COMPREHENSIVE STUDY OF KDD99 DATASET AND DATA MINING TOOLS FOR INTRUSION DETECTION

By

KAMINI NALAVADE * B.B. MESHRAM **

* Research Scholar, Computer Engineering Department, VJTI, Matunga, Mumbai, India.
** Professor & Head, Computer Engineering Department, VJTI, Matunga, Mumbai, India.

ABSTRACT

Due to extensive growth of the Internet and increasing availability of tools and methods for intruding and attacking networks, intrusion detection has become a critical component of network security parameters. Intrusion detection in large data is one of the major challenges for the researchers in this area. Anomaly detection using data mining techniques has attracted the attention of many researchers to overcome the weakness of signature-based IDSs in detecting novel attacks and KDDCUP'99 is the mostly widely used data set for the evaluation of these systems. In this paper, we have conducted a comprehensive study and statistical analysis on KDD dataset. The authors also provide description of features and instances of the dataset. The another important challenge for the researchers in this area is to select an appropriate data mining tool for the analysis. The paper disusses two important and popular tools in this area, weka, Oracle data mining and tanagara. The authors hope that the study carried out in this paper is useful for the reasearcheres in the area of intrusion detection.

Keywords: Intrusion, Security, Dataset, Data Mining, KDDcup.

INTRODUCTION

The continuous improvements in technology have made the use of computers easy for gathering and sharing information using the Internet. Given the different type of attacks like Denial of Service, Probing, Remote to Local, User to Root and others, it is a challenge for any intrusion prevention system to detect a wide variety of attacks. The goal of intrusion detection systems is to automatically detect attack from the continous stream of network data traffic. The research in the intrusion detection field has been mostly focused on anomaly-based and misuse-based detection techniques for a long time. While misuse-based detection is generally favored in commercial products due to its predictability and high accuracy, in academic research, anomaly detection is typically conceived as a more powerful method due to its theoretical potential for addressing novel attacks.

Statistics was one way of analyzing the available data and obtaining results. But with the growing amount of data and advent of computing in various fields, extracting useful information from this data using various sophisticated mathematical models and statistics became possible. This extraction of useful information from large high dimensional databases came to be known as "Data Mining". Data mining is the analysis of observational dataset to find unsuspected relationship and to summarize large amounts of data which is useful in proactive decision making. Data Mining delivers new algorithms that can automatically sift deep into your data at the individual record level to discover patterns, relationships, factors, clusters, associations, profiles, and predictions—that were previously "hidden". Using normal reports, Data mining can produce decisions and create alerts when action is required. In order to apply the data mining techniques to information security, we require datasets. The most popular datasets in the area of intrusion detection are Darpa, KDDcup99 and NSL KDD dataset [2]. The authors used a commonly applied dataset in information security research: The network intrusion dataset from the KDD archive popularly referred to as the KDD 99 Cup dataset. Many researchers have contributed their efforts to analyze the dataset by different

techniques. This paper is an analysis of 10% of KDDcup'99 training dataset based on intrusion detection.

In this paper, Section 1 describes the KDDcup99 dataset with respect to features, attacks and instances. In section-2, we describe the problems in KDD dataset. Section-3 provides details about data mining tools used for the analysis of KDD dataset. Finally the experimentation and results followed by conclusion.

1. KDDCUP 99 Dataset

In this section, we give a detailed analysis of the KDDcup99 dataset. We explain the formation of dataset, features and attacks in the dataset as given below:

1.1 History

The 1998 DARPA Intrusion Detection Evaluation Program was prepared and managed by MIT Lincoln Labs. The objective was 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.

Lincoln Labs set up an environment to acquire nine weeks of raw TCP dump data for a local-area network (LAN) simulating a typical U.S. Air Force LAN. They operated the LAN as if it were a true Air Force environment, but peppered it with multiple attacks. The raw training data was about four gigabytes of compressed binary TCP dump data from seven weeks of network traffic. This was processed into about five million connection records. Similarly, the two weeks of test data yielded around two million connection records[4].

A connection is a sequence of TCP packets starting and ending at some well defined times, between which data flows to and from a source IP address to a target IP address under some well defined protocol. Each connection is labeled as either normal, or as an attack, with exactly one specific attack type. Each connection record consists of about 100 bytes.

