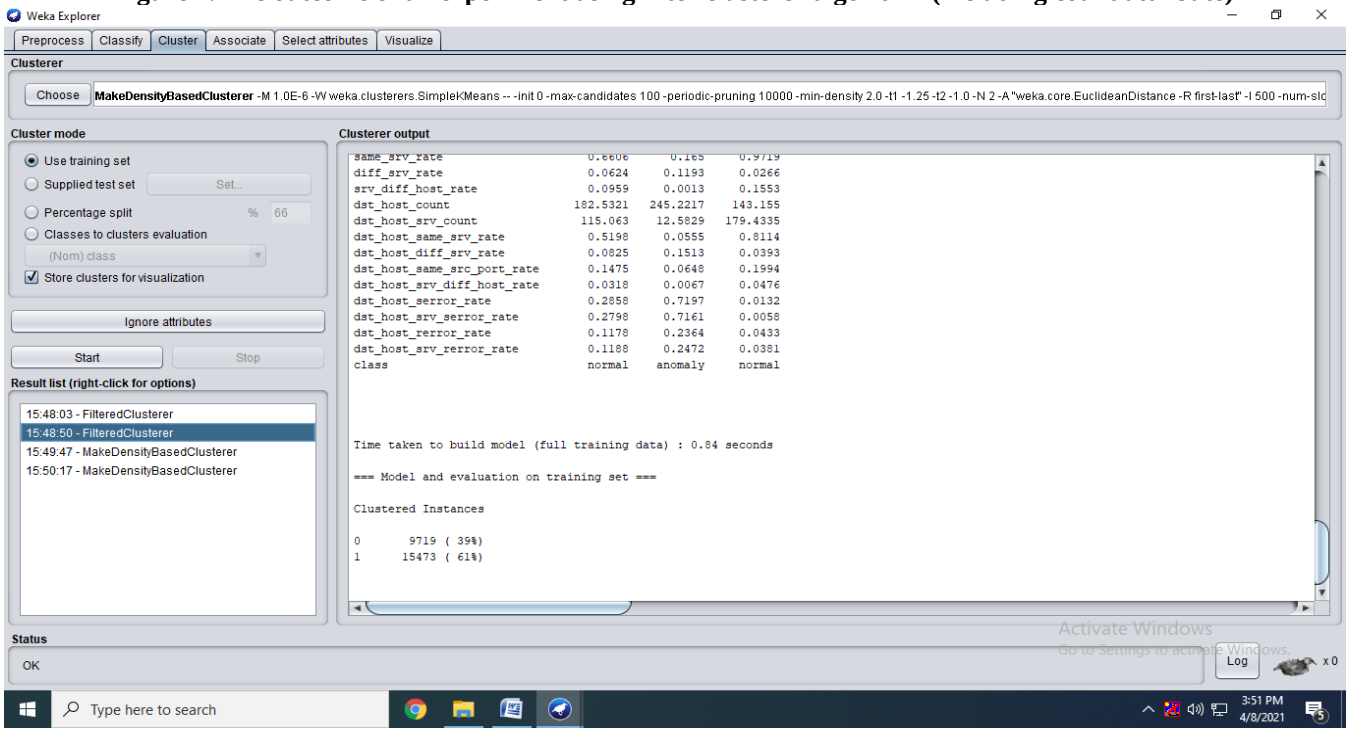

Figure 2. The outcome of an experiment using filtered clusterer algorithm (when count attribute is removed)

