Journal of Zhejiang University-SCIENCE C (Computers & Electronics)
ISSN 1869-1951 (Print); ISSN 1869-196X (Online)
www.zju.edu.cn/jzus; www.springerlink.com
E-mail: jzus@zju.edu.cn

A new data normalization method for unsupervised anomaly intrusion detection*

Long-zheng CAI†1, Jian CHEN2, Yun KE1, Tao CHEN1, Zhi-gang LI1
([1]*Engineering and Commerce College, South-Central University for Nationalities, Wuhan 430065, China*)
([2]*Guangdong Institute of Science and Technology, Zhuhai 519090, China*)
†E-mail: charlescai@yahoo.cn

Received Oct. 18, 2009; Revision accepted Jan. 29, 2010; Crosschecked Aug. 2, 2010

Abstract: Unsupervised anomaly detection can detect attacks without the need for clean or labeled training data. This paper studies the application of clustering to unsupervised anomaly detection (ACUAD). Data records are mapped to a feature space. Anomalies are detected by determining which points lie in the sparse regions of the feature space. A critical element for this method to be effective is the definition of the distance function between data records. We propose a unified normalization distance framework for records with numeric and nominal features mixed data. A heuristic method that computes the distance for nominal features is proposed, taking advantage of an important characteristic of nominal features—their probability distribution. Then, robust methods are proposed for mapping numeric features and computing their distance, these being able to tolerate the impact of the value difference in scale and diversification among features, and outliers introduced by intrusions. Empirical experiments with the KDD 1999 dataset showed that ACUAD can detect intrusions with relatively low false alarm rates compared with other approaches.

Key words: Unsupervised anomaly detection, Data mining, Intrusion detection, Network security
doi:10.1631/jzus.C0910625 **Document code:** A **CLC number:** TP393.08

1 Introduction

There are two kinds of intrusion detection system (IDS): the misuse intrusion detection system (MIDS) and the anomaly intrusion detection system (AIDS). MIDS models the behavior of known attacks, and compares the object's current behavior with these models. If there is a match, there is an attack. One main shortcoming of MIDS is that it can detect only known attacks. With the potential of detecting previously unknown attacks, AIDS is another important part of IDS and a necessary complement to MIDS.

Traditional AIDS, also called supervised AIDS, models normal behavior of objects, and compares current behavior with these models. If there is a significant deviation, there is an attack. Supervised AIDS needs to be trained with clean or labeled data. This greatly restricts its availability. In reality, there are neither clean nor labeled data. Classifying data manually is, however, very slow, expensive, and error-prone.

To reduce the cost of deploying AIDS and to extend the availability of anomaly detection, the study of a new anomaly detection method, unsupervised anomaly detection, is becoming of interest. Unsupervised anomaly detection can detect attacks without the need for clean or labeled training data.

The application of clustering to unsupervised anomaly detection (ACUAD) is studied in this paper. A key element for unsupervised anomaly detection with a clustering method is the definition of the distance function between data records. The main shortcomings of current distance functions are: (1) some functions can calculate only the distance of

* Project supported by the PhD Foundation of Engineering and Commerce College, South-Central University for Nationalities, China

numeric features or nominal features, not both; (2) some functions compute the distance for both numeric and nominal features, but in an unbalanced way; and, (3) distance functions have not considered the impacts of outliers introduced by attacks in unlabelled data. This paper deals with a distance framework for records with mixed numeric and nominal feature data. A heuristic method that computes the distance for nominal features is proposed, taking advantage of an important characteristic of nominal features, i.e., the probability distribution. Robust methods for mapping numeric features and computing their distance are proposed, these being able to tolerate the impact of the difference of feature values in scale, diversification, and outliers introduced by intrusions. Empirical experiments with the KDD 1999 dataset (http://kdd.ics.uci.edu/databases/kddcup99/task.html) showed that ACUAD can detect intrusions with relatively low false alarm rates compared with other detection methods.

