Jurnal Sains dan Teknologi 13 (1), Maret 2014: 1-9
ISSN 1412-6257

A DEPTH PRE-PROCESSING DATA ANALYSIS FOR INTRUSION DETECTION SYSTEM USING OUTLIER DETECTION AND BOX-COX TRANSFORMATION TECHNIQUE

Dahliyusmanto[1], Abdul Hanan Abdullah[2], Syefrida Yulina[3]

[1]Department of Electrical Engineering, Faculty of Engineering, Riau University,
Kampus Bina Widya Km 12, 5 Sp. Panam, Pekanbaru, 28293, Indonesia

dahliyusmanto@lecturer.unri.ac.id

[2]Department of Computer System, Faculty of Computing, Universiti Teknologi Malaysia,
Skudai, Johor Bahru, 81310, Malaysia

hanan@utm.my

ABSTRACT

An Intrusion Detection System (IDS) seeks to identify unauthorized access to computer systems' resources and data using a statistical approach. The scale on which a dataset variable is measured may not the most appropriate for statistical analysis or describing variation, and may even hide the basic characteristics of the data. This paper proposed a pre-processing analysis for detecting unusual observations that do not seem to belong to the pattern of variability produced by the other observations. The pre-processing analysis consists of outliers detection and Transformation. Outliers are best detected visually whenever this is possible. Usually, the original data sets are not normally distributed. If normality is not a viable assumption, one alternative is to make non-normal data look normal. This paper explains the steps for detecting outliers' data and describes the Box-Cox power transformation method that transforms them to normality. The transformation obtained by maximizing lamda functions usually improves the approximation to normality

Keywords : IDS, dataset, outliers, transformation, pre-processing

INTRODUCTION

The methodology of intrusion detection can be divided into two-category: anomaly intrusion detection and misuse intrusion detection. Anomaly intrusion detection refers to detecting intrusion based on the anomalous behaviour of the attackers. Therefore, the distinction by categorizing the good or acceptable behaviour is very important. In the anomaly detection method, a statistical approach and neural net approach are usually taken to detect intrusion attempts. There are many ways in which dataset could be used to characterize normal behaviour of programs, each of which involves building or training a model using traces of normal processes. The enumerating sequences method (Forrest et al, 1996; Hofmer et al, 1998) depend only on enumerating sequences that occur empirically in traces of normal behavior and subsequently monitoring for unknown patterns. Two different methods of enumeration were tried, each of which defines a different model, or generalization, of the data. There was no statistical analysis of these patterns in the earlier work.

Frequency-based methods model the frequency distributions of various events. For the system-call application, the events are occurrences of each pattern of system calls in a sequence. One example of a frequency-based method is the *n-gram vector* used to classify text documents (Damashek, 1995).

Data mining approaches are designed to determine what features are most important out of a large collection of data. In the current problem, the idea is to discover a more compact definition of normal than that obtained by simply listing all patterns occurring in normal. Also, by identifying just the

main features of such patterns, the method should be able to generalize to include normal patterns that were missed in the training data. Lee and others used this approach to study a sample of system call data (Lee et al, 1997; Lee et al, 1998). They used a program called “RIPPER” to characterize sequences occurring in normal data by a smaller set of rules that capture the common elements in those sequences. During monitoring, sequences violating those rules are treated as anomalies. Because the results published in (Lee, 1997) on synthetic data were promising, we chose this method for further testing.

The statistical approach, data sets gained from detection results are used. Further, the data set should be calculated and analysed. When analysing data, we will sometimes find that one value which is far from the others. Such a value is called an "outlier". Given a mean and standard deviation, a statistical distribution expects data points to fall within a specific range. Many researchers have used statistical data analysis. Outliers typically are attributable to one of the following causes; (1) the measurement is observed, recorded, or entered into the computer incorrectly, (2) the measurements come from a different population, and (3) the measurement is correct, but represents a rare event. Sometimes, when we encounter an outlier, we may be tempted to delete it from the analyses. One possibility is that the outlier happened by chance. In this case, we should keep the values in our analyses. The value came from the same population as the other values, so should be included.