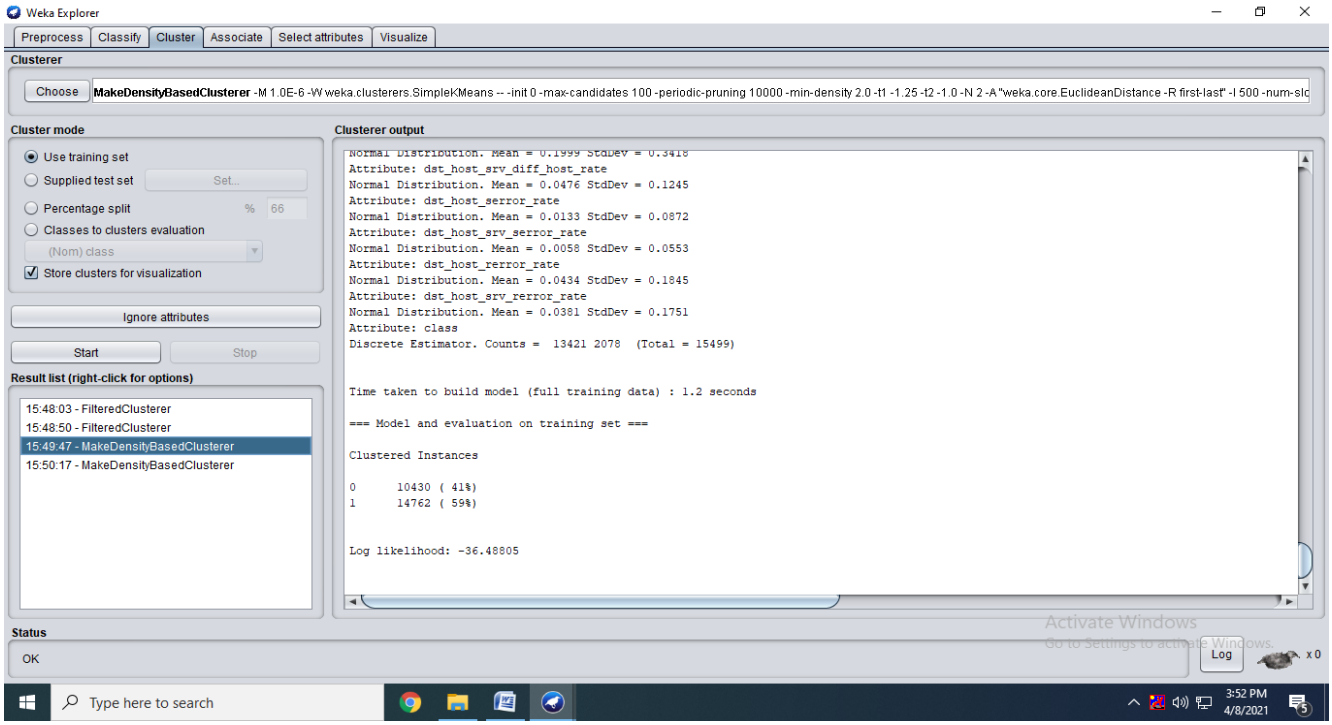

Figure 3. The outcome of an experiment using Make Density-Based Clustering algorithm (including count attribute)

