Anomaly Detection for Network Flow Using Immune Network and Density Peak

Yuanquan Shi[1,2] and Hong Shen[2]
(Corresponding author: Yuanquan Shi)

School of Computer Science and Engineering, Huaihua University[1]
Jinhai Road, Huaihua 418008, China
School of Computer Science, The University of Adelaide[2]
North Terrace, SA 5005, Australia
(Email: syuanquan@163.com)

(Received Oct. 29, 2018; Revised and Accepted May 17, 2019; First Online June 15, 2019)

Abstract

To identify effectively unknown malicious attack behaviors from massive network flows in Internet environment, an Anomaly Detection approach for network flow using Artificial Immune network and Density peak (ADAID) is proposed in this paper. In ADAID, we present an unsupervised clustering algorithm aiNet_DP combining artificial immune network (aiNet) and the clustering algorithm based on density peaks (CDP), where aiNet denotes a coarse-grained clustering algorithm to extract abstract internal images of network flows, CDP denotes a fine-grained clustering algorithm to obtain more precise cluster number and cluster centroids according to the clustering results of aiNet. The clustering labeling algorithm (CLA) and the flow anomaly detection algorithm (FAD) are introduced in ADAID to detect malicious attack behaviors of network flows, where CLA is used for labeling each cluster whether is malicious or not, and the labeled cluster is viewed as detector to identify anomaly network flows by using FAD. To evaluate the effectiveness of ADAID, the ISCX 2012 IDS dataset is used for simulating experiments. Compared with the anomaly detection approach which is based on the aiNet clustering and the aiNet based hierarchical clustering (aiNet_HC), respectively, the results show that ADAID is a radical anomaly detection approach and can achieve higher accuracy rates.

Keywords: Anomaly Detection; aiNet; Clustering Algorithm; Density Peak; Network Flow

1 Introduction

With the rapid development of information technologies and the universal application of electronic productions, network security problem has become severe society focus in our daily life. Nowadays, there are millions of network viruses and malicious attacks in different network environments, and many updated versions of them or novel attacks are produced constantly. The targets of network attacks mainly include network nodes, terminal computers, and smart devices, especially smartphone providing network admission and payment function [9]. To evaluate effectively cyberspace security, many security strategies are employed, such as private protection, firewall mechanism, virus defense, intrusion detection and risk evaluation *etc.*

Anomaly detection is one key component part of the intrusion detection system [11]. Up to now, anomaly detection strategy has been applied to many application areas, such as network security system, industrial control system, and Internet of Things *etc.* The merits of anomaly detection [2,25,27] can detect unknown malicious attacks from the captured network packets real-timely in network system environments. In traditional anomaly detection system, administrators firstly need define the legitimate profiles for the protected network system, the anomaly detection system will alarm if the detected network behaviors aren't normal. Due to the misuse detection strategy, another important intrusion detection method, holding the known malicious attack characteristic and the higher detection rates, some researchers have proposed the improved anomaly detection system combining with misuse detection technique to raise the detection rates (DRs) of known malicious attacks and decrease false alarm rates (FARs) of unknown attacks [3].

Compared with the packet anomaly detection, the flow anomaly detection analyzes network security problem by network flows, and it can solve some problems which are processing time and data reduction [23]. Network flow is viewed as a description approach of network behaviors based on the connections of network terminals and records high-level description of network connections, but network flow isn't real network packet [14]. Network flow is a bidirectional or unidirectional sequence of packets traveling between two network terminals using network protocols (e.g. TCP/UDP) with common features [18]. The most important features of network flow include duration time, source/destination IP address, source/destination port number, and the transferred source/destination packets *etc.* The inherent rules of network flows can be analyzed by the common features of the sending/receiving protocol packets, especially TCP flows. At present, the flow anomaly detection has become a research hotspot, and meanwhile it is regarded as an effective complement of packet inspection [7,8,14].