The paper is organized as follows: the dataset extraction; the steps for detecting outlier data and Box-Cox power transformation, standardizes values, generalized square distances and transformation to near normality.

MATERIALS AND METHODOLOGY

Firstly, this study make a clear distinction about intrusion, intrusion detection, and intrusion detection system. According to Bace & Mell (2001), the intrusion as an attempt to compromise *Confidentiality*, *Integrity* and *Availability* (Mukkamala et al., 2002), or to bypass the security mechanisms of a computer or network. Intrusion detection is the process of monitoring the event occurring in a computer system or network, and analysing them for signs of intrusions. The intrusion detects system is the software of hardware system to automate the intrusion detection process (Bace & Mell, 2001; Stavroulakis & Stamp, 2010).
There are many types and variations of computer network intrusion. In the DARPA 98 intrusion detection evaluation data, which is widely used to evaluate the intrusion detection system, intrusion can be grouped into four main categories, namely Denial of Service, User to Root, Remote to User, and Probes (Kendall, 1999).

Generally, a common drawback of IDS technologies is that they cannot supply absolutely accurate detection. False Positive (FP) and False Negative (FN) are two indicators to assess the degree of accuracy. The former occurs when IDS incorrectly identifies benign activity as being malicious, whereas the latter comes about if IDS fails to identify malicious activity (Stavroulakis & Stamp, 2010). The collection of FP and FN cases from real world traffic, statistically analyse these cases, and propose three findings. First, the great majority of false cases is FNs, because most application behaviour and its content format are self-defined, not conform to the RFC specification. The summarized and refined many of the previous surveys (Debar et al., 2000; Axelsson, 2000; Lazarevic et al., 2005) to give a new perspective of taxonomy for IDSs

Dataset Extraction

The experiment's data for Grid computing intrusion detection model were dataset obtained from Computer Immune Systems (CIS) Lab and Grid Lab test. These datasets were appropriate for host-based intrusion detection aspects in the Grid. CIS has collected several datasets of system calls executed through active process, which include different kinds of programs (e.g. A program that runs as daemons and those that do not), programs that vary widely in their size and complexity, different kinds of intrusions (buffer overflow, symbolic link attacks, and Trojan programs). Some of the normal data are "*Synthetic*" and some are "*Live*". *Synthetic* traces are collected in production environments by running a prepared script; the program options are chosen solely for the purpose of exercising the program, and not to meet any real user requests. *Live* normal data are traces of programs collected during normal usage of a production computer system, while, the data from Grid Lab test are collected from the programs running on Grid services (e.g. *globus-gsiftp*, *globus-url-copy*, *globus-ws-submit*, etc). Nonetheless, both of the datasets will be integrated. Afterward, the datasets have to be

extracted to obtain the characteristics of systems calls. The extraction procedures consist of the number of system calls, the number of processes, and the characteristic system calls itself.

Traces of each program's data sets are recorded in **.int* and gzipped files, because of that extraction process are needed. This research must consider how to extract a file from data sets easily. This research considers the extraction method using an internal viewer program on windows operating system to obtain the number of system call, number of processes, and to identify the characteristic of system call as shown in Figure 1.

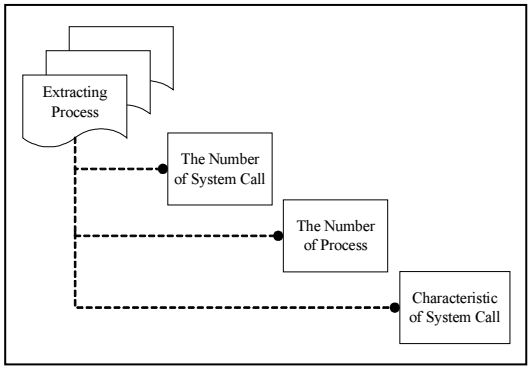

Figure 1. Extracting process data sets of system calls

'The number of system call

Certain of number system call are extracted in a certain number of processes that are in a parent-child relation in the system log. The algorithm used to count system call is shown in Figure 2.

```
Procedure Counting the Number of System Call;
var l, i, a, n, c, j;
l = 170; %maximum number of system call in each activity
      read ( data );
for i = 1 to l
            a ( i ) = find ( data ( : , 1 ) = 1 );
            [ n , c ] = size a ( i );
             j ( i ) = n;
end;
```

Figure 2. Algorithm to count the number of system calls

The number of processes

All system calls are extracted within a number of processes that are in a parent-and-child relationship. The numbers of process can be counted manually because they are a little resulted in extracting process.