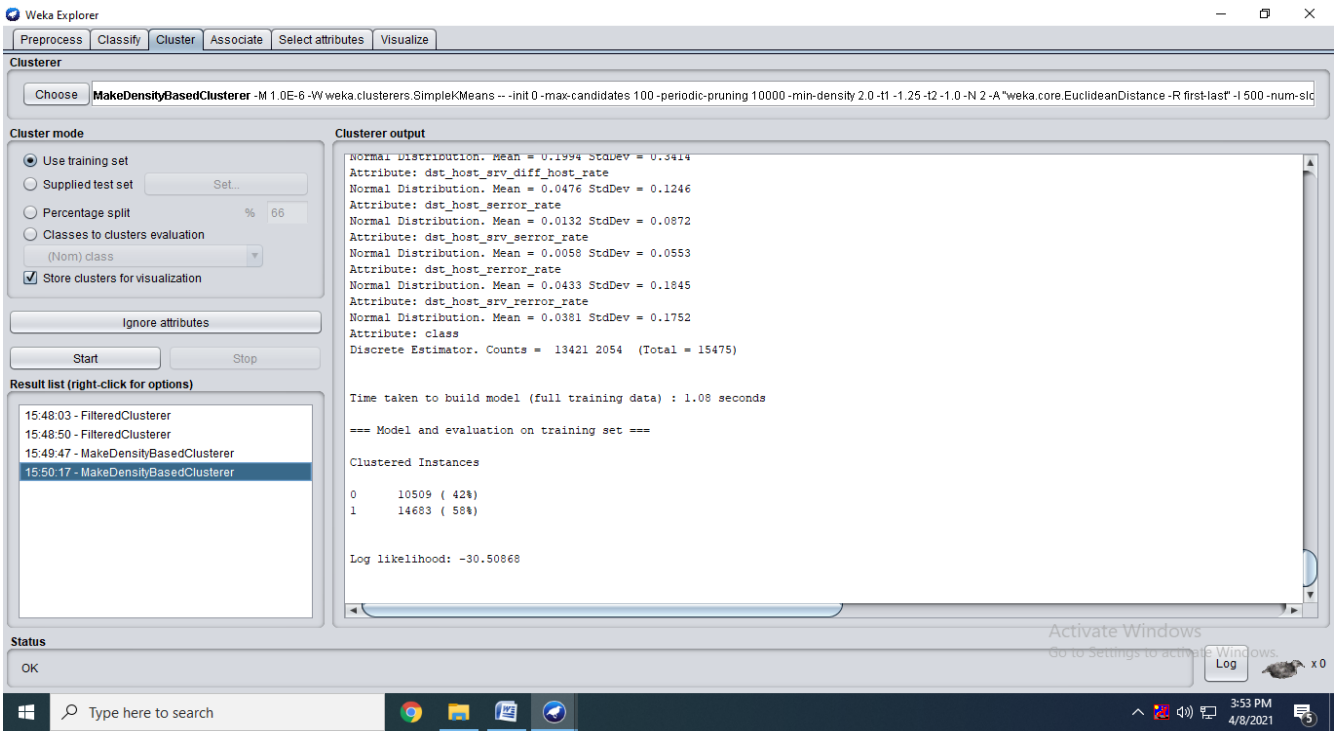

Figure 4. The outcome of an experiment using Make Density-Based Clustering algorithm (removing the count attribute)

Figure 1 displays the outcome of a system using the Filter clusterer algorithm including the count attribute. It shows that the algorithm takes 1.55 seconds to complete the clustering.
Figure 2 displays the outcome of a system using the Filter clusterer algorithm excluding the count attribute. It shows that the algorithm takes 0.64 seconds to complete the clustering.
Figure 3 displays the outcome of a system using the Make Density-Based Clustering algorithm including the count attribute. It shows that the algorithm takes 1.5 seconds to complete the clustering.
Figure 4 displays the outcome of a system using the Make Density-Based Clustering algorithm excluding the count attribute. It shows that the algorithm takes 0.94 seconds to complete the clustering.

Table 2: Final cluster centroids of data filtered with count attribute:

Attribute	Full data(25192.0)	cluster#0 (9695.0)	1 (15497.0)
A1	305.0541	305.0541	162.3509
A2	Tcp	Tcp	tcp
A3	http	Private	http
A4	SF	S0	SF
A5	24330.6282	39374.1000	14919.37572
A6	3491.8472	115.1045	5604.3541
A7	0	0	0
A8	0.0237	0.0175	0.0275
A9	0	0	0.0010
A10	0.198	0.0018	0.3208
A11	0.0012	0.0002	0.0018
A12	0	0	1
A13	0.2279	0	0.3704
A14	0.0015	0.0001	0.0025
A15	0.0013	0.0002	0.0021
A16	0.2498	0.0005	0.4058
A17	0.0147	0.001	0.0233
A18	0.0004	0	0.0006
A19	0.0043	0	0.007
A20	0	0	0
A21	0	0	0
A22	0	0	0
A23	84.5912	166.3895	33.4178
A24	27.6988	9.9234	38.8191
A25	0.2863	0.7253	0.0117
A26	0.2838	0.7212	0.0101
A27	0.1186	0.2467	0.0385
A28	0.1203	0.2486	0.04
A29	0.6606	0.163	0.9718
A30	0.0624	0.1195	0.0266
A31	0.0959	0.0013	0.1551
A32	182.5321	245.2073	143.3221
A33	115.063	12.5472	179.1975
A34	0.5198	0.0554	0.8103
A35	0.0825	0.1512	0.0396
A36	0.1475	0.0636	0.1999
A37	0.0318	0.0067	0.0476
A38	0.2858	0.7215	0.0133
A39	0.2798	0.7179	0.0058
A40	0.1178	0.2368	0.0434
A41	0.1188	0.2478	0.0381
Class	Normal	Anomaly	normal
Time taken to build model (full training data) : 1.11 seconds			
=== Model and evaluation on training set === Clustered Instances 0 9695 (38%) 1 15497 (62%)			

Table 3: Final cluster centroids of data filtered without count attribute:

Attribute	Full data(25192.0)	cluster#0 (9719.0)	1 (15473.0)
A1	305.0541	531.8419	162.6027
A2	Tcp	Tcp	Tcp
A3	http	Private	http
A4	SF	S0	SF
A5	24330.6282	39277.0561	14942.3821

