Network Traffic Anomaly Detection Based on Incremental Possibilistic Clustering Algorithm

YANG Tian-yi and LIU Shi-yue and LIU Jun-yi

School of Automation, Beijing Institute of Technology, Beijing 100081, P.R.China
E-mail: xyyh@bit.edu.cn

Abstract. This paper proposed a Mahalanobis distance based Incremental Possibilistic Clustering (IPC) algorithm to detect abnormal flow. Firstly, the attributes of network flow is extracted by damped incremental statistics. Then the model of normal traffic will be generated by IPC algorithm. To extract the model of high-dimensional data without pre-known number of cluster centers, the algorithm gradually choose outliers as new clustering centers and merges the overlapping clustering centers. Finally, the data that doesn’t belong to any normal model is regarded as abnormal data. By using the Mahalanobis distance instead of the traditional Euclidean distance, the defect that the possibilistic clustering tends to find the features of hypersphere is solved. The experiments show that this method can distinguish normal flow and abnormal flow effectively and reaches the detection rate of 98%.

1. Introduction
Due to the gradual development of internet technology and the popularity of internet of thing(IoT) devices, network security has become a crucial issue in cyberspace. As early as 2014, there were more than 40 million security incidents on the internet, resulting in significant economic losses and leakage of personal privacy [1]. With the accessing of IoT devices, including mobile phones, IP camera and other smart sensor, network traffic has become more and more complex. Although users’ network reliance is increasing continued, users still lack awareness of network security [2]. In this case, a kind of high efficiency and low consumption detection method is required for IoT devices to detect network intrusion.

This paper proposes a Mahalanobis distance based Incremental Possibilistic Clustering (IPC) algorithm for anomaly detection. The attributes are extracted by damped incremental statistics instead of sliding window to reduce memory consumption. The IPC gradually detects outliers as new clustering centers, and uses possibility clustering to adjust and merge cluster centers. Finally the data that doesn’t belong to any feature is regarded as abnormal data.

2. Related works
The anomaly detection model usually identify the attack by generate a detection model from normal traffic or the traffic combined with small percentage of abnormal packets. The detection model will divide the unknown network packets into different categories. The packet which has the characteristics of attack traffic will be assigned to the category of abnormal traffic or not fit into any category. Many researchers proposed unique detection model based on statistics, machine learning, data mining, information theory or deep learning to identify the
unknown attack traffic more accurately and efficiently. A survey [1] presents an in-depth analysis detection techniques. It proposed a generic framework for network anomaly detection, and list some detection model based on classification, statistics, information theory and clustering.

Machine learning is the most popular method of anomaly detection. Damopoulos et using four traditional machine-learning algorithms (Bayesian networks, radial basis function, K-nearest neighbors and random forest) [3] and reported that the detection rate has reached 99.8%. The deep learning is the newest favorite approach. Niyaz et designed a detection model by self-taught learning [4], Meidan et use deep auto-encoders to detect anomalous network traffic. The auto-encoders encode the packet and decode it, while only the normal packet will be restored correctly [5].

Clustering, as a common data mining method, is also suitable for anomaly detection. The paper [6] use K-means, k-medoid, EM clustering and KNN algorithm to detect unknown attack. The result shows that their detection accuracies is high while the anomaly detection module produces high false positive rate. Xie et designed an anomaly detection system based on fuzzy c-means [7], but this methods can’t deal with high-dimensional data, and can’t find the nonconvex structure datasets.

3. Naive clustering algorithm
K-means is a common Clustering algorithm which makes members in a same class have the minimum dispersion via reassigns class members in order to obtain the best clustering results. In K-means clustering, each sample belongs to the nearest cluster center. This method is time-saving and has rapid convergence rate, but the clustering result is not very satisfactory[8]. Based on Zadeh’s fuzzy theory, Bezdek J C proposed fuzzy-C-means clustering algorithm(FCM) [9]. This method extends the membership degree of k-means clustering from 0,1 to the interval $[0,1]$, which makes the sample affect all clusters.

