Impact of Data Set Distinction and Normalization in C5.0 Decision Tree

M. E. Elhamahmy[1], H. N. Elmahdy[2] and Imane Saroit[3]

[1] Cairo University, Department of Information Technology,
Faculty of Computers and Information,
Cairo University, Egypt
mezzat1967@yahoo.com

[2] Cairo University, Department of Information Technology,
Faculty of Computers and Information,
Cairo University, Egypt
info@h-elmahdy.net

[3] Cairo University, Department of Information Technology,
Faculty of Computers and Information,
Cairo University, Egypt
iasi63@hotmail.com

Abstract: Machine learning (ML) techniques are used to implement the anomaly based intrusion detection systems (IDSs). The network based IDS (NIDS) may use (ML) in discriminating the normal from the malicious traffics. The KDD CUP 1999 data set is widely used for training, testing and evaluating the mining models of (NIDS). The complete set of KDD'99 has a big deal of records. So that, most of IDS classification experiments were done using the 10 % of the KDD CUP 1999 data set. Others made random selections of records to derive a challenge data set for training and testing their proposed (NIDS). Having conducted a statistical analysis on the complete set of KDD'99, it is found some shortcoming issues. So, this study proposes a three batch approach to overcome these issues and address its impact on the (NIDS) that uses C5.0 decision tree algorithm. Two batch experiments are maintained to preprocess, clean, and normalize the KDD'99 data set. The last batch experiment The C5.0 decision tree algorithm is trained and tested by using the preprocessed data set. The proposed approach succeeded to address the impact of the mentioned shortcomings found on the used data set on the over all performance of the IDS. The average cost of the proposed model is better than that of the winner entry of the KDD'99 competition.

Key Words: Intrusion Detection Systems, Decision Tree, C5.0 Algorithm, KDD CUP 1999 Data Set, Data Normalization.

1- Introduction

The objective of IDS is to detect all intrusions in an efficient manner [1]. IDS may be classified as misuse and anomaly based. Misuse detector is considered a signature-based detector as well. It has a signature database updated frequently to have the most recent attack signature (pattern that identify the attack). Misuse detector can detect the known attacks with lower wrong alerts. However, it generates many wrong alarms with the novel attacks. Rather than the evasion techniques that may be used to turn around the attack-signature found on the signature-database. On the other hand, the anomaly based detection can potentially detect novel attacks. Progress in applying intelligent solutions and machine learning algorithms to real life problems, evaluating the signature-based detectors are easier than evaluating anomaly detectors [2]. Signature-based detectors can be tested against known attacks; however it is implausible to predict future attacks. Evaluation is the key in making significant such as intrusion detection [3]. One of the most important data sets for testing intrusion detection systems (IDS) is the DARPA/Lincoln Laboratory off-line evaluation data set, or IDEVAL [4]. The network traffic from the 1998 data set was also used to develop the 1999 KDD cup machine learning competition, which had 24 participants. It continues to be used to test intrusion detection methods [6]. The winning entry was submitted by Dr. Bernhard Pfahringer of Austrian Research Institute for Artificial Intelligence [7]. The winning entry used the C5.0 decision tree algorithm, trained by the 10 % of the KDD CUP 1999 data set. The data set used has (494021) connection records, each have 41-features value. Then, evaluated by the testing data set of KDD CUP 1999, which has (311,029) connection records with the same 41-features [7]. From this date on, many researches have been used the same data sets. This data set is usually used as a reference of the improvement of the proposed IDSs. Many of the previous works used to compare their IDSs evaluation results with the winning entry or with each other [8] - [13]. Therefore, the KDD CUP 1999 considered to be a benchmark data set for IDS evaluation. The KDD CUP 1999 data set was derived from IDEVAL background network traffic which was synthesized, for evaluating anomaly detection systems [1]. This background traffic is used to build a normal profile of the network traffic, so that deviations can be reported as suspicious. By statistically analyzing the KDD'99 data set, it is found some shortcomings. One of these shortcomings is the big deal of redundant records. Second shortcoming is found in two features of the (41) features. The data set is characterized by (41) features in addition to "Label" field. It is found one feature has a single value "0" over all both of the training and testing data set. The "num_outbound_cmd" is used for the class of attacks that exploits some "FTP-ports" vulnerability.