As an important machine learning method, clustering analysis is applied widely to solve network security problem, especially detecting malicious attack behaviors from the massive network flows. Clustering analysis is aimed at classifying the given data elements into categories based on their similarity [22]. Clustering, an unsupervised classification approach, doesn't provide available labeled elements during training phase. The procedure of clustering analysis involves four basic stages [30]: Feature selection and extraction, clustering algorithm design and selection, clustering validation, results interpretation. Many researchers think that clustering holds the internal homogeneity and the external separation, *i.e.* elements in a cluster possessing similar pattern. The representative clustering techniques [30] include hierarchical clustering, partitional clustering, and evolutionary clustering *etc.* As one type of the most difficult and challenging problems in machine learning fields, many evolutionary clustering algorithms, such as artificial immune system, genetic algorithm and artificial neural network, are proposed successively to analyze the unsupervised nature problem, and the relevant data spatial distribution is unknown [4,16,31].

In this paper, an Anomaly Detection approach for network flow using Artificial Immune network and Density peak (ADAID) is proposed. To obtain more precise samples and cluster number from network flows, the aiNet [4] is used for coarse-grained clustering, and CDP [22] is adopted for fine-grained clustering according to the output results of coarse-grained clustering. To raise detection rates and decrease false alarm rates, we devise the CLA algorithm in this paper to label normal/abnormal clusters, and ISCX 2012 IDS dataset [26] is adopted to detect anomaly network flows. The mainly contributions of this paper include:

1) Propose an anomaly detection framework (ADAID), to detect malicious attack behaviors of network flows;

2) Propose an unsupervised clustering algorithm (aiNet_DP) combining artificial immune network and density peaks;

3) Propose a cluster labeling algorithm (CLA) to distinguish effectively benign and malicious behaviors of network flows.

The remainder of this paper is organized as follows. We describe a review of the prior researches on the unsupervised anomaly detection based on clustering algorithm and artificial immune network in Section 2. Section 3 describes the proposed ADAID approach based on artificial immune network and density peak for the anomaly detection of network flows. Section 4 illustrates the performance evaluations of ADAID on ISCX IDS dataset. The conclusion is finally given in the last Section.

2 Related Works

The clustering algorithms have been proposed to solve anomaly detection problems of network flow [2]. Portnoy *et al.* [20] proposed a variant of single-linkage clustering based on distance to classify data instances. Leung *et al.* [13] proposed the density-based and grid-based high dimensional clustering algorithm for unsupervised anomaly detection of large datasets. Petrovic *et al.* [19] combined the Davies-Bouldin index of clustering and the centroid diameters of clusters to detect massive network anomaly attacks. Syarif *et al.* [28] investigated the performances of five different clustering algorithms for anomaly detection problem, namely, k-means, improved k-means, k-mediods, expectation maximization (EM) and distance-based outlier detection algorithm. The experimental results show that the distance-based outlier detection algorithm outperform other clustering algorithms, and some researchers have obtained remarkable outcomes by using the clustering-based anomaly detection for network flows. Erman *et al.* [5] proposed a semi-supervised clustering method, which consists of a learner and a classifier, to classify network flows. Munz *et al.* [17] proposed flow anomaly detection approach based on K-means clustering algorithm. The training data used in this approach, which are unlabeled network flows, are separated into clusters of normal and malicious network flows, and the obtained cluster centroids can be used for detecting anomaly behaviors from on-line monitoring data. Ahmed *et al.* [1] used X-means clustering to detect collective anomaly flows. The X-means clustering is a variant of K-means algorithm, and provide an effective strategy to select the number of clusters k. Sheikhan *et al.* [23] proposed NIDS based on artificial neural network for detecting anomaly attacks of network flows. This system identifies malicious and benign flows using multi-layer perceptron neural classifier, and uses the gravitational search algorithm to optimize the interconnection weights of neural anomaly detector. Winter *et al.* [29] presented network intrusion detection approach to analyze anomaly flows, and used One-Class Support Vector Machines to identify malicious network flow. Therefore, the advantages of the anomaly detection approach based on clustering algorithm mainly include:

1) Generate anomaly detectors by self-learning approach;

2) Extract common features from the given dataset;

3) Detect unknown malicious attack behaviors from the changeable network environment.

