Handling Nominal Features in Anomaly Intrusion Detection Problems

Mei-Ling Shyu[1], Kanoksri Sarinnapakorn[1], Indika Kuruppu-Appuhamilage[1],
Shu-Ching Chen[2], LiWu Chang[3], and Thomas Goldring[4]
[1]Department of Electrical and Computer Engineering, University of Miami, Coral Gables, FL 33124
{shyu,ksarin}@miami.edu, ikuruppu@umsis.miami.edu
[2]School of Computer Science, Florida International University, Miami, FL 33199
chens@cs.fiu.edu
[3]Center for High Assurance Computer Systems, Naval Research Laboratory, Washington, DC 20375
lchang@itd.nrl.navy.mil
[4]National Security Agency
tgo@tycho.ncsc.mil

Abstract

Computer network data stream used in intrusion detection usually involve many data types. A common data type is that of symbolic or nominal features. Whether being coded into numerical values or not, nominal features need to be treated differently from numeric features. This paper studies the effectiveness of two approaches in handling nominal features: a simple coding scheme via the use of indicator variables and a scaling method based on multiple correspondence analysis (MCA). In particular, we apply the techniques with two anomaly detection methods: the principal component classifier (PCC) and the Canberra metric. The experiments with KDD 1999 data demonstrate that MCA works better than the indicator variable approach for both detection methods with the PCC coming much ahead of the Canberra metric.

Keywords: Anomaly detection, intrusion detection, indicator variables, multiple correspondence analysis, nominal features, principal component classifier.

1. Introduction

Data streams are ordered sequences of vast continuous data that can be read/accessed only once or a small number of times. Sources of data streams are ubiquitous in daily life. Network intrusion detection systems handle computer network data streams as one of its applications. Currently used intrusion detection systems are classified into two major categories: signature based and anomaly based. A signature recognition system, such as Bro [18] or SNORT [21], operates in much the same way as a virus scanner by searching for a known identity or signature for each specific intrusion event. In contrast, an anomaly detection system such as SPADE [26], NIDES [1], ADAM [2], or PHAD [14] builds a model of normal traffic and detects deviations from the normal model. A large departure of any traffic from the normal model is likely to be anomalous. An extensive review of various approaches to novelty detection can be found in [15, 16].

An ingredient to the success of any intrusion detection system is a set of meaningful features that are extracted from a data stream of network traffic. The features can be quantitative or qualitative. That is, the data can have any scales of measurement, nominal or ordinal scales for qualitative features, and interval scales for quantitative features. As an example, network traffic data may consist of features such as the protocol type in nominal scales, user authorization level in ordinal scales, and package size in interval scales. A feature measured in nominal or ordinal scales can take on numerical or non-numerical values. For nominal scales, the numbers only indicate the category and serve as the names or labels to distinguish one category from another with no orders involved. Which number is assigned to which category is completely arbitrary. On the other hand, in ordinal scales, a feature has the values that indicate not only the category but also the magnitude. The categories have a fixed a priori order that can be ranked. However, if numerical values are used, the distance between scale points need not be equal. In contrast, a quantitative feature in interval scales has numerical values on a well-defined scale, either discrete or continuous, which tells the magnitude with the property of an equal distance. The difference between the scale points is interpreted as the distance between cases on that feature [27].

Some intrusion detection methods process qualitative data naturally, e.g., k-nearest neighbor and decision tree algorithm; while some methods only work with quantitative data, e.g., the methods based on some distance measures and the principal component classifier (PCC) [24, 25]. For those methods in the latter group, qualitative data need to be transformed to numerical values prior to the analysis. Many approaches to quantify qualitative variables have been proposed in the literature. It can be as simple as mapping each category to sequential integer values [10] or converting the name of a category to a decimal number by adding the ASCII values of all its characters [12]. Some researchers go one step further, after getting the integer values for all categories, by implementing scaling. For instance, the nominal values are mapped into integer values ranging from 0 to C-1, where C is the number of categories the nominal feature has. Then each feature is linearly scaled to the range [0.0, 1.0] [22].

Despite the simplicity, there are some criticisms to the aforementioned approaches. Since the categories of a nominal feature are merely labels with no fixed order, the different ordering of the categories will lead to different numerical values for each category. Besides, even with features measured in ordinal scales, assuming an equal distance or a linear scale usually is not sensible. A more preferable method in statistical analysis to quantify the categories of a qualitative variable is to employ the indicator or dummy variables. Indicator variables convert qualitative information into quantitative information by means of a binary coding scheme [17].