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication

A6	3491.8472	114.8202	5613.047
A7	0	0	0
A8	0.0237	0.021	0.0255
A9	0	0	0.0001
A10	0.198	0.0017	0.3213
A11	0.0012	0.0002	0.0018
A12	0	0	1
A13	0.2279	0	0.371
A14	0.0015	0.0001	0.0025
A15	0.0013	0.0002	0.0021
A16	0.2498	0.0005	0.4064
A17	0.0147	0.001	0.0233
A18	0.0004	0	0.0006
A19	0.0043	0	0.007
A20	0	0	0
A21	0	0	0
A22	0	0	0
A23	27.6988	9.9546	38.8443
A24	0.2863	0.7236	0.0117
A25	0.2838	0.7194	0.0101
A26	0.1186	0.2461	0.0386
A27	0.1203	0.248	0.04
A28	0.6606	0.165	0.9719
A29	0.0624	0.1193	0.0266
A30	0.0959	0.0013	0.1553
A31	182.5321	245.2217	143.115
A32	115.063	12.5829	179.4335
A33	0.5198	0.0555	0.8114
A34	0.0825	0.1513	0.0393
A35	0.1475	0.0648	0.1994
A36	0.0318	0.0067	0.0476
A37	0.2858	0.7197	0.0132
A38	0.2798	0.7161	0.0058
A39	0.1178	0.2364	0.0433
A40	0.1188	0.2472	0.0381
Class	Normal	anomaly	Normal

=== Model and evaluation on training set ===
Clustered Instances
0 9719 (39%)
1 1573 (61%)

Table 4: Final cluster centroids of make density with count attribute:

Attribute	**Full data(25192.0)**	cluster#0 (9695.0)	1 (15497.0)
A1	305.0541	305.0541	162.3509
A2	Tcp	Tcp	Tcp
A3	http	Private	http
A4	SF	S0	SF
A5	24330.6282	39374.1009	14919.3572
A6	3491.8472	115.1045	5604.3541
A7	0	0	0
A8	0.0237	0.0175	0.0276

A9	0	0	0.0001
A10	0.198	0.0018	0.3208
A11	0.0012	0.0002	0.0018
A12	0	0	1
A13	0.2279	0	0.3704
A14	0.0015	0.0001	0.0025
A15	0.0013	0.0002	0.0021
A16	0.2498	0.0005	0.4058
A17	0.0147	0.001	0.0233
A18	0.0004	0	0.0006
A19	0.0043	0	0.007
A20	0	0	0
A21	0	0	0
A22	0	0	0
A23	84.5912	166.3895	33.4178
A24	27.6988	9.9234	38.8191
A25	0.2863	0.7253	0.0117
A26	0.2838	0.7212	0.0101
A27	0.1186	0.2467	0.0385
A28	0.1203	0.2486	0.04
A29	0.6606	0.163	0.9718
A30	0.0624	0.1195	0.0266
A31	0.0959	0.0013	0.1551
A32	182.5321	245.2073	143.3221
A33	115.063	12.5472	179.1975
A34	0.5198	0.0554	0.8103
A35	0.0825	0.1512	0.0396
A36	0.1475	0.0636	0.1999
A37	0.0318	0.0067	0.0476
A38	0.2858	0.7215	0.0133
A39	0.2798	0.7179	0.0058
A40	0.1178	0.2368	0.0434
A41	0.1188	0.2478	0.0381
Class	Normal	anomaly	Normal
Time taken to build model (full training data) : 1.2 seconds			
=== Model and evaluation on training set === Clustered Instances 0 10430 (41%) 1 14762 (59%) Log likelihood: -36.48805			

Table 5: Final cluster centroids of make density without count attribute:

Attribute	Full data(25192.0)	cluster#0 (9719.0)	1 (15473.0)
A1	305.0541	531.8419	162.6027
A2	Tcp	Tcp	Tcp
A3	http	Private	http
A4	SF	S0	SF
A5	24330.6282	39277.0561	14942.3821

A6	3491.8472	114.8202	5613.047
A7	0	0	0
A8	0.0237	0.021	0.0255
A9	0	0	0.0001
A10	0.198	0.0017	0.3213
A11	0.0012	0.0002	0.0018
A12	0	0	1
A13	0.2279	0	0.371
A14	0.0015	0.0001	0.0025
A15	0.0013	0.0002	0.0021
A16	0.2498	0.0005	0.4064
A17	0.0147	0.001	0.0233
A18	0.0004	0	0.0006
A19	0.0043	0	0.007
A20	0	0	0
A21	0	0	0
A22	0	0	0
A23	27.6988	9.9546	38.8443
A24	0.2863	0.7236	0.0117
A25	0.2838	0.7194	0.0101
A26	0.1186	0.2461	0.0386
A27	0.1203	0.248	0.04
A28	0.6606	0.165	0.9719
A29	0.0624	0.1193	0.0266
A30	0.0959	0.0013	0.1553
A31	182.5321	245.2217	143.115
A32	115.063	12.5829	179.4335
A33	0.5198	0.0555	0.8114
A34	0.0825	0.1513	0.0393
A35	0.1475	0.0648	0.1994
A36	0.0318	0.0067	0.0476
A37	0.2858	0.7197	0.0132
A38	0.2798	0.7161	0.0058
A39	0.1178	0.2364	0.0433
A40	0.1188	0.2472	0.0381
Class	Normal	anomaly	Normal