Since 1999, KDD'99 [3] has been the most wildely used data set for the evaluation of anomaly detection methods. This data set is prepared by Stolfo et al. [2] and is built based on the data captured in DARPA'98 IDS evaluation program. DARPA'98 is about 4 gigabytes of compressed raw (binary) TCP dump data of 7 weeks of network traffic, which can be processed into about 5 million connection records, each with about 100 bytes. The two weeks of test data have around 2 million connection records. KDD training dataset consists of approximately 4,900,000 single connection vectors each of which contains 41 features and is labeled as either normal or an attack, with exactly one specific attack type. The simulated attacks fall in one of the following four categories:

- *Denial of Service Attack (DoS)*: is an attack in which the attacker makes some computing or memory resource too busy or too full to handle legitimate requests, or denies legitimate users access to a machine.
- *User to Root Attack (U2R)*: is a class of exploit in which the attacker starts out with access to a normal user account on the system (perhaps gained by sniffing passwords, a dictionary attack, or social engineering) and is able to exploit some vulnerability to gain root access to the system.
- *Remote to Local Attack (R2L)*: occurs when an attacker who has the ability to send packets to a machine over a network, but who does not have an account on that machine, exploits some vulnerability to gain local access as a user of that machine.
- *Probing Attack:* is an attempt to gather information about a network of computers for the apparent purpose of circumventing its security controls. It is important to note that the test data is not from the same probability distribution as the training data, and it includes specific attack types not in the training data which make the task more realistic. Some intrusion experts believe that most novel attacks are variants of known attacks and the signature. The datasets contains a total of 22 training attack types. The attacks in each class are as shown in

S.N	Class	Attack Types
1	DOS	Back, Land, Neptune,pod, smurf, Teardrop,
2	U2R	Buffer_overflow, loadmodule, perl, rootkit
3	R2L	ftp_write, guess_passwd, imap,multihop, phf, spy,warezlient, warezmaster
4	Probe	IPsweep,nmap, satan,portsweep

Table 1. Classes of Attacks

Table 1.

1.2 Dataset Features

Attributes in the KDD datasets had all forms – continuous, discrete, and symbolic, with significantly varying resolution and ranges. There are 41 features for each connection record that are divided into discrete sets and continuous sets according to the feature values. The 41 attributes and their types are as shown in Figure1.

KDD'99 features can be classified into three groups:

- *Basic features:* this category encapsulates all the attributes that can be extracted from a TCP/IP connection. Most of these features lead to an implicit delay in detection.
- *Traffic features*: this category includes features that are computed with respect to a window interval and is divided into two groups:
  - *"same host" features*: examine only the connections in the past 2 seconds that have the same destination host as the current connection, and calculate statistics related to protocol behavior,service, etc.
  - *"same service" features:* examine only the connections in the past 2 seconds that have the same service as the current connection.The two aforementioned types of "traffic" features are called time-based. However, there are several slow probing attacks that scan the hosts (or ports) using a much larger time interval than 2 seconds, for example, one in every minute. As a result, these attacks do not produce intrusion patterns with a time window of 2 seconds. To solve this problem, the "same host" and "same service" features are re-calculated, but based on the connection window of 100 connections, rather than a time window of 2 seconds. These features are called connection-based traffic features.
- *Content features*: DoS and Probing attacks involve many connections to some host(s) in a very short period of time; however, the R2L and U2R attacks are embedded in the data portions of the packets, and normally involves only a single connection. To detect these kinds of attacks, we need some features to be able to look for suspicious behavior in the data portion, e.g., number of failed login attempts. These features are called content features.

