Research on the Intrusion Detection Method Based on Differentiated Cluster Center Offset Measure

Yu Chen[1,*], *Hao* Chun[2], and *Qing* Nian[2]

[1]School of Information Science and Engineering, Lanzhou University, Lanzhou, China
[2]Lanzhou Municipal Public Security Bureau, Lanzhou, China

Abstract. In this paper, a kind of local outlier mining method based on differentiated cluster center offset measure is proposed through which the outlier degree of sample can be calculated by use of the normal behavior model constructed by normal data sample and the preset anomaly threshold value, and whether the testing sample belong to intrusion behavior can thus be determined. Furthermore, KDD99 data set is also utilized to test the said method, and the experimental results show that the method proposed in this paper possesses higher detection rate and lower false alarm rate.

Key words: Intrusion detection; Anomaly detection; Outlier mining; Cluster center offset

1 Overview

Essentially, the intrusion detection is a classification problem, and the to-be-detected host audit records or network traffic data can be classified as normal behavior or intrusion behavior [1]. Depending on different detection methods, the intrusion detections can be divided into two categories, i.e. anomaly detection and misuse detection. The local outlier mining method studied in this paper belongs to one of such anomaly detection methods.

The hypothesis of anomaly detection is that the intruder activity is anomalous to the activity of normal subject [2]. The activity profile of normal activity of the subject can be established according to this concept, and comparisons can be made between the activity status of current subject and the activity profile. When the statistical law is violated, the activity may be considered as an intrusion. The difficult problem for anomaly detection is how to establish the activity profile of normal activity and how to design the statistical algorithm, so that normal operations are not considered as intrusion or the real intrusion behavior are not neglected.

Outlier mining technology is very suitable for the completion of anomaly-based intrusion detection. Through the analysis of network data characteristics, two facts can be drawn as below. Firstly, there are significant differences between normal behavior and anomaly behavior. Secondly, in practical applications, the number of anomaly behavior is much lower than that of the normal behavior. In respect of the entire network behavior, intrusion behavior belongs to anomaly data in small numbers, and it can be processed by treating it as outliers in the dataset, which can better reflect the nature of such invasion. At the same time, in respect of other intrusion detection methods, the anomaly detection method based on outlier mining can identify those new categories of attack samples which have not yet appeared [3], which represents the advantage which is not possessed by other detection methods. Therefore, the problem of intrusion detection can be transformed into the outlier mining problem in the network behavior data set.

2 Outlier Mining Method Based on Differentiated Cluster Center Offset Measure

Although many outlier mining methods have been proposed in recently-published literature, yet only a few of them are applied to the network anomaly detection. One of the key reasons leading up to this situation lies in the relatively high computational complexity in many popular outlier mining methods. For example, the well-known density-based local outlier coefficient method [4] and the angle-based outlier mining method [5] belong to the above situation. Furthermore, the directly distance-based mining method is only suitable for the global outlier problem. In the literature [8], a Network Anomaly Detection method based on Change of the Position of cluster centers (NADCP) is proposed. Firstly, the normal samples are clustered, and then the outlier degree corresponding to each of sample is measured by the

* Corresponding author: chenyu15@lzu.edu.cn

position offset of cluster center and with the cluster center taken as reference sample. In this way, the algorithm complexity and limitation can be better solved. However, the fact that the data distribution in reality is usually is usually no uniform, and the sample density of each cluster may be different after clustering, which determines comparatively large difference in the outlier degree in different intra-cluster samples. For this reason, there is no way to measure the outlier degree with a unified set of measurement parameters. Therefore, a kind of outlier mining method based on differentiated cluster center offset measure (IDDM) is proposed in this paper.