In this study, data cleaning and data normalization methods are proposed to solve the mentioned issues, resulting in a new train and test data sets which consist of selected records of the completed KDD'99 data set. The number of records on the train and test data set is reasonable, and free of mentioned shortcomings. So that, it is possible to run the experiment on the complete set after preprocessing without need to randomly select subset records. Consequently, the distribution of different attacks and normal records will vary on the distinct data sets than the original ones. The rest of the work is organized as follows. Section 1 introduces the KDDCUP99 data set which is wildly used in anomaly based IDS. In Section 2, the first review of the issues found in DARPA'98 is discussed and then discuss the possible existence of those problems in KDD'99. Section 3, discusses the shortcoming issues of the KDDCUP'99. The statistical observations of the KDD data set will be explained in Section 4. Section 5, provides some solutions for the existing problems in the KDD data set. Finally, Section 6 provides the conclusions of the work.

2- Related Work

In [5], McHugh notes that "some methodologies used in the evaluation are questionable and may have biased its results". However he concluded that "The detailed analysis of missed detections (new and old) and false alarms is largely independent of the evaluation methodology and thus is unaffected by the criticisms". Mahoney proposed more detailed analysis of the data and confirmed McHugh's speculation that this data have statistically different characteristics from real traffic [2]. In [14], the authors have conducted a statistical analysis of KDDCUP'99 data set resulted in two issues, one of them the big number of redundant records and the other one is the analyze the difficulty level of the records in KDD data set. This study shows that there are more important shortcomings of the same data set. One of the most important issues is bad-labeling of the connection records. As there are 9 records labeled as various attacks, and there are another identical 9 records labeled as normal. This may pop up the question asked by McHouge [5], "when considered attack?" Another issue is the one-value fields; the "num_outbound_cmds" feature has a single value of (0) for all the records in both of the training and testing data sets. The DARPA data set is a synthesized data set and have some shortcomings discussed by McHugh [5]. However, [15] recently argued with the usefulness of the DARPA data set. So that we also believe it still can be applied as an effective publicly available benchmark data set.

3. Data Set

This section introduces to DARBA Data Sets in (section 3.1), then Section 3.2 introduces to the KDD CUP 1999 that is used in this study.

3.1 DARPA Data Sets

In 1998, the Defense Advanced Research Projects Agency (DARPA) intrusion detection evaluation created the first standard corpus for evaluating intrusion detection systems. The 1998 off-line intrusion detection evaluation was the first in a planned series of annual evaluations conducted by the Massachusetts Institute of Technology (MIT) Lincoln Laboratories under DARPA sponsorship. The corpus was designed to evaluate both false alarm rates and detection rates of intrusion detection systems using many types of both known and new attacks embedded in a large amount of normal background traffic [1]. Over 300 attacks were included in the 9 weeks of data collected for the evaluation. These 300 attacks were drawn from 32 different attack types and 7 different attack scenarios. The corpus was collected from a simulation network that was used to automatically generate realistic traffic including the attacks cited above. The 1999 KDD intrusion detection contest uses a version of this dataset [6].

3.2 KDD CUP 1999

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. 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. Attacks fall into four main categories:

- **Denial of Service (DOS)**: A DoS attack is a type of attack in which the hacker makes a memory resources too busy to serve legitimate networking requests, and hence denying users access to a machine e.g. "appache", "smurf", "neptune" and "ping of death" (pod).

- **Remote to Local attacks (R2L)**: an attack in which user sends packets to a machine over the internet, and the user does not have access to expose the machine vulnerabilities and exploit privileges which a local user would have on the computer, e.g. "xlock", "xnsnoop", "ftp_write", and "warezclient".

- **User to Root attacks (U2R)**: The hacker has a normal user account and attempts to gain super user privileges, e.g. "rootkit", "buffer_overflow" and "perl".

- **Probing**: An attack in which the hacker scans the network machines or devices in order to determine holes or vulnerabilities that may be later exploited so as to compromise the system e.g. "satan", "portsweep", "ipsweep" and "nmap".

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. This makes the task more realistic.