Characteristic of system call

System calls are extracted in one or more processes that are in a parent-and-child relationship. The extraction range is based on certain characteristic system calls (e.g., fork, exit, open, etc.) to an activity.

OUTLIERS DETECTION

An outlier is a data point that is located far from the rest of the data. Given a mean and standard deviation, a statistical distribution expects data points to fall within a specific range. Many researchers have used this approach in statistical data analysis. Generally, the contemporary algorithms are hybrid algorithms based on mathematical considerations and random restarts. They exploit the concept of discriminant analysis, which for an outlier is expected to appear big and significant.

There are four steps for detecting outliers:
To make a dot plot for each variable, and the algorithm shown in Figure 3.

```
Procedure Making a Dot Plot;
        read ( data ), ( syscall );
        Boxplot ( data, 0, '+', 0 );
        Set ( gca, 'YTicklabel', syscall );
```

Figure 3. Algorithm to make a dot plot

1) To make a scatter plot for each pair of variables. It must order the distance *d* of each observation from smallest to largest as $d^2_{(1)} < d^2_{(2)} < ... < d^2_{(n)}$, whereas;

$$d^2 = (x - \bar{x}) \, . \, S^{-1} \, (x - \bar{x})' \qquad \text{(A.1)}$$

and the *quantile* of the chi-square distribution with p degrees of freedom is:

$$q_{c,p} \, ((j - 0.5)/n) = X^2_p \, ((n - j + 0.5)/n) \quad \text{(A.2)}$$

The algorithm to make a scatter plot is shown in Figure 4.

```
Procedure Making a Scatter Plot;
var i, d, xchi, xbar, cov, invcov;
read ( data )
    xbar    = mean ( data );
    cov     = cov ( data );
    invcov = inv ( cov );
    d = ( x – xbar ) * invcov * ( x – xbar )';
Sort ( d );
xchi = [ ];
For i = 1 to 49 % 49 are sum of activities
    xchi = [ xchi  chi2inv ( ( i – 0.5 ) / 49, 11 );
end;
scatter ( xchi, d )
```

Figure 4. Algorithm to make a scatter plot

2) To calculate the standardized values *z* on each column and examine these standardized values for large or small values. The algorithm of these as shown in Figure 5.

$$z_{jk} = x_{jk} - \bar{x}_k \; / \sqrt{S_{kk}} \qquad \text{(A.3)}$$

```
Procedure Calculating Standardized Value;
var  n, c, xbar, std, datcen, datstd;
 read ( data );
     xbar = mean ( data );
     std   =  std ( data );
[ n , c ] = size ( data );
    datcen = data – repmat ( xbar, n, 1);
    datstd  = datcen ./ repmat ( std, n, 1 );
```

Figure 5. Algorithm to calculate standardized data

3) To calculate the generalized square distances $(x - \bar{x}) . S^{-1} (x - \bar{x})$ and examine these distances for unusually large values. In a Scatter plot, these would be the points farthest from the origin.

TRANSFORMATION DATA

The transformations are nothing more than *re-expressions* of the data in different units. Appropriate transformations are suggested by (1) theoretical considerations or (2) the data themselves (or both). It has been shown theoretically that the data which are counted can often be made more normal by taking their square roots. Similarly, the *logit transformation* applied to proportions and Fisher's z-transformation applied to correlation coefficients yield quantities that are approximately normally distributed.

To select a power transformer, an investigator looks at the marginal dot diagram or histogram, and decides whether large values have to be "pulled" or "pushed out" to improve the symmetry about the mean. A Q-Q plot or other will check to see whether the tentative normal assumption is satisfactory, and should always examine the final choice.