The scaling of qualitative variables inevitably affects the validity of the results from an intrusion detection method. It is our interest in this paper to explore the use of a scaling method from multiple correspondence analysis (MCA) in the intrusion detection problem. Simple correspondence analysis (CA) is typically used as a graphical technique to study the association of two qualitative variables in a two-way contingency table [6]. The extension of simple CA to more than two qualitative variables case is called MCA. We will examine the effectiveness of MCA as compared to the indicator variable approach in dealing with non-numeric data for some intrusion detection methods. Specifically, we will enhance the ability of the PCC method to handle all types of data. As the method constructs a classifier from principal components, it requires data to be numeric. The experiments with PCC conducted in [24] demonstrated its good performance with numeric features. The method has high detection rates and low false alarms, and does not need to make any distributional assumption on data. It is expected that the additional information on traffic behaviors contained in other non-numeric features, when properly used, will increase the accuracy of the detection further.

The organization of the paper is as follows. Section 2 discusses the application of the indicator variables and the algebraic development of correspondence analysis. The use of MCA as a scaling technique for nominal features is illustrated through a set of experiments with the PCC scheme and another anomaly detection method based on the Canberra metric. Section 3 describes the experimental setting in details and provides an overview of these two detection methods. The results of the experiments are summarized and discussed in Section 4. Section 5 concludes our study.

2. Handling Nominal Features

Nominal features are common in network traffic data stream. However, many intrusion detection methods, particularly statistical based, are designed for numerical data. In order for these methods to utilize information from nominal features in detection, some coding schemes or transformations are exploited. We consider two approaches in this study, namely indicator variables and the multiple correspondence analysis.

2.1. Indicator (Dummy) Variables

There are many different but equivalent ways to quantitatively identify the categories of a nominal feature. The most familiar coding scheme is the binary coding which simply uses a 1 to indicate the occurrence of a category of interest and a 0 to indicate its nonoccurrence [17]. Accordingly, for a nominal feature with C distinct categories, a set of C indicator variables can be generated.

$$x_1 = \begin{cases} 1 & \text{if the category is 1} \\ 0 & \text{otherwise} \end{cases}$$

$$x_2 = \begin{cases} 1 & \text{if the category is 2} \\ 0 & \text{otherwise} \end{cases}$$

$$\vdots$$

$$x_{\mathrm{C}} = \begin{cases} 1 & \text{if the category is C} \\ 0 & \text{otherwise} \end{cases}$$

Since the C indicator variables are linearly dependent, any C-1 out of the C variables sufficiently identify a category.

It is not unusual to find some nominal features in network traffic having a large number of different categories. The conversion of these features to indicator variables will increase the dimensionality of data greatly. Thus, it is helpful to reduce the number of categories by grouping similar values into a few numbers of essential categories before creating indicator variables. Some clustering techniques may be used to achieve this purpose.

2.2. Correspondence Analysis

Categorical data come from the population that has a discrete distribution and require a different type of analysis from continuous data. In the analysis of the categorical data, typically, the first step is to crosstabulate the data and to present the frequencies or counts in a two-way or multiway contingency table. An exploratory technique called correspondence analysis (CA) is designed to analyze the correspondence between the rows and columns of a contingency table with the objective to represent the associations in the table in a low-dimensional space. The singular value decomposition (SVD) is utilized to analyze the data matrix in CA. The outcome of such an analysis is usually a pair of bivariate plots superimposed on one another showing the spatial relationships among the categories of categorical variables. In essence, it is a weighted form of principal component analysis that is appropriate for frequency or categorical data [23].

Using the case of two qualitative variables as described in [8], let $\mathbf{N}$ be an $I \times J$ two-way contingency table whose (i, j) entry is n_{ij}, and then the rows and columns of $\mathbf{N}$ correspond to different categories of two variables. Let $I \geq J$ and assume that $\mathbf{N}$ is of full column rank J. The correspondence matrix $\mathbf{P}$ is defined as $\mathbf{P} = \frac{1}{n}\mathbf{N}$, where n is the total of the frequencies in $\mathbf{N}$.