The following index is minimized in FCM:

$$J(U, V) = \sum_{i=1}^{c} \sum_{j=1}^{n} u_{ij}^m d_{ij}^2 + \sum_{i=1}^{c} \sum_{j=1}^{n} \eta_i (1 - u_{ij})^m \tag{1}$$

where $n$ is the total number of samples, $c$ is the number of clusters, $v_i$ and $x_j$ is the $i^{th}$ and $j^{th}$ vector of data and cluster center. And $d$ is the distance from $x_j$ to $v_i$, $u$ is the membership from $x_j$ to $v_i$, $m$ is a default value of FCM, which will usually be 1.5 or 2.

However, FCM also has several obvious drawbacks such as sensitive to noise and need to pre-determine the number of cluster centers[10]. The most severe problem is that FCM can’t find the unknown because every data point must belong to one cluster obtained from normal traffic[11]. In view of the above question, the paper proposes using possibilistic clustering instead. By relaxing the limit that the sum of membership degree must be one, we get possibility value from all samples. Then the objective function is defined as:

$$J_{FCM}(U, V) = \sum_{i=1}^{c} \sum_{j=1}^{n} u_{ij}^m d_{ij}^2 + \sum_{i=1}^{c} \eta_i \sum_{j=1}^{n} (1 - u_{ij})^m \tag{2}$$

where $U$ is the possibility partition matrix, $u_{ij}$ means the possibility value that $x_j$ belongs to $v_i$. To avoid getting an infinity cluster center, the function adds $\sum_{i=1}^{c} \eta_i \sum_{j=1}^{n} (1 - u_{ij})^m$ as penalty function. $\eta_i$ is the pre-determined value which influence the clustering effect directly. The well-known value of $\eta_i$ is $K \sum_{i=1}^{c} \sum_{j=1}^{n} u_{ij}^m d_{ij}^2$ where $K > 0$ and usually is 1.
4. Anomaly detection based on IPC

In this section, we introduce the whole progress to detect abnormal network traffic. This section first introduces the method of incremental damping statistics, which can reduce the memory overhead of feature extraction. Then the unsupervised possibilities clustering method based on Mahalanobis distance is introduced. Finally, according to this method, an incremental possibilities clustering method is proposed to detect abnormal data.

4.1. Damped incremental statistics

Feature extraction is the first key problem in pattern recognition system. Generally, the behavior characteristic of network flow can be summarized as five fictions: source IP, destination IP, the size of packets, the sum number of packets and the time interval between each packet. There are three problems with extracting those features. Firstly, the packet arrival rate can be very high; secondly, there may be many different conversations at same time; thirdly, packets from different channels may be closely related [12, 13]. It is an ordinary method to maintain a window of packet from each conversation but this method will use a lot of memory and cannot be used on router.

Therefore, we use a framework named Damped Incremental Statistics which is proposed by Yisroel, Mirsky et. This framework uses incremental statistic maintained [14] over a damped window, makes the framework has a $O(1)$ complexity.

Suppose $S = \{x_1, x_2, \cdots x_n\}$ is an unbounded stream, where $x$ is the attribute of a packet (e.g. the size of packet). The mean, variance and standard deviation of $S$ can be updated incrementally by maintaining the tuple $I_S = (N, L_S, S_S)$, where $N$, $L_S$, and $S_S$ are the number, linear sum, and squared sum of the data stream. When a new data $x$ is caught, $I_S$ will update to $(N + 1, L_S + x, S_S + x^2)$.

In order to extract the current status of a data stream, old data must be thrown. The naive approach is to maintain a sliding window. While incremental statistics adds a decay function:

$$d_{\lambda}(t) = e^{-\lambda t}$$

where $\lambda \cdot t > 0$ is a default value and $t$ is the time elapsed since the last observation from stream.