=== Model and evaluation on training set ===
Clustered Instances
0 10509 (42%)
1 14683 (58%)

Table 6: Outcome of filter clusterer algorithm & Make Density Based Clustering algorithm comparison.

Time taken	Output of Algorithms	
	filter clusterer	Make Density Based clustering
Including count attribute	1.11 seconds	1.2 seconds
Excluding count attribute	0.84 seconds	1.08 seconds

V. CONCLUSION

This has become a common problem nowadays. As people have accounts on various platforms and it's very difficult to remember all the passwords. This leads to multiple attempts while accessing any account and in banks, after 3 attempts they are blocked by the bank website for 24 hours which has become trouble for some people. The solution is for this is provided in this paper by excluding the count attribute, this improves the performance positively and their difference can be seen in table 6. The filter clusterer value decreased from 1.11 to 0.84 just by excluding the count attribute and also the difference in make density can be seen as it decreased from 1.2 to 1.08. These differences surely increase the efficiency and reduce the time taken in detection, which leads to a decrease in the false alarm rate. This can be a great help to increase the security as the detection is fast, so there are fewer chances of a breach and delayed detection might lead to some harm to the system.

REFERENCES

1. 1. 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.*
2. *2.* Dorothy E. Denning. 1986 IEEE *"An Intrusion- Detection Model" Computer Society Symposium on Research in Security and Privacy, pp 118-31.*
3. Shu Wu, Member, and Shengrui Wang *"Information-Theoretic Outlier Detection for Large-Scale Categorical Data" VOL. 25, NO. 3, MARCH 2013.*
4. Bhavani Thuraisingham *"Data Mining for Malicious Code Detection and Security Applications" 2009 IEEE/WIC/ACM 2009.*
5. 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.*
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. Francesco Mercaldo, *"Identification of anomalies in processes of database alteration" IEEE 2013.*
8. 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.*
9. 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.*
10. Jonathan J, Davis , Andrew J. Clark "*Data preprocessing for anomaly based network intrusion detection: A review" Elsevier 2011.*
11. V. Chandola,A.Banerjee,V.Kumar, *"Anomaly detection as a survey" ACM Comput. Surv.41(3)(2009)15:1–15:58.*
12. UgoFiore , Francesco, Aniello *"Network anomaly detection with the restricted Boltzmann machine" Neurocomputing 122 (2013) 13–23.*
13. Abdul Samad bin Haji Ismail *"A Novel Method for Unsupervised Anomaly Detection using Unlabeled Data" IEEE 2008.*
14. Bharat Singh, Nidhi Kushwaha and OP Vyas *"Exploiting Anomaly Detections for high Dimensional data using Descriptive Approach of Data mining" IEEE(ICCT) 2013.*
15. *S.* Gnanapriya, *R. Adline Freeda, M. Sowmiya"Evaluation of Clustering Capability Using Weka Tool" International Journal of Innovations in Engineering and Technology (IJIET)2017.*

AUTHORS PROFILE

Pratik Jain, Completed his Master of Engineering from IPS Academy Institute of Engineering and science. Eight Papers are published in different journals.

Ravikant Kholwal, is a final year undergraduate student pursuing Computer Science and Engineering at the Indian Institute of Information Technology, Design, and Manufacturing, Jabalpur. He has developed several projects in the field of Android Development One of his projects is Smart Parking. In this app, Users register their accounts and can book parking slots before reaching their respective destination. They can also pay for their Parking slot using UPI. The frontend part is developed using XML and Java, and in the backend, Firebase is implemented. When the car comes to the parking slot, the sensor sends information to the NodeMCU which in turn sends information to the Firebase database.

Muskan Patidar, is a third-year Computer Science and Engineering student pursuing his undergraduate degree from IPS Academy, Institute of Engineering and Science (Indore). Her area of interest includes web development, web designing and machine learning. She has done an online certification course in python for the gaining more skills in machine learning. She is currently looking forward to the project based on the agricultural problems and help them optimize the production.