A convenient analytical method is available for choosing a power transformation. Box and Cox (1964) had considered the slightly modified family of power transformations, as below:

$$x^{(\lambda)} = \begin{cases} x^{\lambda} - 1 / \lambda & \lambda \neq 0 \\ \ln x & \lambda = 0 \end{cases} \qquad \text{(A.4)}$$

which is continuous in λ for $x > 0$. Given the observations x_1, x_2, x_n, the Box-Cox solution for the choice of an appropriate power, λ is the solution that maximizes the expression. With multivariate observations, a power transformation must be selected for each of the variables. Let $\lambda_1, \lambda_2, \ldots, \lambda_p$ be the power transformations for the p measured characteristics. Each λ_k can be selected to maximize:

$$\ell_k (\lambda) = -n/2 \ln \left[1/n \sum_{j=1}^{n} (x_{jk}^{(\lambda_k)} - x_k^{(\bar{\lambda}_k)})^2 \right] + (\lambda_k - 1) \sum_{j=1}^{n} \ln x_{jk} \qquad \text{(A.5)}$$

where $x_{1k}, x_{1k}, \ldots, x_{nk}$ are the n observations on the kth variable, $k = 1,2, \ldots, p$. From this equation,

$$x_k^{(\bar{\lambda}_k)} = 1/n \sum_{j=1}^{n} x_{jk}^{(\lambda_k)} = 1/n \sum (x_{jk}^{\lambda_k} - 1) / \lambda_k \qquad \text{(A.6)}$$

is the arithmetic average of the transformed observations. The jth transformed multivariate observation is

$$x_j^{(\hat{\lambda})} = \begin{bmatrix} x_{j1}^{(\hat{\lambda}_1)} - 1 / \hat{\lambda}_1 \\ x_{j2}^{(\hat{\lambda}_2)} - 1 / \hat{\lambda}_2 \\ \vdots \\ x_{jp}^{(\hat{\lambda}_p)} - 1 / \hat{\lambda}_p \end{bmatrix} \quad \text{(A.7)}$$

where $\hat{\lambda}_1$, $\hat{\lambda}_2$, …, $\hat{\lambda}_p$ are the values that individually maximize Equation A.4. The procedure only describes the equivalent to make each marginal distribution approximately normal. Although normal margins are not sufficient to ensure that the joint distribution is normal, in practical applications, they may be good enough. If not, it could start with the values $\hat{\lambda}_1, \hat{\lambda}_2$, …, $\hat{\lambda}_p$ obtained from the proceeding transformations and be iterated toward the set of values $\lambda' = [\lambda_1, \lambda_2, \ldots, \lambda_p]$ which collectively maximizes $\ell(\lambda_1, \lambda_2, \ldots, \lambda_p)$

$$= -n/2 \ln \left| S(\lambda) \right| + (\lambda_1 - 1) \sum_{j=1}^{n} \ln x_{j1} + (\lambda_2 - 1) \sum_{j=1}^{n} \ln x_{j2} + \cdots + (\lambda_p - 1) \sum_{j=1}^{n} \ln x_{jp} \quad \text{(A.8)}$$

where $S(\lambda)$ is the sample covariance matrix computed from

$$x_j^{(\hat{\lambda})} = \begin{bmatrix} x_{j1}^{(\lambda_1)} - 1 / \lambda_1 \\ x_{j2}^{(\lambda_2)} - 1 / \lambda_2 \\ \vdots \\ x_{jp}^{(\lambda_p)} - 1 / \lambda_p \end{bmatrix} \quad \text{(A.9)}$$

$$j = 1, 2, \ldots, n$$

Maximizing Equation A.7 not only is substantially more difficult than maximizing the individual expressions in Equation A.4, but is also unlikely to yield remarkably better results. The selection method based on Equation A.7 is equivalent to maximizing a multivariate likelihood over μ, Σ, and λ, whereas the method based on Equation A.4 corresponds to maximizing the *k*th univariate likelihood over μ_k, σ_{kk}, and λ_k. The latter likelihood is generated by pretending there are some λ_k for observations $\left(x_{jk}^{\lambda_1} - 1\right) / \lambda_k$, $j = 1, 2, \ldots, n$ have a normal distribution.