So the window of the framework adds a variable $T_L$ which is the time last packet was caught. The window of damped incremental statistics will be $I_{Sd} = (n, L_S, S_S, T_L)$. The way to update the $I_{Sd}$ is shown below[5]:

i) Compute decay factor $\gamma = d_{\lambda} \cdot (T_{\text{now}} - T_L)$

ii) Process decay $I_{Sd} = (\gamma n, \gamma L_S, \gamma S_S, T_L)$

iii) Insert new value $I_{Sd} = (n + 1, L_S + x, S_S + x^2, T_{\text{now}})$

iv) For variables that require two-dimensional statistics at the same time(for both income and outcome packets), compute $SR_{ij} = \gamma SR_{ij} + r_i r_j$

v) Return value

The statistics which can be computed from the damped incremental framework is shown in Table 1. A total of 23 features can be extracted from a single time window (see Table 2). In this paper we extracts the same set of features from a total of five time windows: $0.1s, 0.5s, 1.5s, 10s$ and $1min$ ($\lambda = 5, 3, 1, 0.1, 0.01$), thus totaling 115 features.
Table 1. Summary of the incremental statistics which can be computed from $S_i$ and $S_j$

| Type   | Notation          | Calculation                                      |
|--------|-------------------|--------------------------------------------------|
|        | weight($w$)       | $w$                                              |
| 1D     | mean($\mu$)       | $L_S/w$                                          |
|        | variance($\sigma$) | $\sqrt{S_S/w - (L_S/w)^2}$                      |
| 2D     | size($O$)         | $\sqrt{\mu_i^2 + \mu_j^2}$                     |
|        | radius($R$)       | $\sqrt{\sigma_i^4 + \sigma_j^4}$               |
|        | covariance($Cov_{ij}$) | $\frac{SR_{ij}}{w_i+w_j}$                |
|        | correlation coefficient($P_{ij}$) | $\frac{Cov_{ij}}{\sigma_i\sigma_j}$             |

Table 2. The features extracted from each time window

| Value                          | Features               | source                              |
|-------------------------------|------------------------|-------------------------------------|
| Size of outbound packets      | $\mu, \sigma$         | Src_ip, Mac-ip, Channel, Socket     |
| Number of packet              | $w$                    | Src_ip, Mac-ip, Channel, Socket     |
| Packet jitter                 | $w, \mu, \sigma$      | Channel                             |
| Size of both in/outbound packets | $O, R, Cov_{ij}, P_{ij}$ | Channel                             |

4.2. Incremental possibilistic clustering algorithm based on Mahalanobis distance

Under normal circumstance, the value of each attribute will stay in the boundary. When the exception, usually network attack, occurs, the value within each of the traffic features will surpass the range of threshold value. The question is, although we get a method which can improve the feature recognition performance, the clusters of normal traffic are not centralized and the shape of data may be an ultra-high dimensional irregular polyhedron. So we use an Incremental possibilistic clustering to get clusters of normal packets. Then we determine whether other packets are normal traffic. If a packet cannot satisfy any cluster extracted from normal traffic, the anomaly will be considered to happen.

4.2.1. Mahalanobis distance. The Mahalanobis distance is a measure of the distance between a point $P$ and distribution $D$, introduced by P. C. Mahalanobis in 1936. It is the distance of the test point from the center of mass divided by the width of the ellipsoid in the direction of the point. Compared with Euclidean distance, it considers the relevance between each dimension and is not affected by the boundary of dataset[15]. It is obviously that the 115 features which are extracted from packet are highly correlated. Using Mahalanobis distance can make the clusters more realistic and reliable. This paper needs to compute the distance between each data to each cluster. While the shape of clusters and data can be considered as same, the function based on Mahalanobis distance is devised as:

$$d_{ij} = (x_j - v_i)^T \Sigma^{-1} (x_j - v_i)^T$$  (4)

where $x_j$ is the data in the dataset $X = x_1, x_2 \cdots x_n$, $v_i$ is one of the clusters, $\Sigma$ is the covariance matrix of $X$. While the covariance matrix may be a singular matrix, the Mahalanobis distance cannot be calculated by the inverse matrix for these datasets. We carry out the pseudo inverse matrix instead by matrix factorization.