2.1 Selection of reference sample

For the anomaly detection method based on outliers, it is a very critical problem to determine and select reference samples. Reference samples can be selected directly from the data set, or obtained by making some calculations on the data set. Due to the fact that samples belonging to one category in a data set are likely to possess similar properties, the seeking of reference samples will be measured in category in this paper, that is, reference samples are selected or extracted from the sample set with normal flow. By analyzing, we consider selecting the cluster center as reference sample because it essentially serves as a linear combination of all samples corresponding to such category, and contains some of the properties that generally possessed by the samples of such category. In the literature [6], the class center is taken as the reference sample of such category in the data set, and the distance between each of sample and the class center to which category it belongs is also used to measure the outlier degree of such sample. Those samples which are further away from the class center are considered as outlier samples. This tactics can determine outliers in a data set with low computational complexity, but such algorithm can easily lead to the false negative of outlier data, and the detection rate of intrusion data is not high. In this paper, the acquisition means of reference sample can be described like this, kmeans clustering algorithm [7] is applied to divide the normal flow set into K unconnected clusters, i.e. C_1, C_2, C_3,..., C_k, and cluster centers of each cluster, i.e. $c_1, c_2, c_3, \ldots, c_k$ can be then extracted.

2.2 Defining the indicator of outlier degree

The influence caused by an addition of one outlier sample is greater than that caused by an addition of one normal sample. When this influence is measured by relative distance, it shows that if an outlier sample is newly added, the relative distance between the current "new" class center and the previous "old" class center is larger. If a normal sample is newly added, the relative distance will be smaller. Based on this idea, the position change of the class center can be observed by the way of "adding a sample" (i.e. "copying" an existing sample in one data set) and further determine whether such sample is an outlier. To be specific, when applied to the network anomaly detection, it is required to take the class center from the normal sample set as reference sample, and determine whether it is an anomaly sample by observing the position change of class center caused after the addition of samples to be detected.

In this paper, the Euclidean distance is used as a distance function to measure the similarity between samples. The formula is shown as follows:

$$D(x_i, c_j) = sqrt(\sum (x_i - c_j)^2) \tag{1}$$

After K cluster centers, i.e. $c_1, c_2, c_3, \ldots, c_k$ are extracted from the normal sample set and taken as the reference sample, the next step is to define and calculate the outlier degree score d of each sample relative to the reference sample. Assuming that x_i is a sample in X, it is required to firstly calculate the respective Euclidean distance between x_i and K cluster centers, i.e. $c_1, c_2, c_3, \ldots, c_k$, extracted from X, find out the cluster center that has the nearest distance away from x_i (mark it as C_r and cluster center as c_r), and select it as the actual reference sample. Next, one target sample having the same characteristic value as x_i (also marked as x_i) is added, and the new cluster center of cluster C_r is marked as c_r' after the addition of such sample. In this case, c_r' can be calculated via the formula listed as below:

$$c_r^{'} = \sum_{x_j \in (c_r \cup x_i)} x_j \tag{2}$$

After the new cluster center c_r' is calculated, the Euclidean distance can be used to measure the offset d between c_r and c_r' before and after the addition of x_i.

$$d = D(c_r, c_r^{'}) \tag{3}$$

In this paper, the offset d is used as outlier degree score which can measure the outlier degree of all samples in x_i relative to C_r having the nearest distance away from it. Meanwhile, the outlier degree of x_i relative to X can also be reflected. Therefore, the larger the d value is, the higher possibility that x_i is an outlier sample will be.

2.3 Establishment of normal behavior model

In order to establish the normal behavior profile, it is also necessary to define an exception threshold value m after the outlier degree score corresponding to each of normal sample has been calculated, so as to determine which what kinds of normal samples can be used to establish the normal behavior profile of the user. The specific means of judgment is to compare the outlier degree score corresponding to each of samples in X and the size of anomaly threshold m, and treat those normal training samples having the outlier degree score less than the anomaly threshold value as samples used to establish user normal behavior profile. At the same time, those samples having too high outlier degree scores will not be used to establish the normal behavior profile, as it may

correspond to the noise data and affect the effect of model detection.

For a network anomaly detection method, different anomaly threshold values tend to have a very large impact on the detection performance. In general, if the anomaly threshold is set too high, the detection rate will be very low. On the other hand, if the anomaly threshold value is set too low, the false alarm rate will be very high. As we know, the data distribution in reality is usually not uniform, thus, the sample density of each intra-cluster is also different after the clustering, which determines the greater differences of outlier degree scores within different intra-cluster samples. For this reason, it is improper to use a unified anomaly threshold value to measure the outlier degree score of sample. The method adopted in this paper is to respectively set different anomaly threshold values $m_1, m_2, m_3, \ldots, m_k$ in K clusters based on the result of initial clustering. The concrete method is to respectively calculate the outlier degree scores corresponding to each sample in cluster $C_1, C_2, C_3, \ldots, C_k$, and then sort in descending order for the outlier degree scores d set in each of clusters. Next, set up a global percentage parameter β, take the βth outlier degree score d in every outlier degree score set and separately assign values to anomaly threshold values $m_1, m_2, m_3, \ldots, m_k$.