4. The Shortcomings of the KDD CUP 1999 Data set

The first shortcoming is the redundant records. In [14], Authors proposed a solution to reduce the redundant records issue. They removed all the repeated records in the entire KDD train and test, and kept only one copy of each record. Figure (1) shows a sample of redundant attack records of the same type.

```
0,icmp,eco_i,SF,18,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,0.000,
0.000,0.000,0.000,1.000,0.000,0.000,1,12,1.000,0.000,1.000,
1.000,0.000,0.000,0.000,0.000,ipsweep.

0,icmp,eco_i,SF,18,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,0.000,
0.000,0.000,0.000,1.000,0.000,0.000,1,12,1.000,0.000,1.000,
1.000,0.000,0.000,0.000,0.000,ipsweep.

0,icmp,eco_i,SF,18,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,0.000,
0.000,0.000,0.000,1.000,0.000,0.000,1,17,1.000,0.000,1.000,
1.000,0.000,0.000,0.000,0.000,ipsweep.

0,icmp,eco_i,SF,18,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,0.000,
0.000,0.000,0.000,1.000,0.000,0.000,1,17,1.000,0.000,1.000,
1.000,0.000,0.000,0.000,0.000,ipsweep.

0,icmp,eco_i,SF,18,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,0.000,
0.000,0.000,0.000,1.000,0.000,0.000,1,17,1.000,0.000,1.000,
1.000,0.000,0.000,0.000,0.000,ipsweep.
```

Figure (1): Redundant records labeled as the same attack type and have the same 41 feature values.

However, they did not study if there are some redundancy between attack records and the normal records. We have found records labeled as one attack which have identical 41-features values with other normal records in KDD'99 data sets. One other issue on the KDD'99 testing data set, it contains (311,029) records, two records of them have the "icmp" value of the "service" feature, so that these records removed as in [14]. The remainder (311,027) records have been reduced as in [14] to become (77,289) distinct records. However, we have found 70 records labeled as different attacks and other identical records have the same feature values and labeled as normal. Figure (1) shows a sample of this issue. We have found also 5 records labeled with different attack types and other identical 5 records labeled with different attack types. So that, we have removed 75 records from the distinct test set to become (77,214) records. A sample of the removed records out of the training and testing sets are shown in Figure (1) and (2) respectively. This may pop up the question asked by McHouge [5], "when considered attack?". Another issue, the feature called "num_outbound_cmds" that indicates the number of outbound commands in an ftp session, which has a (0) value for all the records on both of the training and testing data set. This feature has no effect on any mining model as it has no discrimination of attacks from normal data. Mahoney and Chan [2] analyzed DARPA background network traffic and found evidence of simulation artifacts that could result in an overestimation of the performance of some anomaly detection techniques.

```
1,0,0,1,0,0,1,1,0,1,0,0,0,0,SF,0,0,0,0,0,0,0,0,0,0,0,0,icmp,0
,0,1,0,ecr_i,20,1,0,0,0,0,0,0,normal.

1,0,0,1,0,0,1,1,0,1,0,0,0,0,SF,0,0,0,0,0,0,0,0,0,0,0,0,icmp,0
,0,1,0,ecr_i,20,1,0,0,0,0,0,0,satan.
______________________________________________________________
1,0,0,1,0,0,1,1,0,1,0,0,0,0,SF,0,0,0,0,0,0,0,0,0,0,0,0,icmp,0
,0,1,0,tim_i,564,1,0,0,0,0,0,0,normal.

1,0,0,1,0,0,1,1,0,1,0,0,0,0,SF,0,0,0,0,0,0,0,0,0,0,0,0,icmp,0
,0,1,0,tim_i,564,1,0,0,0,0,0,0,pod.
```

Figure (2): Records have the same 41 feature values and labeled as "normal" as well as some attack on the KDDCUP'99 training data set.

```
0,udp,private,SF,105,146,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1
,0.000,0.000,0.000,0.000,1.000,0.000,0.000,111,110,0.990,
0.020,0.010,0.00,0,0.000,0.000,0.000,0.000,normal.

0,udp,private,SF,105,146,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1
,0.000,0.000,0.000,0.000,1.000,0.000,0.000,111,110,0.990,
0.020,0.010,0.00,0,0.000,0.000,0.000,0.000,snmpgetattack.
```

Figure (3): Records have the same 41 feature values and labeled as "normal" as well as some attack on the KDD'99 test data set.