RESULTS AND DISSCUSSION

Extracting Process of Dataset

There were several different programs used for analysis (normal and intrusion programs). There are three steps to perform an extracting process to obtain characteristics of system calls: the number of system calls triggered by programs running on active processes, the number of processes identified, and the characteristic of system calls to an activity. The overall results of extracting data are shown in Table 1 and Table 2. As seen in Table 1, the research performed extraction process on normal activities and intrusive activities as shown in Table 2, where the activities are divided into 38 normal activities and 11 intrusive activities.

Table 1. Extracting data for normal activities

Program	Normal activity data		
	Number of Traces	Number of Process	Number of System calls
inetd	1	3	544
login	2	35	8912
ps	0	0	0
ftp	2	16	4347
xlock	1	1	179931
named	32	32	178127

Table 2. Extracting data for normal activities

Program	Intrusive activity data		
	Number of Traces	Number of Process	Number of System calls
inetd	1	28	8371
login	2	10	4857
ps	2	4	1800
ftp	2	2	949
xlock	1	4	1800
named	3	1997	329969

The statistical summaries of extracting results are shown in Table 3. From the Table 3, the research selected a few system calls as variables for analysis during extraction process. These variables were selected based on some of the system calls that

correlate to intrusions as introduced by IPA (International of Promotion Agent) of Japan and SNARE of Australia's group research. This research combined these system calls and obtained nine of them that will be in the analysis (i.e., *chdir*, *geteuid*, *open*, *read*, *setgid*, *setuid*, *exit*, *getpgrp*, and *unlink*).

Table 3. Numerical summaries data sets

Variable	N	Sum	Mean 1.0e+003	Std Deviation 1.0e+004
chdir	49	2668	0.054	0.023
geteuid	49	1604	0.033	0.021
open	49	20033	0.409	0.116
read	49	135335	2.762	1.031
setgid	49	135	0.003	0.000
setuid	49	299	0.006	0.001
exit	49	4919	0.100	0.069
getpgrp	49	2664	0.054	0.037
unlink	49	107	0.002	0.000

Detecting Outlier Data

This research, performed the steps to detect outliers which were based from the designed algorithms, they are:

(i) *To make a dot plot for each variable.*
One of the difficulties inherent in multivariate statistics is the problem of visualizing multidimensionality. To rectify this problem, this research used Matlab plot command to display a graph of the relationship between system call variables as shown in Figure 6.

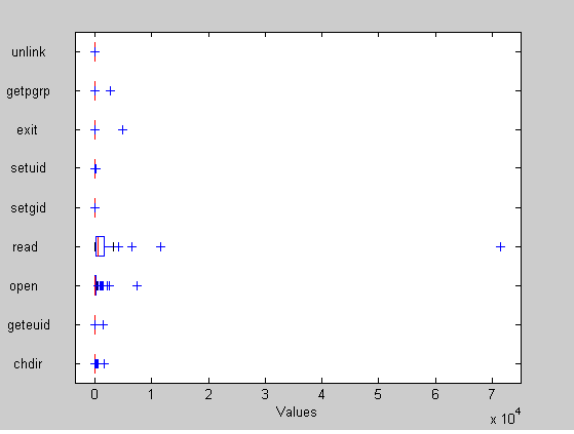

Figure 6. Box plots of system call variables

From the Figure 6, some of features were far from others. There is substantially more variability in the system calls of the *read* (135335) and *open* (20033) than in the other system call. This was caused by the presence of some redundant features of system calls in the original datasets. Therefore, to rectify this problem, the evaluations had to exclude redundant features and find out the most independent and the best

2. To make a scatter plot for each pair of variables.

To construct the scatter plot, the distance (Equation A.9) of each activity must be arranged from smallest to largest. Figure 7 shows the scatter plot for each pair of variables.