Furthermore, let $\mathbf{1}' = (1, 1, \ldots, 1)$, $\underset{I\times 1}{\mathbf{r}} = \underset{I\times J}{\mathbf{P}}\ \underset{J\times 1}{\mathbf{1}}$ and $\underset{J\times 1}{\mathbf{c}} = \underset{J\times I}{\mathbf{P}'}\ \underset{I\times 1}{\mathbf{1}}$ be the vectors of row and column sums of $\mathbf{P}$, and $\mathbf{D}_r$ and $\mathbf{D}_c$ be the diagonal matrices whose diagonal entries are the elements of $\mathbf{r}$ and $\mathbf{c}$ respectively. Then $\mathbf{P} - \mathbf{rc}'$ can be viewed as the matrix of the residuals when fitting the independence model to $\mathbf{P}$, and the scaled matrix $\mathbf{P}^*$ can be constructed as shown in Equation 1, where $\text{rank}(\mathbf{P}^*) = \text{rank}(\mathbf{P} - \mathbf{rc}') \leq J - 1$.

$$\underset{I\times J}{\mathbf{P}^*} = \underset{I\times I}{\mathbf{D}_r^{-1/2}}\,(\underset{I\times J}{\mathbf{P}} - \mathbf{rc}')\,\underset{J\times J}{\mathbf{D}_c^{-1/2}} \tag{1}$$

CA is based on the generalized SVD of $\mathbf{P} - \mathbf{rc}'$, which is equivalent to SVD of $\mathbf{P}^*$ [9]. Let $\mathbf{U}$ and $\mathbf{V}$ be two orthogonal matrices, and $\mathbf{\Lambda}$ be a diagonal matrix that contains the singular values ordered from largest to smallest ($\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_{J-1} > 0$). Hence, $\mathbf{P}^*$ can be represented as given in Equation 2.

$$\underset{I\times J}{\mathbf{P}^*} = \underset{I\times(J-1)}{\mathbf{U}}\ \underset{(J-1)\times(J-1)}{\mathbf{\Lambda}}\ \underset{(J-1)\times J}{\mathbf{V}'} \tag{2}$$

If we define $\tilde{\mathbf{U}} = \mathbf{D}_r^{1/2}\mathbf{U}$ and $\tilde{\mathbf{V}} = \mathbf{D}_c^{1/2}\mathbf{V}$, we have the following:

$$\underset{(J-1)\times I}{\tilde{\mathbf{U}}'}\ \underset{I\times I}{\mathbf{D}_r^{-1}}\ \underset{I\times(J-1)}{\tilde{\mathbf{U}}} = \underset{(J-1)\times(J-1)}{\mathbf{I}}$$

$$\underset{(J-1)\times J}{\tilde{\mathbf{V}}'}\ \underset{J\times J}{\mathbf{D}_c^{-1}}\ \underset{J\times(J-1)}{\tilde{\mathbf{V}}} = \underset{(J-1)\times(J-1)}{\mathbf{I}}$$

From Equations 1 and 2, the SVD of $\mathbf{P} - \mathbf{rc}'$ can be expressed as shown in Equation 3, where the columns of $\tilde{\mathbf{U}}$ define the principal axes for the points representing the column profiles of $\mathbf{P}$, and the columns of $\tilde{\mathbf{V}}$ define the principal axes for the points representing the row profiles of $\mathbf{P}$. These lead to the end-products of CA. That is, the coordinates of the row and column profiles are given in Equations 4 and 5, respectively.

$$\mathbf{P} - \mathbf{rc}' = \tilde{\mathbf{U}}\mathbf{\Lambda}\tilde{\mathbf{V}}' \tag{3}$$

$$\underset{I\times(J-1)}{\mathbf{Y}} = \underset{I\times I}{\mathbf{D}_r^{-1}}\ \underset{I\times(J-1)}{\tilde{\mathbf{U}}}\ \underset{(J-1)\times(J-1)}{\mathbf{\Lambda}} \tag{4}$$

$$\underset{J\times(J-1)}{\mathbf{Z}} = \underset{J\times J}{\mathbf{D}_c^{-1}}\ \underset{J\times(J-1)}{\tilde{\mathbf{V}}}\ \underset{(J-1)\times(J-1)}{\mathbf{\Lambda}} \tag{5}$$

The coordinate pairs of the row (or column) points in the best two-dimensional representation of the data are in the first two columns of $\mathbf{Y}$ (or $\mathbf{Z}$), and produce one bivariate plot in which each row (or column) category is plotted. The two pairs of dimensions (one for the row profiles and one for the column profiles) are merely a scaling of the row and column categories. Therefore, CA can be regarded as a scaling method that determines the scales when the amount of variation explained among profile deviations is maximized.