2 Related works

Leung and Leckie (2005) studied several methods of clustering for unsupervised anomaly intrusion detection. Principal component analysis (PCA) for data reduction and fuzzy adaptive resonance theory (fuzzy ART) for the classifier were used for unsupervised anomaly intrusion detection (Ismail *et al.*, 2008). Eskin (2000) presented a statistical mixture model for unsupervised anomaly detection. Eskin *et al.* (2002) also proposed two feature maps for mapping system call traces and network connection records to a feature space. Three algorithms, cluster, *k*-nearest neighbor (*k*-NN), and support vector machine (SVM), were used for detecting anomaly points in the feature space. Cansado and Soto (2008) used Bayesian networks for unsupervised anomaly detection. Kwitt and Hofmann (2007) employed PCA to detect anomalies in the measurements of certain network traffic parameters.

3 Fixed-width clustering algorithm

There are two assumptions for unsupervised anomaly detection (Eskin *et al.*, 2002). The first assumption is that data records generated by normal activities are vastly outnumbered data records generated by attacks. The second assumption is that attack data records are qualitatively different form normal data records. If data records are mapped to a feature space, according to the second assumption, points of normal records and attack records will be in different areas of the feature space. With the first assumption, the areas of points of normal records will be dense, whereas the areas of points of attack records will be sparse. So, clustering algorithms can be used to determine the sparse areas in the feature space. Data records with points in these areas are attacks.

A fixed-width clustering algorithm partitions data records to clusters according to a distance threshold *w*, which is also called the 'cluster radius'.

The fixed-width clustering algorithm is as follows (Eskin *et al.*, 2002). The first point is the center of the first cluster. For every subsequent point, if its distance to an existing cluster center is less than the cluster width *w*, it is added to that cluster. Otherwise, it is the center of a new cluster. The computing complexity of the fixed-width clustering algorithm is $O(kN)$, where k is the number of clusters, and N is the number of data records.

4 Definition of the distance function

The criterion of clustering is the distance between data records and clusters. Distance between a data record and a cluster is the distance between the data record and the center record of the cluster, so the definition of the distance function between data records is critical.

Suppose data record set $E=\{\boldsymbol{E}_1, \boldsymbol{E}_2, \ldots, \boldsymbol{E}_N\}$. Every record in E is described by p numeric feature $(X_1, X_2, \ldots, X_p)$. $\boldsymbol{E}_i=(x_{i1}, x_{i2}, \ldots, x_{ip})$ and $\boldsymbol{E}_j=(x_{j1}, x_{j2}, \ldots, x_{jp})$ are two records of E. Then in the feature space with straightforward mapping, the distance between $\boldsymbol{E}_i$ and $\boldsymbol{E}_j$ is

$$d(\boldsymbol{E}_i, \boldsymbol{E}_j)=\sqrt{\sum_{k=1}^{p}(x_{ik}-x_{jk})^2}\ . \qquad (1)$$

Eq. (1) is the method of calculating the distance between two points in the Euclidean space, also called the 'Euclidean distance function'. But it cannot be used to calculate the distance between data records in unsupervised anomaly detection. First, Eq. (1) can deal only with records with numeric features. In unsupervised anomaly detection, data records (such as network connections) have both numeric features (such as bytes transferred and duration of connection) and nominal features (such as the application layer protocol, flag, etc.). Second, Eq. (1) does not consider the difference of values in scale, diversification among features, and outliers introduced by intrusions.

It is clear from the above discussion that the Euclidean distance function cannot be used directly in unsupervised anomaly detection. A distance function suitable for unsupervised anomaly detection should have the following characteristics. First, it should have a framework to calculate distances for both numeric features and nominal features, and do so in a balanced manner. Second, there should be a discriminative method that computes the distance for nominal features. Third, there is a need for robust methods for mapping numeric features to the feature space and calculating their distance while at the same time being able to tolerate the impact of the difference of feature values in scale, diversification, and outliers introduced by intrusions.