4.2.2. Unsupervised possibilistic clustering algorithm. By using Partition coefficient (PC) and Partition entropy (PE) as penalty function, the penalty function can be changed to
\[ \frac{\beta}{m^2} \sum_{i=1}^{c} \sum_{j=1}^{n} \left( u_{ij}^{m} \log u_{ij}^{m} - u_{ij}^{n} \right) \] where \( \beta \) is the co-variance of dataset[16]. In PCM based on Mahalanobis distance, it’s recommended to adjust the \( \beta \) to \( \beta = \sum_{i=1}^{n} d(x_i, \bar{x})/n \). Therefore, the result of clustering will be more accordant with division indicators.

Finally, the objective function of possibilistic clustering is defined as:

\[ J_{PCM}(U, V) = \sum_{i=1}^{c} \sum_{j=1}^{n} u_{ij}^{m} d_{ij}^{2} + \frac{\beta}{m^2} \sum_{i=1}^{c} \sum_{j=1}^{n} \left( u_{ij}^{m} \log u_{ij}^{m} - u_{ij}^{n} \right) \] (5)

In order to minimize the objective function, the recursion formula will be:

\[ u_{ij} = e^{-m \sqrt{\beta} d_{ij}/\beta} \] (6)

\[ v_i = \frac{\sum_{j=1}^{n} u_{ij}^{m} x_j}{\sum_{j=1}^{n} u_{ij}^{m}} \] (7)

In (6), \( u_{ij} \) is determined by the distance between the cluster center to the sample. The range of \( u_{ij} \) is 0 to 1. This calculation method relaxes the restriction that the sum of membership degrees is 1, which means each cluster is calculated independently. Therefore, the true minimum number of objective function will be reached if and only if every cluster center comes together. Compared with FCM, PCM needs appropriate initial values, or the result will be really unsatisfied.

4.3. Incremental possibilistic clustering

In PCM, to get the correct clustering result of dataset, the number of cluster center must be more than the right number, and initial cluster center must be with enough dispersivity. Generally, it’s the defect of PCM. But for network traffic whose cluster number, shape and size are unpredictable, We design an Incremental possibilistic clustering algorithm (IPC) to get correct classifications, using PCM with excess number of initial cluster center which can make clustering accurately.

4.3.1. IPC principle. Suppose there is training set \( X = \{x_1, x_2, \cdots, x_n\} \) and clustering centers \( V = \{v_1, v_2, \cdots, v_c\} \). Then calculates the degree matrix of membership \( U \) and set a threshold \( u_{min} \). When the membership between a data point to any cluster center is higher than the threshold, the point will be marked as a member of these clusters. Moreover, if the point cannot classified into any category, this point will be marked as outlier. This paper use normal flow to get the clusters and use these clusters with threshold to detect abnormal packets.

In order to find the cluster center and the threshold of normal packets, IPC use the iterative PCM. Firstly, use FCM to get the initial cluster centers \( V = \{v_1, v_2, \cdots, v_c\} \) with an initial number \( c \). Then put these cluster centers into the PCM. The extra centers from FCM will be gathered and some outliers will be appear. The outliers here may be the truly outliers of the dataset, or the point of undetected aggregates. Therefore, IPC put these outliers into the cluster centers’ set to get a new set \( \newset = \{v_1, \cdots, v_c, x_{i_1}, \cdots, x_{i_m}\} \) and update the cluster number \( c \) into \( c + m \). Finally, repeat the above actions for several times, the missing cluster centers will be retained, and fake centers from truly outliers will move to true centers.

4.3.2. Realization of algorithm. Suppose there are training feature data \( X = \{x_1, x_2, \cdots, x_n\} \) extracted from normal network traffic. The steps for training are as follows.

a) Set initial value.

b) Using FCM to provide initial cluster centers:

c) Using PCM to cluster again and get the membership limit \( u_{min} \):
i) Calculate the distance between x and v.
ii) Update the membership matrix by using (6).
iii) Update the clustering centers V by using (7).
iv) Repeat steps (i) to (iii) until the step reach the limit or V is stabilized.

d) Calculate the max membership degree of each data $u_{max}$.
e) Choose the lowest 3% point of $u_{max}$ as $u_{min}$ or set default $u_{min}$.
f) Add 1/10 of the data whose $u_{max}$ is less than $u_{min}$ to cluster centers V.
g) Re-compute the PCM again.
h) Repeat previous step for several times until the step reach the limit or every outliers has been added to V.
i) Merging Cluster Centers to get the final detection model.