However, SVD can only be used in simple CA with two-way tables, but in a general case when the table is of a higher dimension than two, SVD will not directly work on $\mathbf{N}$. Instead, it will be applied to the indicator matrix $\mathbf{B}$ of size $[n \times (I + J)]$ as defined in [6]. In $\mathbf{B}$, its rows correspond to the observations; while its first *I* (or last *J*) columns are the indicator variables corresponding to the categories of the row (or column) variable of $\mathbf{N}$. For each observation, the value *1* is assigned to the indicator column for each qualitative variable; whereas the remaining indicator columns in that row are zeroes. Greenacre noted the close connection between the CA of $\mathbf{N}$ and $\mathbf{B}$ [6]. For example, when there are n observations from Q categorical variables, $\mathbf{B}$ will have n rows and J columns, where $J = J_1 + J_2 + \ldots + J_{\text{Q}}$, J_q denotes the number of different categories of the q^{th} variable, and $q = 1, 2, \ldots, \text{Q}$. It was also shown that the standard coordinates of the columns in the analysis of $\mathbf{B}$ are identical to the standard coordinates of the rows or columns in the analysis of the inner product of the indicator matrix ($\mathbf{B}'\mathbf{B}$), which is called Burt matrix.

In fact, MCA offers insightful information into the relationships among the categorical variables via visual displays of the first two principal axes from MCA. Hence, the knowledge of the relationships provided by the principal axes should be valuable in detection. In our proposed approach, all the nominal features that have more than two categories will first be scaled using MCA, and then the first two principal axes will be used to represent each of these nominal features. For the nominal features that have only two categories, a binary variable with values 0 and 1 is used.

3. Experiments

To study the potency of MCA and indicator variables in coping with the nominal features found in intrusion detection problems, several experiments on two anomaly detection methods using the KDD 1999 data are conducted. The two methods are the PCC scheme and the detection method based on Canberra metric.

3.1 The KDD CUP 1999 Data

Due to the fact that it is forbidden to generate real intrusions in the real network environment, in this paper, we use the Knowledge Discovery and Data Mining (KDD) 1999 data set in our experiments since KDD CUP 1999 data has been widely used for testing an intrusion detection system [5, 19, 22]. Though we may collect real traffic data from the real network environment, for the purpose of evaluating the intrusion detection methods, we first need to identify the attacks either via expert judgment or via some well-known intrusion detection tools such as BRO and SNORT. However, this does not guarantee the identification will be error-free, and the evaluation will be obscured by such errors. Furthermore, real traffic data usually do not contain a wide variety of attack types and it takes a long time to gather enough attack data to provide an adequate testing. On the other hand, the KDD data set contains a rich set of different attack types, and it is well-accepted and publicly available.

As the KDD test data are from different probability distribution than its training data, only the KDD training data is used in our study. It has 494,021 connection records in the training data set. Here, a connection is a sequence of TCP packets containing values of 41 features and labeled as either normal or an attack. There are 24 attack types, but we treat all of them as one attack group due to our interest in detecting any connections that are not normal. The features include 34 numeric features and 7 symbolic features. For nomial features, some have many different values, and some values have only very few observations. The number of categories for all nominal features is given in Table 1. A complete listing of features and details can be found in [11, 13].

For example, the Service and Flag features have many categories but few observations in some categories. It is better to reduce the number of categories by combining some of them when taking the indicator variables approach. For this purpose, we adopt a clustering technique that uses domain knowledge as the basic concept [3]. For the Service feature, there are 64 discrete values denoting the network service on the destination, and each service is assigned to one of the ports used in TCP [20] to name the ends of the logical connections which carry long term conversations. To provide services to unknown callers, a service contact port (a well-known port) assigned by the Internet Assigned Numbers Authority [7] is defined. The well-known ports range from 0 to 1023. Besides the well-known ports, there are two other categories of ports named *Registered Ports* and *Dynamic and/or Private Ports*. The *Registered Ports* are those from 1024 to 49151 and the *Dynamic and/or Private Ports* are spanned from 49152 through 65535. By analyzing the applications of the ports that are used by the services residing in the training set in details, each port is then grouped into eight clusters depending on their usage. The basic clusters are given as follows.

1. Services used to get the remote access of another machine (e.g., telnet, ssh).
2. Services used in file and document transfer (e.g., ftp, tftp).
3. Services used in mail transfer (e.g., smtp, imap4).
4. Services used in web applications (e.g., http).
5. Services used to get system parameters and statistics (e.g., systat, netstat).
6. Services used in name servers (e.g., hostname, domain).
7. Services used in ICMP protocol.
8. Others.

Table 1. Nominal features used in experiments

Features	Description	# Categories
Protocol	Type of the protocol (TCP, UDP,ICMP)	3
Service	Network service on the destination, e.g. http, telnet, ftp, etc.	64
Flag	Normal or error status of the connection	11
Land	1 if connection is from/to the same host port; 0 otherwise	2
logged-in	1 if successfully logged in; 0 otherwise	2
is_host_login	1 if the login is a "host" login; 0 otherwise	2
is_guest_login	1 if the login is a "guest" login; 0 otherwise	2

The Flag feature of a connection consists of 13 values and we cluster them by the nature of the connections into 6 groups as shown in Table 2. Note that the Flag feature in the KDD 1999 data only has 11 values.