Artificial immune network is one of important theories of artificial immune system inspired by vertebrate immune system, and holds some merits of artificial immune system, such as self-learning, self-adaption, self-organization and immune memory *etc.* [24]. According to immune network theory [10], the binding between idiotopes (molecular portions of an antibody) located on B cells and paratopes (other molecular portions of an antibody) located on B cells has a stimulation effect for B cells, and the interaction of B cells within a network will produce to a stable memory structure and account for the retainment of memory cells. For clustering algorithm inspired by immune network theory, the antibodies in immune network will be suppressed when similarity between antibodies is higher, conversely, they will be stimulated [4]. As a result, the expected network will be generated and its redundant antibodies will be eliminated. In recent years, artificial immune network has been employed by intrusion detection system to cluster anomaly malicious behaviors. Liu *et al.* [6] proposed an unsupervised anomaly detection algorithm based on artificial immune network, and the hierarchical agglomerative clustering is employed to help clustering analysis. Shi *et al.* [25] proposed an unsupervised UADINK approach based on K-means improved by immune network theory to detect anomaly behaviors of network flows. Lau *et al.* [12] proposed an unsupervised anomaly detection architecture which is capable of online adaptation inspired by immune network theory. Rassam *et al.* [21] investigated artificial immune network for clustering malicious attacks of intrusion detection system, and the rough set principle is employed to get the key element features of the given dataset so as to enhance detection rate of this system. These mentioned anomaly detection approaches show that artificial immune network can be used effectively for clustering network flows and refining detectors of anomaly detection system.

3 The Proposed ADAID

The proposed ADAID approach is an unsupervised anomaly detection strategy, and provides an automatic mechanisms to detect anomaly behaviors of network flows, therefore, it doesn't need the samples labeled by experts in order to cluster network flows. The framework of ADAID is shown in Figure.1. ADAID mainly includes four aspects:

1) Obtain network flows. They can be generated by replaying network packets of the given benchmark dataset or captured by real network world.

2) Select common features of network flows. We need select typical features of each network flow which can identify easily network behaviors in order to effectively distinguish malicious attack behaviors.

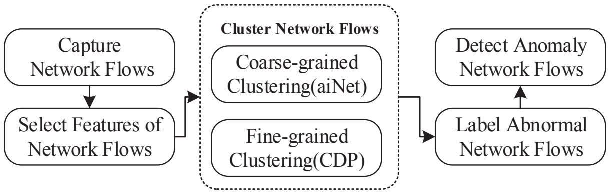
Capture Network Flows
Select Features of Network Flows
Cluster Network Flows
Coarse-grained Clustering(aiNet)
Fine-grained Clustering(CDP)
Detect Anomaly Network Flows
Label Abnormal Network Flows

Figure 1: The framework of ADAID

3) Cluster network flows. It relates to two stages, namely the coarse-grained stage and the fine-grained stage. In the coarse-grained clustering stage, the aiNet model is introduced firstly for clustering samples from the given dataset [4]. The CDP algorithm [22],which is the fine-grained clustering, is used for clustering the output results of the coarse-grained clustering, and the aim that employ the CDP algorithm is to refine the cluster centroids from the previous stage and improve the attack detection accuracy of network flows.

4) Label abnormal network flows. After the final cluster centroids are obtained, each cluster centroid represents one of class network flows. Therefore, these cluster centroids need be labeled as abnormal/normal network flows so that ADAID can detect easily anomaly attacks of network flows. The relevant models and algorithms that compose ADAID are described as the following subsections, namely, artificial immune network (aiNet), clustering algorithm based on density peaks (CDP), clustering labeling algorithm (CLA) and flow anomaly detection algorithm (FAD).