Attribute	Category	Attribute	Category
duration	Continue	is_guest_login	Discrete
protocol_type	Discrete	count	Continue
service	Discrete	srv_count	Continue
flag	Discrete	serror_rate	Continue
src_bytes	Continue	srv_serror_rate	Continue
dst_bytes	Continue	rerror_rate	Continue
land	Discrete	srv_rerror_rate	Continue
wrong_fragment	Continue	same_srv_rate	Continue
urgent	Continue	diff_srv_rate	Continue
hot	Continue	srv_diff_host_rate	Continue
num_failed_logins	Continue	dst_host_count	Continue
logged_in	Discrete	dst_host_srv_count	Continue
lnum_compromised	Continue	dst_host_same_srv_rate	Continue
lroot_shell	Continue	dst_host_diff_srv_rate	Continue
lsu_attempted	Continue	dst_host_same_src_port_rate	Continue
lnum_root	Continue	dst_host_srv_diff_host_rate	Continue
lnum_file_creations	Continue	dst_host_serror_rate	Continue
lnum_shells	Continue	dst_host_srv_serror_rate	Continue
lnum_access_files	Continue	dst_host_rerror_rate	Continue
lnum_outbound_cmds	Continue	dst_host_srv_rerror_rate	Continue
is_host_login	Discrete	label	Discrete

Figure 1. Features of KDD CUP Dataset

The protocols that are considered in KDD dataset are TCP, UDP, and ICMP that are explained below:

- *TCP:* TCP stands for "Transmission Control Protocol". TCP is an important protocol of the Internet Protocol Suite at the Transport Layer which is the fourth layer of the OSI model. It is a reliable connection-oriented protocol which implies that data sent from one side is sure to reach the destination in the same order. TCP splits the data into labeled packets and sends them across the network. TCP is used for many protocols such as HTTP and Email Transfer.
- *UDP:* UDP stands for "User Datagram Protocol". It is similar in behavior to TCP except that it is unreliable and connection-less protocol. As the data travels over unreliable media, the data may not reach in the same order, packets may be missing and duplication of packets is possible. This protocol is a transaction-oriented protocol which is useful in situations where delivery of data in certain time is more important than losing few packets over the network. It is useful in situations where error checking and correction is possible in application level.

• *ICMP*: ICMP stands for "Internet Control Message Protocol". ICMP is basically used for communication between two connected computers. The main purpose of ICMP is to send messages over networked computers. The ICMP redirects the messages and it is used by routers to provide the up-to-date routing information to hosts, which initially have minimal routing information. When a host receives an ICMP redirect message, it will modify its routing table according to the message.

1.3 Dataset Records

In order to know how to read the data from the audit data, we need to analyze how the audit da ta is being recorded. This dataset contains a standard set of data to be audited, which includes a wide variety of intrusions

Viewer

Relation: kdd_cup_1999-weka.filters.supervised.instance.SpreadSubsample-M0.0-X0.0-S1-weka.filters.supervised.instan...

No.	duration Numeric	protocol_type Nominal	service Nominal	flag Nominal	src_bytes Numeric	dst_bytes Numeric	land Nominal	wrong_fragment Numeric	urgent Numeric	hc Num
1	0.0	tcp	other	REJ	0.0	0.0	0	0.0	0.0	
2	0.0	tcp	private	REJ	0.0	0.0	0	0.0	0.0	
3	0.0	tcp	private	REJ	0.0	0.0	0	0.0	0.0	
4	0.0	tcp	other	REJ	0.0	0.0	0	0.0	0.0	
5	0.0	icmp	eco_i	SF	8.0	0.0	0	0.0	0.0	
6	0.0	icmp	eco_i	SF	8.0	0.0	0	0.0	0.0	
7	0.0	tcp	private	SH	0.0	0.0	0	0.0	0.0	
8	1.0	tcp	rje	RSTR	0.0	0.0	0	0.0	0.0	
9	0.0	tcp	private	REJ	0.0	0.0	0	0.0	0.0	
10	0.0	tcp	other	S0	0.0	0.0	0	0.0	0.0	
11	0.0	tcp	private	RSTR	0.0	0.0	0	0.0	0.0	
12	0.0	tcp	other	REJ	0.0	0.0	0	0.0	0.0	
13	0.0	tcp	private	SH	0.0	0.0	0	0.0	0.0	
14	0.0	tcp	other	REJ	0.0	0.0	0	0.0	0.0	
15	0.0	icmp	eco_i	SF	8.0	0.0	0	0.0	0.0	
16	0.0	icmp	eco_i	SF	8.0	0.0	0	0.0	0.0	
17	831.0	tcp	other	RSTR	1.0	0.0	0	0.0	0.0	
18	0.0	tcp	other	REJ	0.0	0.0	0	0.0	0.0	
19	0.0	tcp	private	REJ	0.0	0.0	0	0.0	0.0	
20	0.0	tcp	private	RSTR	0.0	0.0	0	0.0	0.0	
21	0.0	icmp	eco_i	SF	18.0	0.0	0	0.0	0.0	
22	0.0	tcp	private	REJ	0.0	0.0	0	0.0	0.0	
23	0.0	icmp	eco_i	SF	8.0	0.0	0	0.0	0.0	
24	0.0	tcp	private	RSTR	0.0	0.0	0	0.0	0.0	
25	0.0	tcp	other	REJ	0.0	0.0	0	0.0	0.0	
26	0.0	tcp	private	REJ	0.0	0.0	0	0.0	0.0	
27	0.0	tcp	other	REJ	0.0	0.0	0	0.0	0.0	
28	0.0	icmp	eco_i	SF	8.0	0.0	0	0.0	0.0	
29	0.0	icmp	eco_i	SF	8.0	0.0	0	0.0	0.0	
30	0.0	tcp	other	S0	0.0	0.0	0	0.0	0.0	
31	0.0	tcp	private	RSTR	0.0	0.0	0	0.0	0.0	
32	0.0	icmp	eco_i	SF	8.0	0.0	0	0.0	0.0	
33	1.0	tcp	private	RSTR	0.0	0.0	0	0.0	0.0	
34	0.0	icmp	eco_i	SF	18.0	0.0	0	0.0	0.0	