5. Experiments and result Analysis

This paper uses three sets of data to compare with FCM algorithm. The first is iris datasets to qualitative analysis the influence of initial cluster centers. The second dataset is a self-made dataset to analysis the ability of finding arbitrary shape clustering. Finally the data from network flow is used to test the effect on anomaly detection.

5.1. Qualitative analysis

The first experiment used iris dataset. The parameter m is set to 1.5 and the number of initial clusters is the independent variables. Figure 1 shows that when cluster center is equal to the real cluster number, both FCM and IPC can perform well. With the increasing number of initial clusters, Figure 2 shows that the wrong clustering centers appeared by using FCM while the IPC still shows a Reliable result.

Figure 1. Iris dataset and the clustering result of FCM(red star) and IPC(yellow star) with 2 cluster center.

Figure 2. Iris dataset and the clustering result of FCM(red star) and IPC(green star) with 3 initial cluster center.

The second experiment used a self-made two-dimensional datasets that data points are random distributed on a ring ($r \in [40, 50]$). The number of initial cluster center is set to 10 and m=1.5. After clustering (Figure 3), the cluster center from FCM (red circle) cannot satisfied with the real data. The IPC detect 98 cluster centers and these centers reflect the true shape of data.
5.2. IPC application to real datasets

The experiment used the dataset ISCX-Botnet 2014 created by Beigi E B et al.[17]. The dataset are composed of network flow from honeypot or through infecting computers with a given bot binary in a controlled environment. The dataset is including the abnormal packets and normal packets.

Firstly, we used damped incremental statistics to extract the attributes of normal network flow. 30 thousand of continued normal packets were used as training dataset. 10 thousand of packets combined with 47 normal packets and 9953 abnormal packets were used as testing dataset. This paper used MATLAB to run the IPC. Initial number of cluster center was set to 5. The parameter $m$ was set to 1.5 and $\beta$ was set to 180. Ceiling of the total steps was set to 100,000 and each turn of PCM cannot be more than 1/10 of the total steps. The FCM and PCM will stop until the movement of each point is less than $1 \times 10^{-4}$ and the whole IPC will stop when all outliers have been chosen as the cluster center or the step reach the limit. The threshold to identify the outliers is set to the 3% lowest membership after the first turn of PCM.

| Type          | Total Number | Detected | Precision |
|---------------|--------------|----------|-----------|
| training set  | 40000        | 382      | 99.0%     |
| abnormal packet | 9953       | 9837     | 98.8%     |
| normal packet | 10000        | 86       | 0.86%     |

After training, the IPC finally detect 141 cluster centers while FCM just find 5 overlapping clustering centers. The threshold to detect anomaly traffic $u_{\text{min}} = 1.9334 \times 10^{-5}$. In order to evaluate the effect of IPC in anomaly detection, we calculate the precision and detection rate for the test. The result is shown in the Table 3.

From the result, we can see that model can detect 98.8% abnormal packets, while the error rate reached 1%. The algorithm has low precision when detect the abnormal packets from mixed network traffic. In this paper, IPC based on Mahalanobis distance and damped incremental
statistics can learn from normal packets and warn the researchers and internet users of coming network attack.

6. Conclusion
This paper introduces an anomaly detection method - incremental possibilities clustering. The method extract multi-dimensional features of network flow through damped incremental statistics that reduced memory overhead. Then using Mahalanobis distance to deal with the problem in dimension. The core algorithm can reduce the constraints of parameters and make it possible to deal with high-dimension data.