However, [14] mentioned that the aforementioned simulation artifacts do not affect the KDD data set since the 41 features used in KDD are not related to any of the weakness mentioned in [2].

5. Experiments

We used an Intel Core2Duo processor of 800 GHz with 1 GB Ram, and so used SPSS Clementine version 12.0 as data mining software tool [16] in this experiment. The first batch is the reduction procedure to remove the redundant records. Figure (4) presents the reduction procedure to reduce the redundant records. We conducted the experiments in three batches. The first batch used to remove the redundant and malformed records out of the KDD'99 datasets. The second batch is the data set normalization and preprocessing. The third experiment is to use a decision tree classifier trained on the distinct 10% of KDD Cup'99 training dataset, and evaluate the results using the distinct KDD Cup'99 test dataset.

5.1 Dataset Reduction

Table (1) presents the breakdowns and reduction rates of the original KDD CUP'99 data set and its distinct dataset for training in (Table 1-a) and test in (Table 1-b) respectively. Regarding to the results of the KDD'99 classifier learning contest [8], the actual distributions of attack types in the training and test 10% datasets are as shown in Table (1-a) and Table (1-b) respectively, under the column labeled Original' Most of the studies comes later on and compare its results with the winning entry used.

5.2 Preprocessing

Normalization: Preprocessing is often required before using any data mining algorithms to improve the results' performance [18].

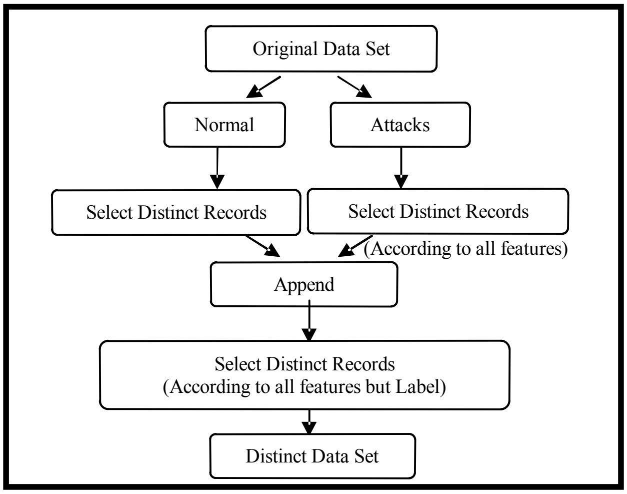

Figure (4): The reduction procedure to remove redundant records from the KDDCUP'99 datasets. the normal labeled records or another attacks are necessary and achieve the same objectives of removing the redundant records.

Table (1-a): The breakdowns and reduction rates of the original 10% of the KDD CUP'99 training data set and its distinct dataset.

Label	Original		Distinct		Reduction Rate
	Count	%	Count	%	%
DoS	391458	79.24	54571	37.48	86.06
Normal	97278	19.69	87832	60.33	9.71
Probe	4107	0.83	2130	1.46	48.13
R2L	1126	0.23	999	0.69	11.27
U2R	52	0.01	52	0.04	0
Total	494021	100	145584	100	70.53

Table (1-b): The breakdowns and reduction rates of the original test of the KDD CUP'99 data set and its distinct dataset

Label	Original		Distinct		Reduction Rate
	Count	%	Count	%	%
DoS	229853	73.91	23568	30.52	89.75
Normal	60593	19.48	47880	62.01	20.98
Probe	4166	1.34	2677	3.47	35.74
R2L	16189	5.20	3019	3.91	81.35
U2R	228	0.07	70	0.09	69.30
Total	311029	100	77214	100	75.17