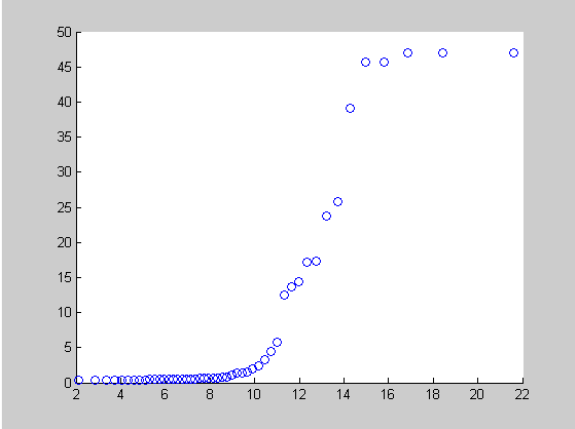

Figure 7. Scatter plots for the system call data

The points as in Figure 7 were found not lying long the line with slope1. The smallest distance was *xloc15* (*d2*=3. 4398).

3. To calculate the standardized values.
The standardized values are based on the sample mean and variance, calculated from 49 activities. They are calculated for each column of system call variables and examine these standard values for large or small value. In this research, "large" must be interpreted relative to the sample size and the number of variables. There is *no* X *p* standardized values. In this paper $n = 49$ and $p = 9$, there are 441 values. The value 6.8571 on standardized data might be considered large for moderate sample size.

4. To calculate the generalized square distance.
The generalized distances are calculated using Equation A.9 and they are examined for unusually large values. In a scatter plot, these would be the points farthest from the origin. The last column

reveals that the activities *login1*, *ftp*, *xlock17*, *inetd-i*, *login1-i*, *ps2-i*, *xlock1-i*, and *ftp-i*, are a multivariate outlier, since $X_9^2(.005) = 23.59$; yet all of individual measurements are well within their respective univariate scatter. Activities *login2*, *ps1*, *xlock13*, *xlock16*, and *login2-i*, also have large square distance values.

Table 4. The activities with large squared distance

No.	Activities	Squared Distance
1	login1	39.086
2	ftp	46.944
3	xlock17	23.763
4	inetd-i	46.940
5	login1-i	45.639
6	ps2	25.748
7	xlock1-i	47.020
8	ftp-i	45.655
9	login2	14.468
10	ps1	13.675
11	xlock13	17.257
12	xlock16	17.178
13	login2-i	12.560

The thirteen activities (*login1*, *ftp*, *xlock17*, *inetd-i*, *login1-i*, *ps2-i*, *xlock1-i*, and *ftp-i*, *login2*, *ps1*, *xlock13*, *xlock16*, and *login2-i*) with large squared distances (Table 4) stand out from the rest of the pattern in Figure 8. Once these thirteen points are removed, the remaining pattern conforms to the expected straight-line relation.

Transformation Process

The scatter plot of system call data in Figure 7 indicates that the activities deviate from what would be expected if they were normally distributed. Since all the observations are positive, it performs a power transformation of the data which, it hope will produce results that are more nearly normal. Restricting the research attention to the family of Box-Cox transformations in Equation 2.7, the analysis must find that value of $\lambda_1, \lambda_2, \ldots, \lambda_p$ (p = measured characteristics) maximizing the function $l_1(\lambda), l_2(\lambda), .. , \ell_k(\lambda)$ (k = system call variables), in Equation A.9. The pairs (λ, $l(\lambda)$) are listed in the following Table 5 for several values of λ.

Table 5. The value of λ maximizing the function $\ell_k(\lambda)$

λ	$\ell_1(\lambda)$	$\ell_2(\lambda)$	$\ell_3(\lambda)$	$\ell_4(\lambda)$
-1	-72.465	**-21.070**	-308.623	-467.927
-0.9	-71.035	-21.432	-293.234	-445.893
-0.8	-69.962	-22.045	-278.461	-424.615
-0.7	-69.307	-22.948	-264.555	-404.298
-0.6	**-69.150**	-24.197	-251.904	-385.222
-0.5	-69.593	-25.873	-241.062	-367.754
-0.4	-70.772	-28.103	-232.684	-352.342
-0.3	-72.867	-31.079	-227.317	-339.479
-0.2	-76.115	-35.112	**-225.136**	-329.613
-0.1	-80.807	-40.667	-225.885	**-323.068**
0	-87.284	-48.375	-229.082	-320.015
0.1	-95.881	-58.925	-234.280	-320.524
0.2	-106.855	-72.782	-241.172	-324.612