For a sample to be tested, i.e. x_i , the first step is to judge the cluster to which category it belongs according to the distance between it and the center of K clusters. Then, it is necessary to calculate the outlier degree score d of it relative to the cluster to which it belongs. Finally, comparison shall be made between such d and the anomaly threshold value m_i, to which cluster it belongs. If d is greater than m_i, such sample will be determined as anomaly data, and normal data conversely.

3 Implementation of Intrusion Detection Method Based on Differentiated Cluster Center Offset Measure

The outlier mining method based on cluster center offset measure is applied in the intrusion detection, and the specific intrusion detection process is shown in **Fig. 1**. To this end, the first step is to carry out preprocessing of testing samples, and realize data standardization. Next, the algorithm proposed in this paper is utilized to calculate the outlier degree of samples, and ultimately output the category labels of samples.

3.1 Data preprocessing

In this paper, KDD99 data set is adopted for experiment, and each entry of record of such data set comprises 41 feature attributes, including 38 numeric type features, and 3 nominal attribute type features, i.e. "protocol_type", "service", "flag", respectively. If the numerical computation can't be directly used in the nominal attributes, there is the need for mapping. The approach taken in this paper is to map the nominal attributes as binary attributes (0 and 1). For "service" attribute, as there are as many as 70 kinds of valuations, the characteristic dimension of data set will be greatly increased after the mapping, and the computational complexity may be increased and the classification effect may be affected. For this reason, it will be abandoned. For the two attributes of "protocol_type" and "flag", there are a total of 14 valuations respectively mapped as binary value attribute (0 and 1). Therefore, after the characteristic mapping, the original data set will be transformed from 41 feature attributes to 52 numeric type feature attributes.

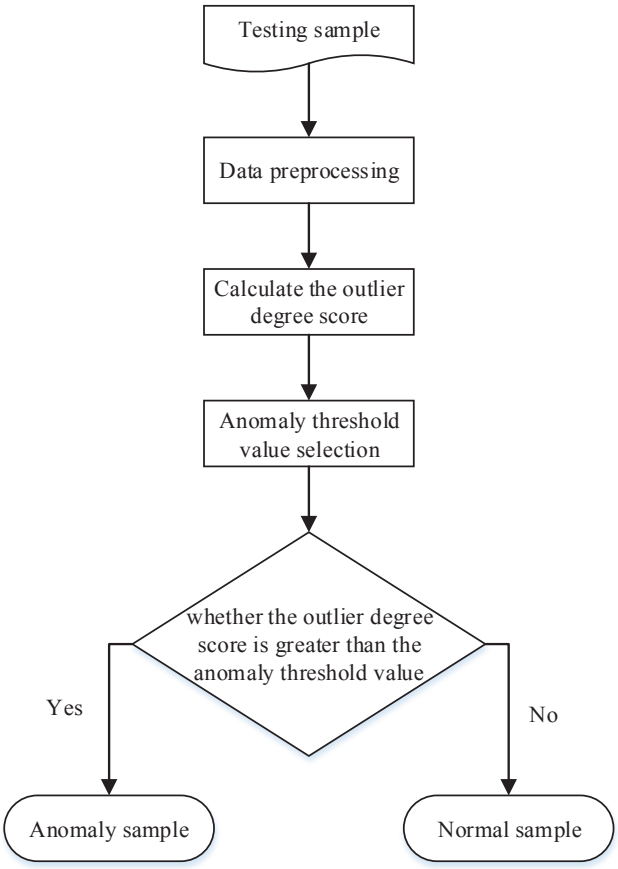

Fig. 1. intrusion detection process.