4.1 Distance framework for data records

Suppose data record set $E=\{\boldsymbol{E}_1, \boldsymbol{E}_2, \ldots, \boldsymbol{E}_N\}$. $(\boldsymbol{X}, \boldsymbol{Y})=(\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_p, \boldsymbol{Y}_1, \boldsymbol{Y}_2, \ldots, \boldsymbol{Y}_m)$ represents a data record with p numeric features and m nominal features. For a data record, numeric feature $\boldsymbol{X}_k$ ($k\in\{1, 2, \ldots, p\}$) has a real value. Every nominal feature $\boldsymbol{Y}_k$ ($k\in\{1, 2, \ldots, m\}$) has a domain of category values DOM($\boldsymbol{Y}_K$). Records $\boldsymbol{E}_i$ and $\boldsymbol{E}_j$ can be represented as

$$\boldsymbol{E}_i=(x_{i1}, x_{i2}, \cdots, x_{ip}, y_{i1}, y_{i2}, \cdots, y_{im}),$$
$$\boldsymbol{E}_j=(x_{j1}, x_{j2}, \cdots, x_{jp}, y_{j1}, y_{j2}, \cdots, y_{jm}).$$

Similar to the format of the Euclidean distance, the distance between data records $\boldsymbol{E}_i$ and $\boldsymbol{E}_j$ is defined as

$$d(\boldsymbol{E}_i, \boldsymbol{E}_j)=\sqrt{\sum_{k=1}^{p}\left\|x_{ik}-x_{jk}\right\|^2+\sum_{k=1}^{m}\left\|y_{ik}-y_{jk}\right\|^2}, \quad (2)$$

where $\sum_{k=1}^{p}\| x_{ik}-x_{jk}\|^2$ is the quadratic sum of distances of all numeric features, and $\sum_{k=1}^{m}\| y_{ik}-y_{jk}\|^2$ is the quadratic sum of distances of all nominal features.

Eq. (2) is a distance function framework between two data records with numeric and nominal feature mixed data. The distance of numeric and nominal features will be discussed in the following sections.

4.2 Distance of numeric features

With straightforward mapping, the contribution of numeric features to the squared distance between two records $\boldsymbol{E}_i$ and $\boldsymbol{E}_j$ is

$$\sum_{k=1}^{p}\left\|x_{ik}-x_{jk}\right\|^2=\sum_{k=1}^{p}(x_{ik}-x_{jk})^2. \quad (3)$$

In view of the difference of values in scale and diversification among attributes, with straightforward mapping, contributions of some features are significantly larger than those of other features. This will result in using features in an unbalanced way. To balance the impact of different features on the distance between records, a feature mapping method instead of straightforward mapping, should be used.

One possible mapping method is normalization. Finding the maximum value of a feature, all values of this feature are normalized to this maximum value. By doing so, all feature values will be mapped to interval [0, 1]. But normalization has drawbacks when used in unsupervised anomaly detection. If there is a large value introduced by mistake or attacks, most values will be mapped to a very narrow interval. In unsupervised anomaly detection, it is not infrequent to have large outliers. So normalization mapping is not suitable for unsupervised anomaly detection.

Another feature mapping method is ‘standardization’. In this method, for a feature, all values are will be normalized to the number of standard deviation away from the mean. Suppose dataset $E=\{\boldsymbol{E}_1, \boldsymbol{E}_2, \ldots, \boldsymbol{E}_N\}$ has N records. Values of feature $\boldsymbol{X}_k$ in E form a dataset $\{x_{1k}, x_{2k}, \ldots, x_{Nk}\}$. u_k and σ_k are the mean and standard deviation of $\boldsymbol{X}_k$, respectively. Then value x_{ik} ($i\in\{1, 2, \ldots, N\}$), after standardization, is

$$\frac{x_{ik}-u_k}{\sigma_k}. \quad (4)$$

For a dataset, the mean is a location estimator, and the standard deviation is a scatter estimator. The mean and standard deviation can be used to describe an approximately normal dataset, since 99.7% of data are within the distance of three times standard deviation from the mean. The mean and standard deviation are more robust than the extremum. Mapping a feature based on its mean and standard deviation is more reasonable than based on its extremum.