Table 2. Clusters of flag features

Cluster	Name	Description
Cluster 1	S0	Connection attempt was seen, no reply
	REJ	Connection attempt was rejected
Cluster 2	S1	Connection was established but not terminated
	SF	Normal establishment and termination
	OTH	No SYN was seen, just midstream traffic
Cluster 3	S2	Connection was established and close attempt by originator was seen (but no reply from responder)
	RSTO	Connection was established, originator aborted (sent a RST)
Cluster 4	S3	Connection was established and close attempt by responder was seen (but no reply from originator)
	RSTR	Established, responder aborted
Cluster 5	RSTOS0	Originator sent a SYN followed by a RST, SYN ACK was not seen from the responder
	SH	Originator sent a SYN followed by a FIN, SYN ACK was not seen from the responder
Cluster 6	RSTRH	Responder sent a SYN ACK followed by a RST, SYN was not seen from the originator
	SHR	Responder sent a SYN ACK followed by a FIN, SYN was not seen from the originator

3.2 Anomaly Detection by Canberra Metric

Many anomaly detection methods base their detection criterion on some distance measure. Here we use the Canberra metric that was studied by Emran and Ye [4].

The Canberra metric is defined for nonnegative variables only. Let $\mathbf{x} = (x_1, x_2, \ldots, x_p)'$ and $\mathbf{y} = (y_1, y_2, \ldots, y_p)'$ be two p-dimensional observations. The distance between observations **x** and **y** as measured by the Canberra metric is

$$d(\mathbf{x}, \mathbf{y}) \;=\; \sum_{j=1}^{p} \frac{|x_j - y_j|}{(x_j + y_j)} \qquad (6)$$

The procedure commonly used to detect multivariate outliers in a data set is to measure the distance of each observation from the center of the data. A large distance value would indicate the observation might be an outlier.

Let $\mathbf{X}_1$, $\mathbf{X}_2$, ..., $\mathbf{X}_n$ be a random sample from a multivariate distribution with the mean vector μ, where $\mathbf{X}_i = (X_{i1}, X_{i2}, \ldots, X_{ip})'$, $i = 1, 2, \ldots, n$. The sample mean vector is $\bar{\mathbf{X}} \;=\; \frac{1}{n} \sum_{i=1}^{n} \mathbf{X}_i$ and the Canberra distance of an observation **X** from the mean vector is $d(\mathbf{X}, \bar{\mathbf{X}})$. Any observation **X** that has the distance larger than a threshold value is considered an outlier. Since the distribution of this distance is hard to derive even under the normality assumption, the threshold is figured from the empirical distribution of the distance.

3.3 Principal Component Classifier (PCC)

PCC differentiates attacks from normal instances using an outlier detection rule which is constructed from principal components of normal training sample [25]. Let $y_1, y_2, \ldots, y_p$ be principal component scores of **x**, and $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_p \geq 0$ be eigenvalues of the sample correlation matrix. The classifier consists of two functions, the major components part $\sum_{j=1}^{q} \frac{y_j^2}{\lambda_j}$ and the minor components $\sum_{j=p-r+1}^{p} \frac{y_j^2}{\lambda_j}$. Shyu et al. [25] suggested using q major components that can explain about 50 percents of the total variation in the standardized features and using r minor components whose variances or eigenvalues are less than 0.20. The decision criterion is of the form:

Classify an instance **x** as an attack if

$$\sum_{j=1}^{q} \frac{y_j^2}{\lambda_j} \;>\; c_1 \quad \text{or} \quad \sum_{j=p-r+1}^{p} \frac{y_j^2}{\lambda_j} \;>\; c_2$$

Classify **x** as a normal instance if

$$\sum_{j=1}^{q} \frac{y_j^2}{\lambda_j} \;\leq\; c_1 \quad \text{and} \quad \sum_{j=p-r+1}^{p} \frac{y_j^2}{\lambda_j} \;\leq\; c_2$$

c_1 and c_2 are critical values such that the classifier would produce the desired false alarm rate. They are typically set based on the empirical distributions of $\sum_{j=1}^{q} \frac{y_j^2}{\lambda_j}$ and $\sum_{j=p-r+1}^{p} \frac{y_j^2}{\lambda_j}$ in the training data. For more details on the PCC method, see [24, 25].