The data set after feature conversion is still not suitable for direct numerical calculation. As there are major differences in terms of the magnitude between every feature attribute, it is difficult to obtain good classification result. Thus, normalization adjustment of data is required. "Maximum-minimum" normalization method is adopted in this paper, enabling the end value to be mapped within the range of [0-1]. The transition function is listed as below:

$$x' = \frac{x - min}{max - min} \tag{1}$$

Among them, x stands for the original sample data, x' for the normalized sample data, max for the maximum value of sample data, and min for the minimum value of sample data.

3.2 Algorithm description

The outlier mining algorithm based on differentiated cluster center offset measure is described as follows:

Outlier mining algorithm based on differentiated cluster center offset measure

Input: X stands for the normal sample set used for training, K for the initial clustering number, m for anomaly threshold value, β for the percentage parameter, and x_t for the sample to be detected.

Output: Category label of the sample x_t to be detected.

1: Kmeans algorithm is applied to cluster the normal sample set X as $C_1, C_2, C_3, \ldots, C_k$, and output the cluster center of each cluster $c_1, c_2, c_3, \ldots, c_k$.
2: **for** each sample $x_i \in X$ **do**
3: Find its reference cluster center c_i from $c_1, c_2, c_3, \ldots, c_k$.
4: Calculate the outlier degree scores d_i corresponding to each x_i.
5: **end for**
6: The outlier degree scores d_i corresponding to each x_i shall be divided to $D_1, D_2, D_3, \ldots, D_k$ according to the cluster to which it belongs.
7: Sort the outlier degree scores d corresponding to $D_1, D_2, D_3, \ldots, D_k$ in descending order.
8: The percentage parameter β is applied to calculate anomaly threshold values of each cluster $m_1, m_2, m_3, \ldots, m_k$ corresponding to every outlier degree scores set D.
9: Find the reference cluster center c_t for x_t (the sample to be detected).
10: Calculate the outlier degree score d_t of x_t.
11: To compare between the outlier degree score d_t of x_t and the anomaly threshold value m_t of the cluster represented by the reference cluster center c_t.
12: **if** $d_t > m_t$ **do**
13: x_t will be determined as anomaly data.
14: **else**
15: x_t will be determined as normal data.
16: **end if**

4 Experiment and Result Analysis

KDD99 standard data set is selected in the experiment, and such data set comprises training set and testing set. All category labels are selected from "10% dataset" data as "normal" data sample, 97278 entries are used in training set for the construction of normal behavior model. There are more than 300000 lines of data samples in KDD99 standard test set ("corrected" file). Due to its overwhelming data size, only the front 50000 lines of data samples are selected as the testing set for this experiment. Different initial number of cluster K and anomaly threshold parameter β are applied to train these two parameters by use of experimental results. The testing performance of this experimental method for the testing set can be examined by use of three indicators, i.e. detection rate (probability of correct identification of invasion sample), false alarm rate (probability of normal sample identified as intrusion) and correct classification rate (probability of correct identification of sample).

In the training set, Kmeans algorithm is used to divide the dataset into clusters, the clustering number K is set as 10, and the anomaly threshold parameter β is respectively taken as 1%, 3%, 5%, 7% and 10%. The detection performance of the algorithm proposed in this paper is given in **Table 1.**

Table 1. Experimental results of detection performance at different β values (K = 10)

β (%)	detection rate (%)	false alarm rate (%)	correct classification rate (%)
1	94.45	0.55	95.09
3	94.57	1.88	95.02
5	94.62	3.49	94.86
7	94.67	4.73	94.74
10	94.69	7.13	94.46

As seen from **Table 1**, with the increase of β value the detection rate obtained by the test set continues to increase, but the false alarm rate also increases gradually. When β value is equal to 10%, the false alarm rate has obviously reached a very high level. The reason for this lie in the fact that along with the continuous decrease of anomaly threshold values, a portion of samples identified as normal samples when high anomaly threshold values are taken will be determined as anomaly threshold values when low anomaly threshold values are taken. However, these portions of samples contain not only a number of samples having aggressive real attributes, but also contain some normal samples having normal real attributes. Accordingly, both detection rate and false alarm rate have increased to some extent in this method. As revealed in the experimental results in Table 1, when β=1%, the detection performance is ideal. As revealed in further experiments under the condition of different clustering numbers K, when β value is less than 1%, the detection effect drops drastically.