Data normalization is one of the preprocessing procedures in data mining. It aims to scale the data so as to fall within a small specified range such as -1.0 to 1.0 or 0.0 to 1.0. Normalization before applying the mining algorithm is effective in reducing the learning time as well as the importance of the features. In real applications, because of the differences in range of attributes' value, one attribute might overpower the other one. Normalization prevents outweighing attributes with large range like 'salary' over attributes with smaller range like age [18]. The goal is to equalize the size or magnitude and the variability of these attributes. There are many methods for data normalization, which include Min-max normalization, Z-score normalization and normalization by decimal scaling. Min-max normalization performs a linear transformation on the original data. Suppose that $\min_a$ and $\max_a$ are the minimum and the maximum values for attribute A. Min-max normalization maps a value v of A-v in the range (0, 1) by computing:

$$v' = \frac{v - \min_a}{\max_a - \min_a} \qquad (1)$$

In Z-score normalization, the values for an attribute A are normalized based on the mean and standard deviation of A. A value v of A is normalized to v by computing:

$$v' = \frac{v - \text{average}}{\text{std. dev}} \qquad (2)$$

Where, average and std. dev. are the mean and the standard deviation, respectively of certain attribute. This method of normalization is useful when the actual minimum and maximum of the attributes are unknown, or when there are outliers that dominate the min-max normalization.

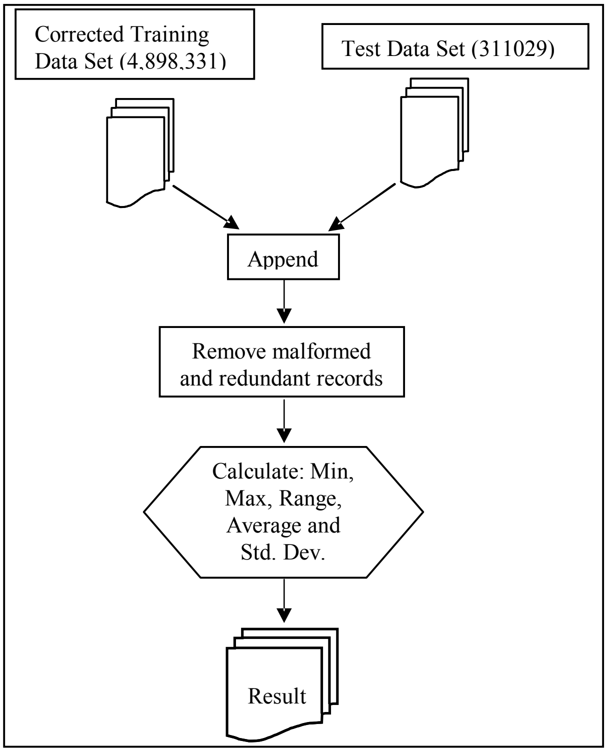
Corrected Training Data Set (4,898,331)
Test Data Set (311029)
Append
Remove malformed and redundant records
Calculate: Min, Max, Range, Average and Std. Dev.
Result

Figure (5): Statistically analysis of the KDD'99 data set.

We choose the max-min normalization to be applied on the selected columns (none-normalized features) of the KDD'99 data set. It is the suiTable normalization procedure as it specifies a range of (0.00 to 1.00). As shown in Figure (5), we need to calculate the min, max, range, average and the standard deviation for the numeric features (38 out of 41 features) of the KDD'99 data set. As we should generalize the both of training and testing sets, we append the both sets at first. Then remove the malformed records such as the two records contain "icmp" value in "service" column [14]. And so, we remove the redundant records that have the identical 41-feature values but leave just one record. The results of the statistical analysis show that the "num_outbound_cmds" feature has min, max, average, and std. dev. of (0). So, we removed this feature. The features that needed to be normalized are: duration, src_bytes, dst_bytes, hot, count, srv_count, num_compromised, num_root, dst_host_count, dst_host_srv_count, and num_file_creations.