Figure 2. KDD CUP Dataset View

simulated in a military network environment. Figure 2. shows the KDD cup Dataset view

- kddcup.data.zip The full data set (18M; 743M Uncompressed)
- kddcup.data_10_percent.zip A 10% subset. (2.1M; 75M Uncompressed)
- kddcup.newtestdata_10_percent_unlabeled.zip (1.4M; 45M Uncompressed)
- kddcup.testdata.unlabeled.zip (11.2M; 430M Uncompressed)
- kddcup.testdata.unlabeled_10_percent.zip (1.4M;45M Uncompressed) [4]

The audit data is processed for data mining purpose and is split into two files, the training set which contains around five million rows and the test set with 10% of the training set. KDD dataset is divided into labeled and unlabeled records. Each labeled record consisted of 41 attributes (features) [2] and one target value. Target value indicated the attack category name.

There are around 5 million (4,898,430) records in the labeled dataset, which was used for training all classifier models discussed in this paper. A second unlabeled dataset (311,029 records) is provided as testing data [3]. The total number of records in the original labeled training dataset is 972,780 for Normal, 41,102 for Probe, 3,883,370 for DoS, 52 for U2R, and 1,126 for R2L attack classes.

2. Problems IN KDD99 Dataset

As it is mentioned in the previous section, KDD'99 is built based on the data captured in DARPA'98 which has been criticized by McHugh [4], mainly because of the characteristics of the synthetic data. As a result, some of the existing problems in DARPA'98 remain in KDD'99. In the following, we review the issues in KDD'99[1].

- The KDD 1999 Cup dataset has a very large number of duplicate records. The huge number of redundant records, which causes the learning algorithms to be biased towards the frequent records, and thus prevent them from learning unfrequent records which are usually more harmful to networks such as U2Rand R2L attacks.
- Traffic collectors such as TCPdump, which is used in DARPA'98, are very likely to become overloaded and drop packets in heavy traffic load. However, there was no examination to check the possibility of the dropped packets.
- Portnoy et al.observed that the distribution of the attacks in the KDD data set is very uneven which made crossvalidation very difficult. Many of these subsets contained instances of only a single type. For example, the 4th, 5th, 6th, and 7th, 10% portions of the full data set contained only smurf attacks, and the data instances in the 8th subset were almost entirely neptune intrusions.
- Smurf and neptune are two types of DoS attacks that constitute over 71% of the testing data set which completely affects the evaluation.
- Smurf and neptune attacks in the KDD training data set generate large volumes of traffic, they are easily detectable by other means and there is no need of using anomaly detection systems to find these attacks.