Different initial clustering numbers K will also have a greater impact on the detection performance of test set. Specific experimental results will be given to different K values (4, 6, 8, 10, 14, 20), and the parameter of anomaly threshold value β will be taken as 1%.

Table 2. Experimental results of detection performance at different β values (β=0.01)

clustering number K	detection rate (%)	false alarm rate (%)	correct classification rate (%)
4	94.29	0.93	94.90
6	94.44	1.11	95.00
8	94.42	0.50	95.07
10	94.45	0.55	95.09
14	94.63	0.72	95.22
20	94.52	0.68	95.13

As revealed in the results of **Table 2**, when the values of clustering number *K* are taken within the range of 4-20, there are no significant fluctuations in the detection performance. It can be found by comparison that the detection performance can achieve a better effect if when the value of *K* is taken close to 8-10.

Table 3. Comparative results of detection performance obtained by other anomaly detection methods.

method	detection rate (%)	false alarm rate (%)	correct classification rate (%)
PSO-KM[9]	86	2.8	-
SOM[10]	91.5	14.5	-
CSI-KNN[11]	91.4	2.6	92.5
KSVM[12]	88.71	-	-
NADCP[8]	91.48	2.65	92.62
IDDM	94.45	0.55	95.09

Finally, the algorithm proposed in this paper is compared with several anomaly detection algorithms of the same category, and it is concluded that the algorithm proposed in this paper has obvious advantages in terms of detection performance.

5 Conclusion

In this paper, a kind of outlier mining method based on differentiated cluster center offset measure is proposed and also applied in the network intrusion detection. As revealed in the experimental results in KDD99 dataset, this method can help obtain higher detection rate and lower false alarm rate, and achieved better detection results when compared with similar methods. The next step is to combine the misuse detection means, further improve the detection effect, and apply this method to the network real-time intrusion detection.

References

1. A.A. Aburomman, M.B.I. Reaz. A survey of intrusion detection systems based on ensemble and hybrid classifiers. Computers & Security, **65**, (2017), 135-152.
2. M.H. Bhuyan, D.K. Bhattacharyya, J.K. Kalita. Network Anomaly Detection: Methods, Systems and Tools. IEEE Communications Surveys & Tutorials, (2013), 1-34.
3. R. Sonawane, T. Tajane, P. Chavan et al. Anomaly based intrusion detection network system. Software Engineering and Technology, **8(3)**, (2016), 66–69.
4. M.M. Breunig. H.P. Kriegel, R.T. Ng. LOF: identifying density-based local outliers. ACM, **29(2)**, (2000), 93-104.
5. H.P. Kriegel, M.S. Hubert, A. Zimek. Angle-based outlier detection in high-dimensional data. Acm Sigkdd International Conference on Knowledge Discovery & Data Mining, (2008), 444-452.
6. C.F. Tsai, K.C. Cheng. Simple instance selection for bankruptcy prediction. Knowledge-Based Systems, **27(3)**, (2012), 333-342.
7. T. Liang. Research on Intrusion Detection Technology Based on Clustering Analysis. Chongqing University, (2010).
8. C. Guo. Research on the Key Technology of Network Intrusion Detection Based on Data Mining [Ph.D. Dissertation]. Beijing: Beijing University of Posts and Telecommunications, (2014).
9. L. Xiao, Z. Shao, G. Liu. K-means Algorithm Based on Particle Swarm Optimization Algorithm for Anomaly Intrusion Detection. World Congress on Intelligent Control & Automation, **2**, (2006), 5854-5858.
10. H.G. Kayacik, A.N. Zincir-Heywood, M.I. Heywood. A hierarchical SOM-based intrusion detection system. Engineering Applications of Artificial Intelligence, **20(4)**, (2007), 439-451.
11. L. Kuang, M. Zulkernine. An anomaly intrusion detection method using the CSI-KNN algorithm. Acm Symposium on Applied Computing, (2008), 921-926.
12. U. Ravale, N. Marathe, P. Padiya. Feature Selection Based Hybrid Anomaly Intrusion Detection System Using K Means and RBF Kernel Function. Procedia Computer Science, **45(39)**, (2015), 428-435.