3.1 The aiNet Model

The artificial immune network (aiNet) model is inspired by the clone selection principle and immune network theory of vertebrate immune system. The aiNet model [4] is firstly used for analyzing and filtering the crude dataset, and an internal image of all data samples in dataset, namely a refined relationship map, is constructed by immune evolution mechanisms, such as self-organizing, self-adaptive and self-learning *etc.* Therefore, the aiNet model is regarded as a coarse-grained method to refine some important features from complex information data. At present, the aiNet model has been introduced in pattern recognition, clustering data, and data compression *etc.*

The aiNet model is given in Figure 2. Its mainly aim is to search optimal memory antibodies of antigen ag_j by immune optimization strategies. This model may generate a memory antibody subset M_j in terms of the given antigen ag_j. After all antigens are travelled, the memory antibody set M will aggregate and storage the optimal antibodies. The antibody of M will be suppressed in each iterative operation of this model in order to avoid similar antibodies entering next generation. The memory antibody set

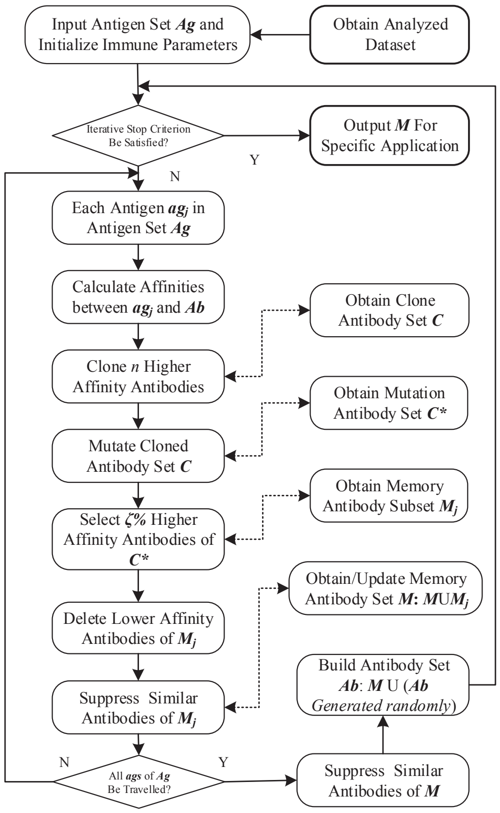
Input Antigen Set Ag and Initialize Immune Parameters
Obtain Analyzed Dataset
Iterative Stop Criterion Be Satisfied?
Output M For Specific Application
Y
N
Each Antigen ag_j in Antigen Set Ag
Calculate Affinities between ag_j and Ab
Obtain Clone Antibody Set C
Clone n Higher Affinity Antibodies
Obtain Mutation Antibody Set C*
Mutate Cloned Antibody Set C
Obtain Memory Antibody Subset M_j
Select ζ% Higher Affinity Antibodies of C*
Obtain/Update Memory Antibody Set M: M∪M_j
Delete Lower Affinity Antibodies of M_j
Build Antibody Set Ab: M ∪ (Ab Generated randomly)
Suppress Similar Antibodies of M_j
N
Y
All ags of Ag Be Travelled?
Suppress Similar Antibodies of M

Figure 2: The flowchart of the aiNet model

M will be outputted as the final results or preprocessing data of the specific application system if the iterative stop criterion of this model is satisfied, for example, obtaining the cluster number/centroids of the relevant clustering algorithms. Therefore, the design of immune optimization strategies is a vital phase to improve the evolution learning capabilities of aiNet [4], such as clonal selection, immune mutation, and antibody suppression *etc.*

3.2 The CDP Algorithm

The clustering algorithm based on density peaks (CDP) [22] mainly includes three aspects:

1) Compute the local density ρ_i for each data point i of the given dataset, and the minimum distance δ_i between the data point i and any other data points with higher density.

2) Obtain cluster centroids by the drawn decision graph in terms of the local density and the minimum distance of each data of dataset, the cluster centroids possess both wider distance and higher density.

3) Assign each remaining data point of dataset to the same cluster centroid as its nearest neighbor of high density. The CDP algorithm can fast search and find density peaks by the specific functions which are used for calculating local density and distance of each data point of dataset.