6) There is no exact definition of the attacks. For example, probing is not necessarily an attack type unless the number of iterations exceeds a specific threshold. Similarly, a packet that causes a buffer overflow is not always representative of an attack. Overestimation of the performance of some anomaly is identified by anomalous source IP addresses or anomalies in the TCP window size field. [1]

3. Data Mining Tools for the Analysis of KDDCUP99 Dataset

Different types of data mining tools are available and each have its own merits and demerits, for the analysis of 10% of KDD 99 training dataset. We carried out association rule mining and clustering on the KDD dataset using following tools. As discussed in the previous section, we know that KDD dataset consists of contiouns and discrete value. Association rule mining takes different forms.

3.1 Association Rule Mining for Binary Value

For binary weighted value, it is easy to find out the frequent item set. The frequent item set is generated by the Apriori algorithm from a large number of data set. In association

rules mining, weights are considered as the highest priority. In those rules, Apriori algorithm can be imagined as two steps. Firstly it generates candidate sets. Secondly, it prunes the entire non-frequent item set after each step using the minimum support and the weight of the item from the data base. Many item sets could be eliminated by the pruning process which are not frequent.

3.2 Association Rule Mining for Continuous Value

If a particular attribute takes a value in the range [0...1] it is taken as a continuous attribute in the Tanagra tool. This could be taken as Fuzzy data and hence fuzzy weighted Association rule mining as described in fuzzy mining paper could be used here. The weight of fuzzy data can be defined as Fuzzy Item Weight (FIW). Now Fuzzy Item set Transaction Weight (FITW) is the aggregate weights of all the fuzzy sets associated with the items in the item set present in a single transaction. From this FITW, support and Confidence value can be calculated as per generalizing the notion of support.

3.3 Association Rule Mining for Discrete value

The normalization of the data becomes very difficult if the range of values that an attribute in the data set can take is very large . The traditional approach to deal with this type of data in association analysis is to convert each value into a set of binary values. The discrete attributes are normalized i.e. we find a set of thresholds that can be used to convert the attributes into a categorical variable. This kind of normalization affects the accuracy of the rule generation technique which may lead to higher misclassification rate.

The authors have carried out the analysis of attack on 10% of KDD 99 training dataset using K-means clustering technique and association rule mining. We had clustered the training dataset which consisted of 494,019 records and made 3 clusters and discovered association rules with dicrete features. This paper concentrates on Weka, and Tangara tool. The details are given below:

Weka

Weka is a collection of machine learning algorithms for data mining tasks. The algorithms can either be applied directly to a dataset or called from your own Java code. Weka contains tools for data pre-processing, classification, regression, clustering, association rules, and visualization. It is also well suited for developing new machine learning schemes. Some example datasets are included in the Weka distribution. For example, a jarfile containing 37 classification problems, originally obtained from the UCI repository (datasets-UCI.jar, 1,190,961 Bytes). A jarfile containing 37 regression problems, obtained from various sources (datasets-numeric.jar, 169,344 Bytes). A jarfile containing 6 agricultural datasets obtained from agricultural researchers in New Zealand (agridatasets.jar, 31,200 Bytes)[5]. Figure 3. shows Apriori algorithm result with Weka

3.4 Association Rule Mining using Weka explorer

Result of Apriori algorithm on KDDdataset discrete features is shown in Figure 3. As shown in figure, Weka can be used to preprocess data, feature selection and construction and apply data mining techniques such as clustering, classification and rule mining. Figure 4. shows the Clustering result with Weka K-means clustering using Weka explorer

Tanagara

TANAGRA is a free DATA MINING software for academic and research purposes. It proposes several data mining methods from exploratory data analysis, statistical learning, machine learning and databases area. This project is the successor of SIPINA which implements various supervised learning algorithms, especially an

Figure 3. Apriori algorithm result with Weka

interactive and visual construction of decision trees. TANAGRA is more powerful, it contains some supervised learning but also other paradigms such as clustering, factorial analysis, parametric and nonparametric statistics, association rule, feature selection and construction algorithms.TANAGRA is an "open source project" as every researcher can access to the source code, and add his own algorithms, as far as he agrees and conforms to the software distribution license. The main purpose of Tanagra project is to give researchers and students an easy-to-use data mining software, conforming to the present norms of the software development in this domain (especially in the design of its GUI and the way to use it), and allowing to analyse either real or synthetic data. Figure 5. shows the Apriori Result with Tanagara