Table 3. Number of indicator variables for nominal features

Features	# Categories	# Indicator Variables
Protocol	3	2
Services	8	7
Flag	6	5
Land	2	1
Logged-in	2	1
is_host_login	2	1
is_guest_login	2	1

3.4 Experimental Framework

Our experiments involve training and testing the classifiers, and we sample the data from the KDD training data set to use in both stages. Out of the 494,021 connection records in the training set, 396,742 records are attacks and 97,279 records are normal connections. The experiments are carried out as follows.

- Each training data set consists of 5,000 normal connections selected by systematic sampling from 97,279 normal connection records in the KDD training data.
- To assess the accuracy of the classifiers, we do five independent experiments, each with different training samples. The classifiers are tested with a test set composed of 10% of the attack data. This amounts to 39,674 attack connections randomly selected from the 396,742 records in the KDD training data set.
- As the detection rate depends on the critical values which are determined by the false alarm rate that we specify, we vary the false alarm rate from 1% to 10% in the experiments. We then estimate the false alarm rate by the observed false alarms that actually take place using a total of 92,279 normal connections remained in the KDD training set after 5,000 records are selected as a training sample described earlier.
- For the indicator variables approach, we cluster and convert the Service and Flag features to indicator variables. This gives us a data set with 52 variables, 34 numeric and 18 indicator variables shown in Table 3.
- We apply MCA to the nominal features that have more than 2 categories. For those with only two categories, we convert them to binary variables having values 0 and 1. Hence, the data set consists of 44 variables, which include 34 original numeric variables, 4 nominal features that have binary values, 6 variables from principal axes resulting from applying MCA to the Protocol, Service, and Flag features.

Table 4. Average detection rates (%) of PCC and Canberra metric. Standard deviations of detection rates are shown in the parentheses.

False Alarm	PCC (Dummy)	PCC (MCA)	Canberra (Dummy)	Canberra (MCA)
1%	69.85 ($\pm$38.01)	99.20 ($\pm$0.33)	3.70 ($\pm$1.04)	17.64 ($\pm$31.81)
2%	99.21 ($\pm$0.26)	99.27 ($\pm$0.31)	4.86 ($\pm$0.91)	74.30 ($\pm$1.88)
4%	99.46 ($\pm$0.28)	99.55 ($\pm$0.37)	5.81 ($\pm$0.87)	75.77 ($\pm$2.68)
6%	99.72 ($\pm$0.18)	99.67 ($\pm$0.26)	14.05 ($\pm$4.06)	78.49 ($\pm$4.24)
8%	99.74 ($\pm$0.18)	99.69 ($\pm$0.22)	27.89 ($\pm$0.05)	91.41 ($\pm$11.35)
10%	99.79 ($\pm$0.18)	99.71 ($\pm$0.22)	30.75 ($\pm$5.92)	93.25 ($\pm$12.27)

4 Experimental Results and Discussion

Presented in Table 4 are the average detection rates from 5 independent experiments. In comparing the performance of the PCC and the Canberra metric when using indicator variables and MCA, it is obvious that MCA works much better than the indicator variables with the Canberra metric. For PCC, MCA and indicator variables do not give any significantly different results when the false alarm rate is 2% or more. Both yield very high detection rates, all above 99%. At 1% false alarm level, however, indicator variables cannot compete with MCA. The performance from the indicator variables is not consistent as seen from the standard deviation of 38.01%. Sometimes it can do well, and sometimes it does poorly. On the other hand, MCA can still help PCC detect very well with the detection rate higher than 99% again.

We take a closer look at the detection results for each attack type. In KDD 1999 training data, there are 24 attack types that fall into 4 big categories: DOS - denial-of-service, Probe - surveillance and other probing, U2R - unauthorized access to local superuser (root) privileges, and R2L - unauthorized access from a remote machine. Figures 1-4 show the receiver operating characteristic (ROC) curves of these four attack types. It can be easily seen from these figures that using MCA always helps us detect DOS attack type better than using the indicator variables approach. For