5.3 C5.0 Decision Tree Algorithm

C4.5 is an algorithm used to generate a decision tree developed by Ross Quinlan [17]. C4.5 is an extension of Quinlan's earlier ID3 algorithm. Decision trees generated by C4.5 can be used for classification, and for this reason, C4.5 is often referred to as a statistical classifier. C4.5 builds decision trees from a set of training data in the same way as ID3, using the concept of information entropy. The training data is a set

$$\mathbf{S = s1, s2...}$$

of already classified samples. Each sample

$$\mathbf{s_i = x1, x2...}$$

is a vector where x1,x2,... represent attributes or features of the sample. The training data is augmented with a vector

$$\mathbf{C = c1, c2...}$$

Where c1, c2... represent the class to which each sample belongs. Quinlan went on to create C5.0 and See5 (C5.0 for Unix/Linux, See5 for Windows) which he markets commercially. C5.0 offers a number of improvements on C4.5. [17] Some of these are:

- Speed - C5.0 is significantly faster than C4.5 (several orders of magnitude)
- Memory usage - C5.0 is more memory efficient than C4.5
- Smaller decision trees - C5.0 gets similar results to C4.5 with considerably smaller decision trees.
- Support for boosting - Boosting improves the trees and gives them more accuracy.
- Weighting - C5.0 allows you to weight different attributes and misclassification types.
- Winnowing - C5.0 automatically winnows the data to help reduce noise.
- C5.0/See5 is a commercial and closed-source product, although free source code is available for interpreting and using the decision trees and rule sets it outputs.

We used the decision tree C5.0 classifier included in SPSS Clementine version 12.0 [16]. The C5.0 decision tree is used with the distinct data set that preprocessed in section 4.2. Then, we test the result model of trained IDS with the distinct test data set. The distinct test set should be preprocessed in the same manner as the trained set. The results are compared to the winner entry of the KDD'99 competition results [7]. The average cost is calculated as well as the standard deviation and the standard error. The results show the effect of the redundant records on the detection accuracy and on the produced rule set a well.

6. Results and Discussion

Table 2. Confusion matrix of testing the proposed model with the distinct KDD'99 data set.

	Probe	**Normal**	**DoS**	**R2L**	**U2R**	**Total**
Probe	1,682	626	229	133	7	2677
Normal	722	47,064	72	12	5	47875
DoS	63	1,486	21,987	32	0	23568
R2L	3	1,895	0	699	282	2879
U2R	3	162	0	12	38	215
Total	2473	51233	22288	888	332	77214

Table 3. Comparison of the proposed model results to the winner entry of the KDD'99.

	Avg. Cost	Std. Dev	Std. Error
Winner Entry	0.2331	0.8834	0.003
Proposed IDS Model	0.1817	0.9592	0.003

Table (2), shows the confusion matrix of testing the proposed model with the distinct KDD'99 data set (77,214) records. Table (3), shows the comparison of the average cost, standard deviation and standard error of the proposed model results comparable to the winner entry. According to the same cost matrix used to evaluate the competitor's entries [8], the proposed model has a lower average cost, with the same standard error. It implies the effect of data cleaning, removing the redundant records, and data normalization on the mining tools used. The rare classes may be due to the data skewness as the percent of U2R and R2L classes are 0.27 % and 3.73% respectively.

7. Conclusion

In this study, a new approach is proposed to preprocess, clean, and normalize the KDD CUP'99 data sets. A proposed distinct procedure is applied to derive a clean, distinct and normalized data set. Then the decision tree C5.0 classifier algorithm is trained and tested using the derived distinct data sets. Although the derived data sets are a subset of the original one, the effectiveness of the decision tree C5.0 algorithm is doing better with a minimal average cost. It implies the impact of the redundant, malformed and none-normalized records on the IDS performance. To do that, the merge both of the training and testing data sets of the KDD'99 in one set is done first. Then, conducted a statistical analysis of each feature of the (41) features resulted in two important issues. One of them is the "num_outbound_cmds" feature which has a value of (0) for all the records on both of the training and testing data set. The second issue is the "is_host_login" feature which has only two records with value (1) and the rest of (4,898,429) records have (0) value for this feature. On the testing dataset there are only (12) records have this feature of value (1) and the rest records have (0) value. So, both of them are removed. The features became 39-features only. The redundant records represent a percent over 70% for the both datasets, the training and the test dataset. So that, the results of the IDS that trained and tested by conducting the original KDD'99 dataset with its redundant records may be not accurate enough. It may be biased toward the most redundant records. The dominate min-max normalization approach is used for 11 features to resize the range space for each feature of them and specifies it between 0.00 and 1.00. The preprocessing of the used data set is very effective on the performance of the mining algorithms used. Rather than it resulted in reduced data sets for training and testing purposes. The reduction rates of the KDD'99 data sets are 70.53 and 57.17% for the training and testing data set respectively. To address the impact of the data set cleaning and normalization the (NIDS) model that uses the C5.0 decision tree algorithm is trained with the preprocessed 10_percent of the KDD'99 data set. Then it is tested with the preprocessed test set of the KDD'99. The results are evaluated using the same evaluation measures used in KDD CUP'99 competition. The average cost of the proposed model is 0.1817. It is better than the average cost of the winner entry. However the training time and size of the training data set is reduced due to data sets reduction. It proves the impact of data cleaning and data normalization before applying the IDS machine learning algorithms.