Although the mean and standard deviation can be good estimators for a normal dataset, they are not robust estimators for unsupervised anomaly detection because their values can change dramatically in the presence of outliers introduced by attacks.

The median is a more robust location estimator of a dataset than the mean. Sorting a dataset $\{x_{1k}, x_{2k}, \ldots, x_{Nk}\}$ from smallest to largest, the one in the middle is the median. Note that at least 50% of the observations in the dataset would have to be contaminants before the median would become arbitrarily large or small. Conversely, a single outlier introduced by attacks can significantly affect the mean.

Shorth, standing for the 'shortest half', is a scatter estimator more robust than the standard deviation. shorth is computed as follows (Knorr, 2002).

Step 1: Sort dataset $\{x_{1k}, x_{2k}, \ldots, x_{Nk}\}$ from smallest to largest, and obtain dataset $\{x_{(1)k}, x_{(2)k}, \ldots, x_{(N)k}\}$, where $x_{(1)k}$ is the smallest, and $x_{(N)k}$ is the largest.

Step 2: For j=1 to N/2, compute $D_j=x_{(j+N/2)k}-x_{(j)k}$.

Step 3: Select the minimum value from D_j ($j\in\{1, 2, \ldots, N/2\}$), D_m for example.

Step 4: $\text{shorth}_k=0.75\times D_m$.

Similarly, at least 50% of the observations in the dataset would have to be contaminants before shorth would become arbitrarily large or small. Thus, in unsupervised anomaly detection, shorth is a more robust scatter estimator than the standard deviation when there are outliers introduced by attacks.

Suppose the median and shorth of feature $\boldsymbol{X}_k$ are median_k and shorth_k, respectively. Then value x_{ik}, after standardization based on median and shorth, is

$$\frac{x_{ik}-\text{median}_k}{\text{shorth}_k}. \tag{5}$$

After being standardized with expression (5), the contribution of all numeric features to the squared distance between data records $\boldsymbol{E}_i$ and $\boldsymbol{E}_j$ is

$$\sum_{k=1}^{p}\left\|x_{ik}-x_{jk}\right\|^2=\sum_{k=1}^{p}\left(\frac{x_{ik}-x_{jk}}{\text{shorth}_k}\right)^2. \tag{6}$$

4.3 Distance of nominal features

In Eq. (2), the contribution of nominal features to the squared distance between records $\boldsymbol{E}_i$ and $\boldsymbol{E}_j$ is

$$\sum_{k=1}^{m}\left\|y_{ik}-y_{jk}\right\|^2. \tag{7}$$

According to the definition of a data record, each nominal feature $\boldsymbol{Y}_k$ ($k\in\{1, 2, \ldots, m\}$) has a category that belongs to DOM($\boldsymbol{Y}_k$). Assuming there are n_k categories in DOM($\boldsymbol{Y}_k$), i.e.,

$$\text{DOM}(\boldsymbol{Y}_k)=\{y_{k,1}, y_{k,2}, \ldots, y_{k,n_k}\}, \tag{8}$$