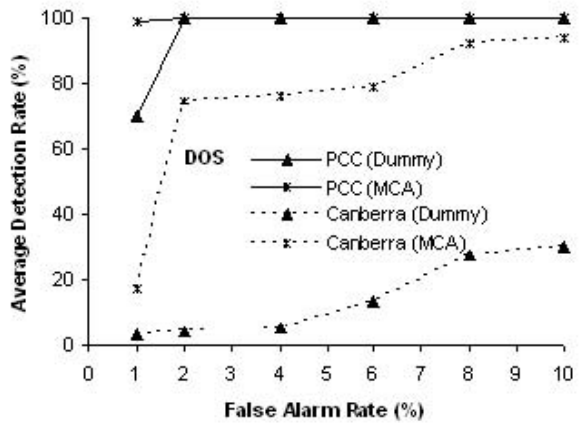

Figure 1. ROC curves for DOS attack type

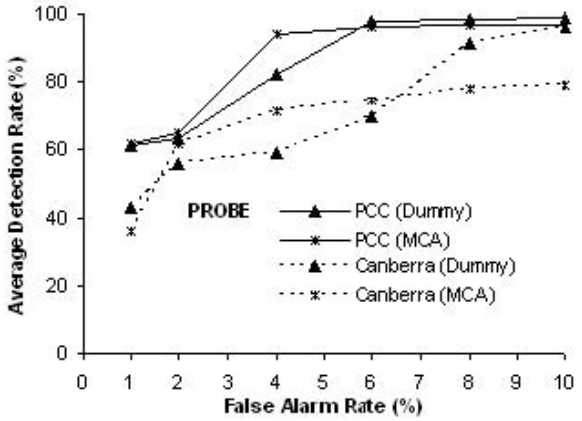

Figure 2. ROC curves for Probe attack type

other attack types, with PCC, most of the time, MCA is better or about the same as the indicator variables. With the Canberra metric, MCA may be better or may be worse in the Prob and R2L groups depending on the level of false alarm.

Indicator variables approach is easier to use than MCA, but when a feature has many different categories and some categories have very few observations, it may be necessary to combine some categories to reduce the number of categories to a more manageable level. As in our experiments, it appears that clustering the categories for the Service and Flag features can cause some information to be overlooked, and thus makes the indicator variables approach to be less effective. Overall speaking, we conclude that MCA is a better choice when we have to deal with nominal features.

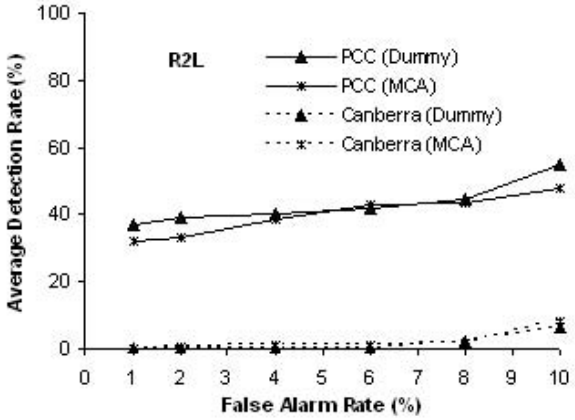

Figure 3. ROC curves for R2L attack type

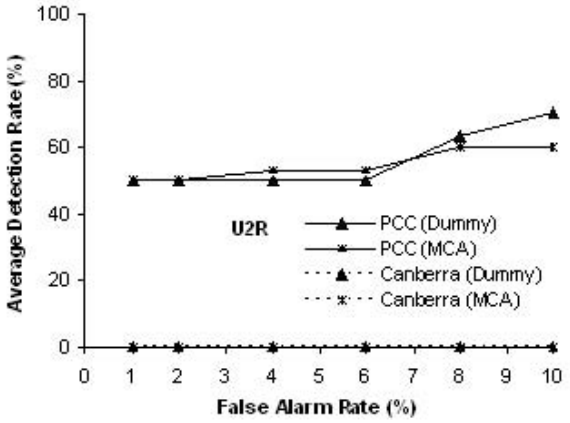

Figure 4. ROC curves for U2R attack type

5. Conclusion

This paper discusses how to handle the nominal features found in computer network data stream for intrusion detection purposes. We incorporate qualitative information from the symbolic features in the detection via the use of an optimal scaling method called MCA, and a coding scheme called indicator variables. Both approaches convert the categories of nominal features to numerical values, which then can be used together with other numeric features in any detection methods. In a comparative study where we experiment with two anomaly detection methods, the PCC and the Canberra metric, MCA is found to perform better than the indicator variables.

6. Acknowledgement

For Mei-Ling Shyu, this research was supported in part by NSF ITR (Medium) IIS-0325260. For Shu-Ching Chen and Mei-Ling Shyu, this research was supported in part by Naval Research Laboratory (NRL)/ITT: 176815J.

References

[1] D. Anderson, T. Lunt, H. Javitz, A. Tamaru, and A. Valdes. Detecting Unusual Program Behavior Using the Statistical Component of the Next-Generation Intrusion Detection Expert System (NIDES). Technical Report SRI-CSL-95-06,Computer Science Laboratory, SRI International, 1995.