$\ell_5(\lambda)$	$\ell_6(\lambda)$	$\ell_7(\lambda)$	$\ell_8(\lambda)$	$\ell_9(\lambda)$
3.003	-47.884	**11.659**	**38.402**	**30.542**
2.032	-46.545	10.418	36.191	28.655
0.889	-45.570	8.888	33.674	26.593
-0.442	-44.991	6.995	30.765	24.342
-1.976	-44.842	4.632	27.333	21.881
-3.731	-45.151	1.637	23.184	19.189
-5.724	-45.948	-2.242	18.027	16.240
-7.977	-47.257	-7.397	11.444	13.006
-10.508	-49.099	-14.410	2.887	9.454
-13.339	-51.491	-24.064	-8.258	5.546
-16.489	-54.447	-37.187	-22.495	1.243
-19.978	-57.975	-54.317	-40.027	-3.498
-23.822	-62.080	-75.415	-60.682	-8.717

The curve of $\ell_k(\lambda)$ versus λ that allows the more exact determination $\lambda_1 = -0.6$, $\lambda_2 = -1$, $\lambda_3 = -0.2$, $\lambda_4 = -0.1$, $\lambda_5 = -1$, $\lambda_6 = -0.6$, $\lambda_7 = -1$, $\lambda_8 = -1$, and $\lambda_9 = -1$.

It is evident from both the table 5 and the plot values of λ maximize $\ell_k(\lambda)$. This research choose these λ, the data x_j in Equation A.9 were reexpressed as:

$$x_j^{(\hat{\lambda})} = \begin{bmatrix} \frac{x_{j1}^{(-0.6_1)} - 1}{-0.6} \\ \frac{x_{j2}^{(-1)} - 1}{-1} \\ \frac{x_{j3}^{(-0.2)} - 1}{-0.2} \\ \frac{x_{j4}^{(0.1)} - 1}{0.1} \\ \frac{x_{j5}^{(-1)} - 1}{-1} \\ \frac{x_{j6}^{(-0.6)} - 1}{-0.6} \\ \frac{x_{j7}^{(-1)} - 1}{-1} \\ \frac{x_{j8}^{(-1)} - 1}{-1} \\ \frac{x_{j9}^{(-1)} - 1}{-1} \end{bmatrix} \quad j = 1, 2, 3, \ldots, 49$$

A scatter plot was constructed from the transformed quantities. This plot is shown in Figure 8.