we can map the value of feature $\boldsymbol{Y}_k$ of a record to an n_k-dimensional space $\mathbb{R}^{n_k}$. If the category of feature $\boldsymbol{Y}_k$ of the record is $y_{k,j}$ ($j\in\{1, 2, \ldots, n_k\}$), then it has a coordinate of 1 in the jth dimension in space $\mathbb{R}^{n_k}$, all other coordinates being 0. For example, assuming the category of feature $\boldsymbol{Y}_k$ of record $\boldsymbol{E}_i$ is $y_{ik}=y_{k,2}$, its coordinate in space $\mathbb{R}^{n_k}$ is (0, 1, 0, ···, 0). Similarly, assuming the category of feature $\boldsymbol{Y}_k$ of record $\boldsymbol{E}_j$ is $y_{jk}=y_{k,2}$, its coordinate in space $\mathbb{R}^{n_k}$ is also (0, 1, 0, ···, 0). Then, the distance between y_{ik} and y_{jk} is

$$\begin{aligned}&\left\|y_{ik}-y_{jk}\right\|\\&=\sqrt{(0-0)^2+(1-1)^2+(0-0)^2+\ldots+(0-0)^2}=0.\end{aligned}$$

If $y_{jk}\neq y_{k,2}$, for example, $y_{jk}=y_{k,3}$, its coordinate in space $\mathbb{R}^{n_k}$ is (0, 0, 1, 0, ···, 0). Then the distance between y_{ik} and y_{jk} is

$$\begin{aligned}&\left\|y_{ik}-y_{jk}\right\|\\&=\sqrt{(0-0)^2+(1-0)^2+(0-1)^2+(0-0)^2+\ldots+(0-0)^2}\\&=\sqrt{2}.\end{aligned}$$

From the above discussion, we have

$$\left\|y_{ik}-y_{jk}\right\|^2=\begin{cases}0, & y_{ik}=y_{jk},\\ 2, & y_{ik}\neq y_{jk}.\end{cases} \tag{9}$$

Eq. (9) does not consider the relative frequency of categories of a nominal feature. Category distribution is an important characteristic for a nominal feature. Since for unsupervised anomaly detection, normal data records vastly outnumber attack records, categories of low frequency should be introduced by attack records, or conversely, categories of high frequency should be introduced by normal records. To partition normal records and attack records in the feature space, Eq. (9) should be multiplied by a coefficient α, which is related to the category distribution of a nominal feature.

The coefficient α should make two records with high frequency categories of nominal features closer, and two records with low frequency categories further. Suppose there are two record pairs $\{\boldsymbol{E}_i, \boldsymbol{E}_j\}$ and $\{\boldsymbol{E}_l, \boldsymbol{E}_m\}$. Their categories of nominal feature $\boldsymbol{Y}_k$ are y_{ik}, y_{jk}, y_{lk}, and y_{mk}. The numbers of occurrences of these categories in record set E are $n_{y_{ik}}, n_{y_{jk}}, n_{y_{lk}}$, and $n_{y_{mk}}$, respectively. Then,

$$n_{y_{ik}} + n_{y_{jk}} > n_{y_{lk}} + n_{y_{mk}} \Rightarrow \alpha_{ijk} < \alpha_{lmk}. \tag{10}$$

The times of occurrence of categories y_{ik} and y_{jk} in record set E is $n_{y_{ik}} + n_{y_{jk}}$, and the probability having these categories is $(n_{y_{ik}} + n_{y_{jk}})/N$, where N is the number of records in E. The distance between y_{ik} and y_{jk} should be an inverse ratio of their occurrence probability, so α is defined as

$$\alpha_{ijk} = \frac{N}{n_{y_{ik}} + n_{y_{jk}}}. \tag{11}$$

Combining Eqs. (9) and (11), the contribution of nominal features to the squared distance between records $\boldsymbol{E}_i$ and $\boldsymbol{E}_j$ is

$$\left\|y_{ik} - y_{jk}\right\|^2 = \begin{cases} 0, & y_{ik} = y_{jk}, \\ \dfrac{2N}{n_{y_{ik}} + n_{y_{jk}}}, & y_{ik} \neq y_{jk}. \end{cases} \tag{12}$$

Now we can analyze the relative scale of distances for numeric and nominal features.