The second purpose of TANAGRA is to propose to researchers an architecture allowing them to easily add their own data mining methods and to compare their perfor mances. TANAGRA acts more as an experimental platform in order to let them go to the essentials of their work, dispensing them to deal with the unpleasant part in the programmation of these kind of tools: the data management. The third and last purpose, in direction of novice developers, consists in diffusing a possible methodology for building this kind of software. They should take advantage of free access to source code, to look how this sort of software is built, the problems to avoid, the main steps of the project, and which tools and code libraries to use for. In this way, Tanagra can be considered as a pedagogical tool for learning programming techniques. [6] Figure 6. shows the K-means clustering Result.

- Apriori Algorithm Results on KDD dataset using

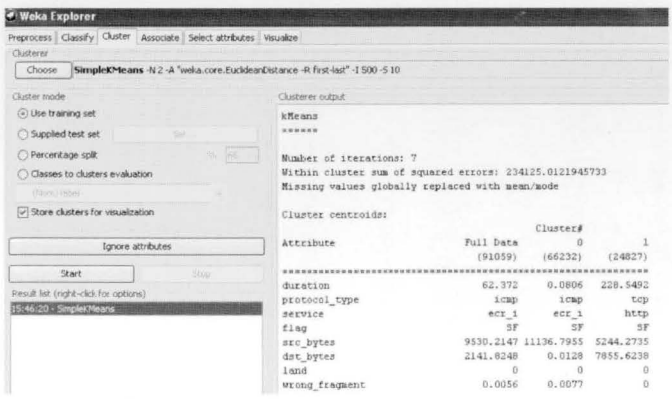

Figure 4. K-means Clustering result with Weka

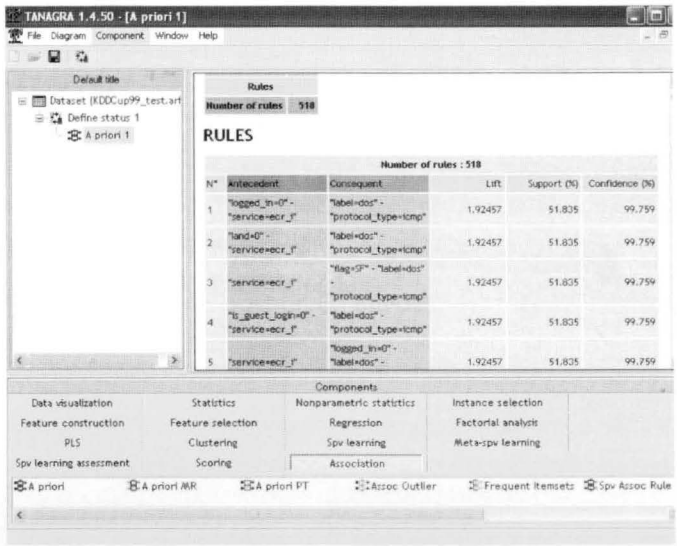

Figure 5. Apriori Result with Tanagara

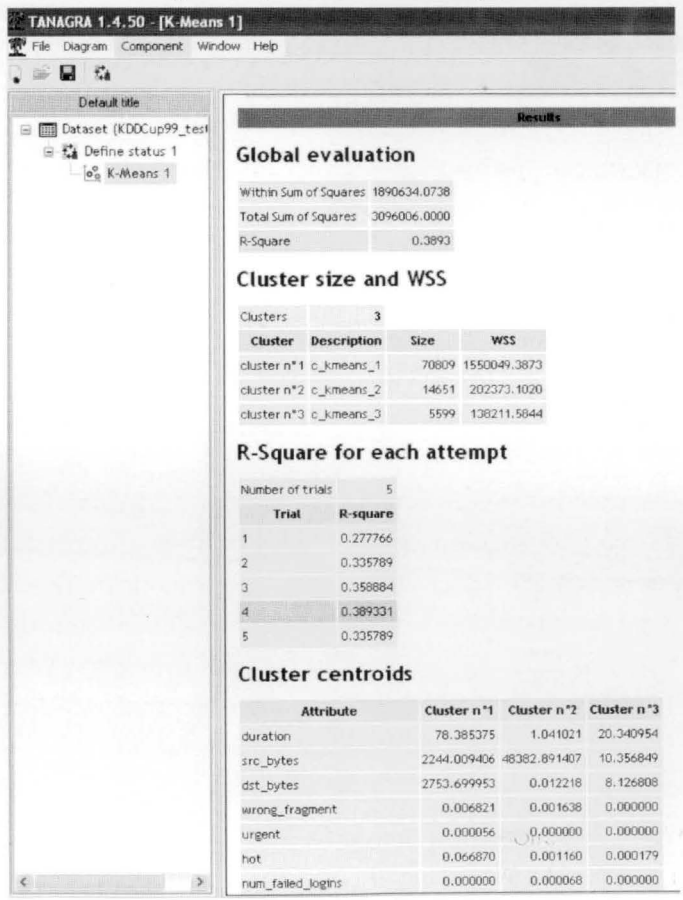

Figure 6. K-means clustering Result with Tanagara

Tanagara

- Clustering Results on KDDdataset using Tanagara tool

Conclusion

In this paper, we statistically analyzed the entire KDD data set. The analysis showed that there are two important issues in the data set which highly affects the performance of evaluated systems, and results in a very poor evaluation of anomaly detection approaches. To solve these issues, a new data set is proposed, NSL-KDD, which consists of selected records of the complete KDD data set. This data set is publicly available for researchers through website and has many advantages over the original KDD data set. We also demonstrated some of the data mining tools which are freely available and which can be effctively used to perform data mining tasks on the KDD dataset. Because of the lack of public data sets for network-based IDSs, we believe the work presented in the paper can be of help to researchers to compare different intrusion detection methods.

References