[2] D. Barbara, N. Wu, and S. Jajodia. Detecting novel network intrusions using Bayes estimator. In *Proceedings of the First SIAM Int'l Conf. on Data Mining (SDM01)*, 2001.

[3] M. C. Clark, L. O. Hall, C. Li, and D. B. Goldgof. Knowledge based (re-)clustering. In *The 12th International Conference on Pattern Recognition*, pages 245–250, Jerusalem, Israel, October 1994.

[4] S. M. Emran and N. Ye. Robustness of Canberra metric in computer intrusion detection. In *Proceedings of the 2001 IEEE Workshop on Information Assurance and Security, US Military Academy*, pages 80–84, New York, 2001.

[5] J. Gomez and D. Dasgupta. Evolving fuzzy classifiers for intrusion detection. In *Proceedings of the 2002 IEEE Workshop on Information Assurance*, New York, June 2002.

[6] M. J. Greenacre. *Theory and Applications of Correspondence Analysis*. Academic Press, London, 1984.

[7] IANA:Internet Assigned Number Authority. http://www.iana.org.

[8] R. A. Johnson and D. W. Wichern. *Applied Multivariate Statistical Analysis, 5th Ed.* Prentice-Hall, New Jersey, 2002.

[9] I. Jolliffe. *Principal Component Analysis*. Springer-Verlag, 2002.

[10] H. G. Kayaclk. Hierarchical self organizing map based IDS on KDD benchmark. Master's thesis, Dalhousie University, 2003.

[11] KDD99. KDD Cup 1999 Data. http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html.

[12] K. Labib and R. Vemuri. NSOM: A real-time network-based intrusion detection system using self-organizing maps. *Networks Security, submitted*, 2002.

[13] W. Lee and S. Stolfo. A framework for constructing features and models for intrusion detection systems. *Transactions on Information and System Security*, 3(4):227–261, 2000.

[14] M. Mahoney and P. K. Chan. Learning nonstationary models of normal network traffic for detecting novel attacks. In *Proceedings of the 8th International Conference on Knowledge Discovery and Data Mining*, pages 376–385, 2002.

[15] M. Markou and S. Singh. Novelty detection: A review part 1: Statistical approaches. *Signal Processing*, 83(12):2481–2497, Dec. 2003.

[16] M. Markou and S. Singh. Novelty detection: A review part 2: Neural network-based approaches. *Signal Processing*, 83(12):2499–2521, Dec. 2003.

[17] J. Neter, M. H. Kutner, W. Wasserman, and C. J. Nachtschiem. *Applied Linear Statistical Models, 4th Ed.* McGraw-Hill/Irwin, New York, 1996.

[18] V. Paxson. Bro: A system for detecting network intruders in real-time. In *Proceedings of the 7th USENIX Security Symposium*, 1998.

[19] L. Portnoy, E.Eskin, and S. J.Stolfo. Intrusion detection with unlabeled data using clustering. In *Proceedings of the ACM CSS Workshop on Data Mining Applied to Security*, Philadelphia, PA, 2001.

[20] J. Postel. Transmission Control Protocol - DARPA Internet Program Protocol Specification. STD 7, RFC 793, USC/Information Sciences Institute, September 1981.

[21] M. Roesch. Snort - lightweight intrusion detecion for networks. In *Proceedings of USENIX LISA*, Nov 1999.

[22] M. Sabhnani and G. Serpen. Application of machine learning algorithms to KDD intrusion detection dataset within misuse detection context. In *Proceedings of the 2003 Int'l Conference on Machine Learning; Models, Technologies and Applications*, pages 623–630, New York, June 2003.

[23] SAS Online Doc version 8. SAS Institute Inc., 2000.

[24] M.-L. Shyu, S.-C. Chen, K. Sarinnapakorn, and L. Chang. A novel anomaly detection scheme based on principal component classifier. In *Proceedings of the IEEE Foundations and New Directions of Data Mining Workshop, in conjunction with the Third IEEE International Conference on Data Mining (ICDM'03)*, pages 172–179, Florida, Nov. 2003.

[25] M.-L. Shyu, S.-C. Chen, K. Sarinnapakorn, and L. Chang. *Principal Component-based Anomaly Detection Scheme*. Physica-Verlag, 2004. accepted for publication.

[26] S. Staniford, J. Hoagland, and J. McAlerney. Practical automated detection of stealthy portscans. *Journal of Computer Security*, 10(1–2):105–136, 2002.

[27] Thomson. Scale of Measurement, Statistics Workshops. http://www.wadsworth.com/, 2004. Thomson Wadsworth.