For numeric features, the standardization method used is (x−median)/shorth. Since there are no existing methods for analyzing its value, we can analyze distance scale mapping with $(x-u)/\sigma$ instead. For random variables with a normal or approximately normal distribution, these two methods have similar value scales.

Suppose the values of numeric feature X obey a normal distribution. Then 68.3% of values satisfy $|x-u|<\sigma$, and 95.4% values conform to $|x-u|<2\sigma$. After standardization with $(x-u)/\sigma$, there are 68.3% of values in interval (−1, 1), and 95.4% of values in interval (−2, 2). Thus, there are 46.6% of distance for numeric features in interval (0, 2), and 91% of distance in interval (0, 4).

For a nominal feature, according to Eq. (12), when values are different, the distance is $\sqrt{2N/(n_{y_{ik}} + n_{y_{jk}})}$, which is a value larger than $\sqrt{2}$.

From the above discussion, we can see that numeric and nominal features have roughly the same distance scale.

5 Experimental results and analysis

The KDD (1999) dataset is a widely used benchmark for IDS evaluation. The dataset contains 4 898 431 network connection records. The proportion of attack records to normal ones in the dataset is very large. We generated the experimental dataset from the KDD data with the records whose feature ‘logged in’ is ‘1’ (connections logged in successfully). The resulting dataset contains totally 703 066 records, among which 3377 records are attacks.

Each record is described with 34 numeric features and 7 nominal features. Numeric features are duration, src_bytes, etc. Nominal features are protocol_type, service, etc. The KDD 1999 dataset has both normal data records and attacks. There are 24 attack types, falling into four main categories: DoS (denial of service), R2L (remote to local), U2R (user to root), and PROBE.

We used 7 numeric features and 2 nominal features among the whole 41 features for intrusion detection. The 7 numeric features are duration, src_bytes, dst_bytes, count, srv_count, dst_host_count, and dst_host_srv_count. The 2 nominal features are service and flag. Within the 9 features used, duration, src_bytes, dst_byte, service, and flag are basic features of network connections, and count, srv_count, dst_host_count, and dst_host_srv_count are traffic features computed using a 2-s time window.

Table 1 is the test results in Eskin *et al.* (2002) and of our method ACUAD. Eskin *et al.* (2002) presented three unsupervised detection algorithms: cluster, *k*-NN, and SVM. ACUAD was tested with clustering threshold *w*=450 and *w*=500, respectively. Fig. 1 is the ROC (receiver operating characteristic) curves of these algorithms. Since ACUAD with *w*=450 and *w*=500 had almost the same test results, they were overlapped with each other to form a single curve. Fig. 1 shows that ACUAD had higher detection rates and lower false alarm rates than the algorithms described in Eskin *et al.* (2002).

Table 1 Test results of the algorithms proposed in Eskin *et al.* (2002) and our proposed ACUAP*

Algorithm	Detection rate & false alarm rate (%)**
Cluster	(93, 10), (66, 2), (47, 1), (28, 0.5)
k-NN	(91, 8), (23, 6), (11, 4), (5, 2)
SVM	(98, 10), (91, 6), (67, 4), (5, 3)
ACUAD, *w*=450	(93, 6.9), (80.9, 3.3), (68.6, 0.9), (60, 0.8), (12.8, 0.6), (5, 0.4)
ACUAD, *w*=500	(94.6, 7.6), (82, 3.3), (68.6, 0.8), (62.2, 0.7), (15.5, 0.5)

* Data for Cluster, *k*-NN, and SVM are adopted from Eskin *et al.* (2002). ** In (*a*, *b*), *a* is the detection rate (%) and *b* is the corresponding false alarm rate (%). *k*-NN: *k*-nearest neighbour; SVM: support vector machine; ACUAD: application of clustering to unsupervised anomaly detection. *w* is the clustering threshold