The experiment shows that IPC can find the pattern of training dataset and identify the abnormal data. But it means that the training dataset must be broad enough. In addition, the rule of merge clusters is the distance between each cluster centers. Density-based clustering algorithms can be added to IPC to get better clustering result.

References
[1] Mohiuddin Ahmed, Abdun Naser Mahmood, and Jiankun Hu. A survey of network anomaly detection techniques. Journal of Network and Computer Applications, 60:19–31, 2016.
[2] Eleonora Borgia. The internet of things vision: Key features, applications and open issues. Computer Communications, 54:1–31, 2014.
[3] Dimitrios Damopoulos, Sofia A Menesidou, Georgios Kambourakis, Maria Papadaki, Nathan Clarke, and Stefanos Gritzalis. Evaluation of anomaly-based ids for mobile devices using machine learning classifiers. Security and Communication Networks, 5(1):3–14, 2012.
[4] Ahmad Javaid, Quamar Niyaz, Weiqing Sun, and Mansoor Alam. A deep learning approach for network intrusion detection system. In Proceedings of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies (formerly BIONETICS), pages 21–26. ICST (Institute for Computer Sciences, Social-Informatics and . . ., 2016.
[5] Yair Meidan, Michael Bohadana, Yael Mathov, Yisroel Mirsky, Asaf Shabtai, Dominik Breitenbacher, and Yuval Elovici. N-bait—network-based detection of botnet attacks using deep autoencoders. IEEE Pervasive Computing, 17(3):12–22, 2018.
[6] Iwan Syarif, Adam Prugel-Bennett, and Gary Wills. Unsupervised clustering approach for network anomaly detection. In International Conference on Networked Digital Technologies, pages 135–145. Springer, 2012.
[7] Linquan Xie, Ying Wang, Liping Chen, and Guangxue Yue. An anomaly detection method based on fuzzy c-means clustering algorithm. In Second International Symposium on Networking and Network Security (ISNNS, 10) Jinggangshan, PR China, pages 2–4. Citeseer, 2010.
[8] Yang Yan Xu Xiaodong and Li Gang. Network traffic anomaly detection based on k-means clustering. Wireless Communication Technology, 22(4):21–26, 2013.
[9] S Askari, N Montazerin, and MH Fazel Zarandi. Generalized possibilistic fuzzy c-means with novel cluster validity indices for clustering noisy data. Applied Soft Computing, 53:202–283, 2017.
[10] Liu Pengcheng Tang Chenghua and Tang shensheng. Anomaly instution behavior detection based on fuzzy clustering and features selection. Journal of Computer Research and Development, 52(3):718–728, 2015.
[11] Wang Lina Luo Min and Zhang Huanguo. An unsupervised clustering-based intrusion detection method. ACTA Electronica Sinica, 31(11):1713–1716, 2003.
[12] K Munivara Prasad, A Rama Mohan Reddy, and K Venugopal Rao. Dos and ddos attacks: defense, detection and traceback mechanisms—a survey. Global Journal of Computer Science and Technology, 2014.
[13] Guo XJ, Zhou AP, and cheng G. Highspeed network traffic measurement method. Journal of Software, 25(1):135–153, 2014.
[14] Yisroel Mirsky, Tomer Doitschinman, Yuval Elovici, and Asaf Shabtai. Kitsune: an ensemble of autoencoders for online network intrusion detection. arXiv preprint arXiv:1802.09089, 2018.
[15] Zu zhiwen and Li Qin. Km-fcm: A fuzzy clustering optimization algorithm based on mahalanobis distance. journal of Hebei University of Science and Technology, 39(2):159–165, 2018.
[16] Zuo Chun.Cheng Hu Yating and Qu Fuheng. Multi-scale possibilistic clustering algorithm. Journal of Changchun University of Science and Technology(Natural Science Edition), 33(4):124–127, 2010.
[17] Elaheh Biglar Beigi, Hossein Hadian Jazi, Natalia Stakhanova, and Ali A Ghorbani. Towards effective feature selection in machine learning-based botnet detection approaches. In 2014 IEEE Conference on Communications and Network Security, pages 247–255. IEEE, 2014.