[1]. **Mahbod Tavallaee, Ebrahim Bagheri, Wei Lu, and Ali A. Ghorbani, (2009).** "A Detailed Analysis of the KDD CUP 99 Data Set", *IEEE Symposium on Computational Intelligence and Security and Defense Applications (CISDA)*

[2]. **Mohammad Khubeb Siddiqui and Shams Naahid (2013),** Analysis of KDD CUP 99 Dataset using Clustering based Data Mining *International Journal of Database. Theory and Application* Vol.6, No.5 pp.23-34.

[3]. **J. McHugh, (2000).** "Testing intrusion detection systems: a critique of the 1998 and 1999 darpa intrusion detection system evaluations as performed by lincoln laboratory," *ACM Transactions on Information and System Security*, Vol. 3, No. 4, pp. 262–294.

[4]. http://www.kdnuggets.com/datasets/kddcup.html

[5]. www.cs.waikato.ac.nz/ml/weka/.

[6]. http://eric.univ-lyon2.fr/~ricco/tanagra/en/tanagra.html.

[7] **M. Mahoney and P. Chan, (2003).** "An Analysis of the 1999 DARPA/LincolnLaboratory Evaluation Data for Network Anomaly Detection," *Lecture Notes in Computer Science*, pp. 220–238.

[8]. **L. Portnoy, E. Eskin, and S. Stolfo, (2001).** "Intrusion detection with unlabeled data using clustering," *Proceedings of ACM CSS Workshop on Data Mining Applied to Security*, Philadelphia, PA, November.

ABOUT THE AUTHORS

Kamini C. Nalavade is currently pursuing her PhD in the Department of Computer Engineering, VJTI, Mumbai. She received her B.E. Degree in Computer Science and Engineering from the SGGS, College of Engineering and Technology, Nanded in 2001 and M.Tech Degree in Computer Engineering from Veermata Jijabai Technological Institute (VJTI), Mumbai in 2007. She has published more than 20 papers in international journals and Conferences. Her research interest includes Intrusion Detection, Network Security, Data Mining and Data Privacy.

Dr. B. B. Meshram is currently working as a Professor and Head in Computer Technology Department in Veermata Jijabai Technological Institute (VJTI), Matunga, Mumbai. He has published more than 200 papers in National & International Conferences & Refereed Journals. He has submitted more than five patents in his research interest area. His areas of interest include Object Oriented Database Management Systems, Computer Network Security and Multimedia Systems.