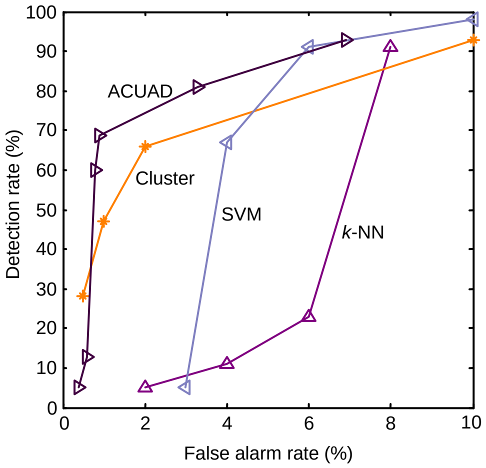

Fig. 1 Receiver operating characteristic (ROC) curves of the different algorithms

k-NN: *k*-nearest neighbour; SVM: support vector machine; ACUAD: application of clustering to unsupervised anomaly detection

Anomaly intrusion detection will be affected by the attack rate in the data as a whole. This is because one main assumption of anomaly intrusion detection is that data records generated by normal activities vastly outnumber data records generated by attacks. We still need to make sure ACUAD is robust with respect to the attack rate. Four sets of experiments were conducted with attack rates of 0.48%, 2.4%, 4.6%, and 8.8%. The test results are shown in Table 2.

Table 2 Test results of ACUAD with different attack rates (r_1–r_9)

Cluster width	Detection rate & false alarm rate (%)*								
	r_1=0.48%	r_2=2.4%	r_3=4.6%	r_4=8.8%	r_5=12.6%	r_6=16.2%	r_7=27.9%	r_8=43.6%	r_9=60.7%
w=450	(93, 6.9)	(94.1, 7.2)	(95.6, 8.5)	(91.7, 9.8)	(93, 9.5)	(81.5, 17.2)	(99.6, 21.3)	(99.3, 23.0)	(53.5, 22.3)
	(80.9, 3.3)	(82.2, 3.2)	(84.9, 4.1)	(68.6, 1.0)	(68.6, 1.0)	(77.9, 8.4)	(53.5, 21.3)	(52.9, 23.0)	(21.1, 15.5)
	(68.6, 0.9)	(68.4, 0.8)	(68.6, 0.9)	(22.7, 1.0)	(21.2, 0.8)	(32.7, 2.6)	(31.2, 17.4)	(31.2, 19.2)	(19.7, 15.2)
	(60, 0.8)	(21.8, 0.8)	(22.1, 0.8)	(20.2, 0.8)	(15.2, 0.7)	(23.4, 0.8)	(24, 1.9)	(15.5, 10.7)	(8.9, 6.7)
	(12.8, 0.6)	(16.1, 0.7)	(14.6, 0.8)	(14.9, 0.7)					
	(5, 0.4)	(6.3, 0.6)	(8.6, 0.8)	(9.2, 0.7)					
w=500	(94.6, 7.6)	(94.8, 7.9)	(95.6, 8.4)	(91.5, 5.1)	(92.9, 5.9)	(73.4, 15.8)	(99.6, 21.3)	(54.4, 21.8)	(53.5, 22.3)
	(82, 3.3)	(83.6, 3.5)	(84.9, 3.8)	(68.5, 0.9)	(68.6, 0.7)	(68.5, 0.8)	(54.4, 21.3)	(28.8, 16.4)	(18.6, 15.1)
	(68.6, 0.8)	(68.6, 0.9)	(68.6, 0.8)	(21.6, 0.8)	(21.5, 0.7)	(21.8, 0.8)	(47.1, 14.5)	(12.7, 9.5)	(7.8, 6.7)
	(62.2, 0.7)	(22, 0.9)	(21.3, 0.7)	(14.1, 0.7)	(10.6, 0.6)	(10.8, 0.7)	(14.5, 14.5)	(11.7, 11.7)	(3.7, 0.5)
	(15.5, 0.5)	(16.3, 0.8)	(17.9, 0.7)	(10.7, 0.7)					