8. References

[1] **Lippmann, R. et al.,** "The 1999 DARPA Off-Line Intrusion Detection Evaluation". *Computer Networks.* 34(4) 579-595, (2000). Data is available on http://www.ll.mit.edu/IST/ideval/.

[2] **Mahoney M., and Chan P. K.** "An Analysis of the 1999 DARPA/Lincoln Laboratory Evaluation Data for Network Anomaly Detection". *6th Intl. Symp. Recent Advances in Intrusion Detection.* pp. 220-237, (2003).

[3] **Witten, I. H. and Frank, E.** "Data Mining, Practical Machine Learning Tools and Techniques with Java implementations". *Morgan Kaufmann.* USA (2000).

[4] **Lippmann, R.P. and Haines J.** "Analysis and Results of the 1999 DARPA Off-Line Intrusion Detection Evaluation". *In Recent Advances in Intrusion Detection, Third International Workshop*, Proc. RAID 162-182**.** (2000).

[5] **McHugh John** "Testing Intrusion Detection Systems: A Critique of the 1998 and 1999 DARPA intrusion detection system evaluations as performed by Lincoln Laboratory". *ACM TISSEC.* 3(4) pp. 262-291, Nov. (2000).

[6] **KDD CUP 1999.** It is available on: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html, October (2007).

[7] **Charles Elkan.** " Results of the KDD' 99 Classifier Learning". University of California. San Diego, Available at: http://cseweb.ucsd.edu/~elkan/clresults.html. (2000).

[8] **Kemmerer, R. A. and Vigna, G.** "Intrusion detection: A brief history and overview", *IEEE Security and Privacy Magazine (supplement to Computer.* vol. 35, no. 4, pp. 27-30, April (2002).

[9] **Maheshkumar Sabhnani, Gürsel Serpen** "KDD Feature Set Complaint Heuristic Rules for R2L Attack Detection" *Security and Management*. 310-316, (2003).

[10] **Wa'el M. Mahmud, Hamdy N. Agiza, Elsayed Radwan.** "Intrusion Detection Using Rough Sets based Parallel Genetic Algorithm Hybrid Model". *World Congress on Engineering and Computer Science (WCECS 2009),* October (2009).

[11] **Mukkamala S., Sung A. H.** "Feature Selection for Intrusion Detection Using Neural Networks and Support Vector Machines". *Journal of the Transportation Research Board of the National Academies*. (2003).

[12] **Mukkamala S., Sung A., Abraham A.** "Intrusion detection using ensemble of soft computing and hard computing paradigms". *J. Network Comput. Appl.* 28 (2) 167–182, (2005).

[13] **Mukkamala, S., Sung, A.H., and Abraham, A.** "Designing Intrusion Detection Systems: Architectures, Challenges and Perspectives". (2003).

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

[15] **C. Thomas, Vishwas Sharma, N. Balakrishnan.** "Usefulness of DARPA data set for intrusion detection evaluation". *Proceedings of SPIE,* vol. 6973, pp. 69730G-69730G-8, March (2008).

[16] **SPSS Clementine is available on:** http://www.spss.com/software/modeling/modeler/

[17] **Quinlan J. R.** "Improved use of continuous attributes in c4.5". *Journal of Artificial Intelligence Research*, 4:77-90, (1996).

[18] **Luai A. Shalabi, Zyad Shaaban and Basel Kassabeh.** "Data Mining A Preprocessing Engine". *J. Computer Sci.,* 2(9): 735-739. (2009)