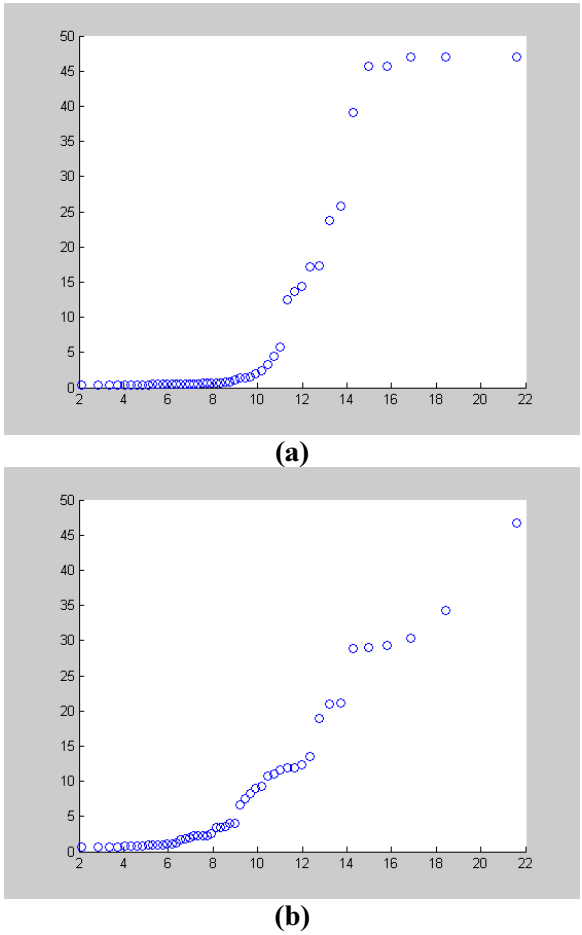

Figure 8. Scatter plots of (a) the original and (b) the transformed system call

As shown in Figure 8 (b), the quantile pairs fall very close to a straight line, this research would conclude from this evidence that $x_j^{-0.6}, x_j^{-1}, x_j^{-0.2}, x_j^{0.1}, x_j^{-1}, x_j^{-0.6}, x_j^{-1}, x_j^{-1}$, and x_j^{-1} are approximately normal.

CONCLUSIONS

Outliers occur when the relative frequency distribution of the data set is extremely skewed, because such a distribution of the data set has a tendency to include extremely large or small observations. If outliers are identified, they should be examined for content, as was done in the case of the data on system call in this paper. Depending upon the nature of the outliers and the objectives of the investigation, outliers may be deleted or appropriately weighted in a subsequent analysis. Even though many statistical techniques assume normal populations, those based on the sample mean vectors usually will not be disturbed by a few moderately outliers.

Referring to transform data, it is understood that the transformation obtained by maximizing $\ell_k(\lambda)$ usually improves the approximation to normality. However, there is no guarantee that even the best choice of λ will produce a transformed set of values that adequately conform to a normal distribution.

ACKNOWLEDGMENTS

The authors thank UTM FC Grid Computing Research Group for helpful discussions and suggestions. We also thank the MIT AI Lab and the UNM CS researcht group for allowing us to collect data on their production systems.

REFERENCES

Bace, R., & Mell, P. (2001). Intrusion Detection Systems. National Institute of Standards and Technology (NIST). Technical Report, 800-31.

Mukkamala, S., Janoski, G., & Sung, A. (2002). Intrusion Detection Using Neural Networks and Support Vector Machines. *Proceedings of the 2002 International Joint Conference on Neural Networks (IJCNN'2002)*. May 12-15. Honolulu, USA, 1702-1707.

Stavroulakis, P., & Stamp, M. (2010). Handbook of Information and Communication Security. New York: Springer-Verlag.

Kendall, K. (1999). *A Database of Computer Attacks for the Evaluation of intrusion Detection Systems*. Bachelor Thesis, Massachusetts Institute of Technology.

Debar, H., Dacier, M., & Wespi, A. (2000). A Revised Taxonomy for Intrusion Detection System. *Annual Journal of Telecommunications*. (55), 361-78.

Axelssons. (2000). Intrusion Detection Systems: a Survey and Taxonomy. Technical Report. 99-15 (2000), 1-27.

Lazarevic, A., Kumar, V., and Srivastava, J. (2005). Managing Cyber Threats, Issues, Approaches, and Challenges. New York: Springer-Verlag.

Jhonson, R. A., and Wichern, D. W. (1998). *Applied Multivariate Statistical Analysis*, 4th Ed. New Jersey: Prentice Hall.

Cohen, W. (1995). Fast effective rule induction. In *Machine Learning: the 12th International Conference*. Morgan Kaufmann.

Damashek, M. (1995). Gauging similarity with n-grams: Languageindependent categorization of text. *Journal of Science*, (267):843–848.

Hofmeyr, S. A., Forrest, S., and Somayaji, A. (1998). Intrusion detection using sequences of system calls. *Journal of Computer Security*, (6):151–180.

W. Lee, W., Stolfo, S. J., and Chan, P. K. (1997). Learning patterns from UNIX process execution traces for intrusion detection. In *AAAI Workshop on AI Approaches to Fraud Detection and Risk Management*, pages 50–56. AAAI Press.

Lee, W. and Stolfo, S. J. (1998). Data mining approaches for intrusion detection. In *Proceedings of the 7th USENIX Security Symposium*.