* In (*a*, *b*), *a* is the detection rate (%) and *b* is the corresponding false alarm rate (%). ACUAD: application of clustering to unsupervised anomaly detection. *w* is the clustering threshold. For r_1–r_9, the numbers of attacks are all 3377, and the numbers of the total records are 703 066, 143 315, 73 346, 38 361, 26 700, 20 869, 12 123, 7750, and 5563, respectively

Table 2 shows that when the attack rate was not larger than 12.6%, the false alarm rate changed little as the attack rate changed. This means that ACUAD is robust enough with respect to the attack rate. At the same time, there were some circumstances in which the performance of ACUAD worsened. For example, when the attack rate increased from 12.6% to 16.2%, the detection rate decreased from 93% to 81.5%, and the false alarm rate increased from 9.5% to 17.2%. Also, when the attack rate increased from 43.6% to 60.7%, the detection rate decreased from 99.3% to 53.5%, although the false alarm rate changed little, with 23.0% and 22.3%, respectively. This is resulted from the characteristics of cluster algorithms. When the attack rate reaches a certain level, clusters with attacks cannot be differentiated from clusters of normal data.

6 Conclusions

We study ACUAD, an application of clustering to unsupervised anomaly detection. With ACUAD, data records are mapped to a feature space. Anomalies are detected by determining which points lie in the sparse regions of the feature space. Key elements for this method to be effective are feature mapping approaches and the definition of a distance function. We propose a unified normalized distance function framework for records with numeric and nominal feature mixed data, and use these two kinds of feature in a balanced manner. A heuristic method that computes the distance for nominal features is proposed. This method takes advantage of an important characteristic of nominal features, their probability distribution. Robust methods for mapping numeric features and computing their distance are proposed, and these can tolerate the impact of the difference of values in scale, diversification among features, and outliers introduced by intrusions. Empirical experiments with the KDD 1999 dataset showed that ACUAD can detect intrusions at relatively low false alarm rates compared with other approaches.

Future work includes on-line intrusion detection with clusters formed by ACUAD as the detection models, and methods that can find the clustering threshold automatically.

References

Cansado, A., Soto, A., 2008. Unsupervised anomaly detection in large databases using Bayesian networks. *Appl. Artif. Intell.*, **22**(4):309-330. [doi:10.1080/08839510801972801]

Eskin, E., 2000. Anomaly Detection over Noisy Data Using Learned Probability Distributions. Proc. Int. Conf. on Machine Learning, p.255-262. [doi:10.1109/ICCSA.2008.70]

Eskin, E., Arnold, A., Prerau, M., Portony, L., Stolfo, S., 2002. A Geometric Framework for Unsupervised Anomaly Detection: Detecting Intrusions in Unlabeled Data. *In*: Barbara, E., Jajodia, S. (Eds.), Applications of Data Mining in Computer Security. Kluwer Academic Publishers, Norwell, MA, USA, p.272.

Ismail, A.S.H., Abdullah, A.H., Bak, K.B.A., Nqudi, M.A., Dahlan, D., Chimphlee, W., 2008. A Novel Method for Unsupervised Anomaly Detection Using Unlabelled Data. Proc. Int. Conf. on Computational Sciences and Its Applications., p.252-260. [doi:10.1109/ICCSA.2008.70]

Knorr, E.M., 2002. Outliers and Data Mining: Finding Exceptions in Data. PhD Thesis, University of British Columbia, Canada, p.74.

Kwitt, R., Hofmann, U., 2007. Unsupervised Anomaly Detection in Network Traffic by Means of Robust PCA. Proc. Int. Multi-Conf. on Computing in the Global Information Technology, p.37-41. [doi:10.1109/ICCGI.2007.62]

Leung, K., Leckie, C., 2005. Unsupervised Anomaly Detection in Network Intrusion Detection Using Clusters. Proc. 28th Australasian Conf. on Computer Science, **102**:333-342.