For the CDP algorithm [22], Equations (1) and (2) are used for calculating ρ_i of each data point i, where d_c represents a cutoff distance, Equation (3) is used for calculating δ_i between each data point i and any other points with higher density, Equation (4) is used for discovering the power law distribution of all data points, and some data points that possess higher γ can be selected as cluster centroids.

$$\rho_i = \sum_{j \neq i} \chi(d_{ij} - d_c) \tag{1}$$

$$\chi = \begin{cases} 1, if(d_{ij} - d_c) < 0 \\ 0, \quad otherwise \end{cases} \tag{2}$$

$$\delta_i = \min_{j:\rho_j > \rho_i} (d_{ij}) \tag{3}$$

$$\gamma_i = \rho_i \cdot \delta_i \tag{4}$$

According to the idea of ADAID, the CDP algorithm is viewed as a fine-grained clustering algorithm to classify effectively network flows, and the clustered data in CDP are the refined network flows that are learned by aiNet. The CDP algorithm is described by Algorithm 1.

Algorithm 1 The CDP Algorithm

```
Input: Memory antibody set $M$ refined by aiNet
Output: Cluster number set $T$ of $M$
Start
Calculate the distance $d$ between each data point and any other data points in $M$, and find a cutoff distance $d_c$ according to $d$ of each data point in $M$
Calculate $\rho_i$,$\delta_i$,$\gamma_i$ by Equation (1), Equation (3) and Equation (4), respectively
Determine cluster centroids according to the power law distribution $\gamma$ of all data points
Assign the rest of data points in $M$ to the corresponding cluster centroid according to $\rho_i$, and finally obtain cluster number set $T$
End
```

3.3 The CLA Algorithm

The Cluster Labeling Algorithm (CLA) is used for labeling each cluster as normal/abnormal detector of network flows, and then these generated detectors are used for distinguishing malicious/benign network flows. In CLA, the labeled results for the corresponding clusters will influence anomaly detection performance of ADAID. The CLA algorithm is described by Algorithm 2.

3.4 The FAD Algorithm

The aim of the flow anomaly detection algorithm (FAD) is that provides an anomaly detection function for network

Algorithm 2 The CLA Algorithm

```
Input: Memory antibody set $M$, Cluster number set $T$, Training dataset $Ag$, Recognition threshold $Rt$
Output: Label set $Nal$ of clusters
Start
Determine size of $Nal$, preprocess $Ag$
for each antigen of $Ag$ do
    Calculate affinity of each antibody in $M$
    Find an antibody with maximum affinity, and accumulate the appeared times of this antibody
end for
for each different cluster number in $T$ do
    Accumulate the matched times of different antibodies of $M$ with antigens of $Ag$, and the cluster number of each antibody should keep same with $T$
    Calculate percent ratio $Pr$ that each different cluster has recognized antigens of $Ag$
    if $Pr$ is not less than $Rt$ then
        Storage the number of this cluster and label this cluster as normal cluster in $Nal$
    else
        Storage the number of this cluster and label this cluster as abnormal cluster in $Nal$
    end if
end for
End
```

flows. Therefore, administrators can obtain network security situation by using FAD, and then some security strategies can be deployed timely. The FAD algorithm is described by Algorithm 3.

Algorithm 3 The FAD Algorithm

```
Input: Memory antibody set $M$, Cluster number set $T$, Label set $Nal$, Test dataset $Tag$
Output: Alarmed network flows which can match abnormal clusters of $Nal$
Start
Preprocess the test dataset $Tag$
for each antigen of $Tag$ do
    Calculate affinities between each antibody in $M$ and this antigen
    Choose an antibody with maximum affinity, and identify its cluster number in $T$
    if cluster number of this chosen antibody in $T$ is equal to abnormal cluster in $Nal$ then
        Alarm and Output this antigen, namely find an abnormal network flow
    end if
end for
End
```

4 Experimental Results

4.1 Dataset Description

To verify the effectiveness of the proposed ADAID, the ISCX 2012 IDS dataset [26] is adopted as benchmark dataset to detect malicious behaviors of network flows. This dataset includes seven days capturing data with overall 2,450,324 network flows, and is designed by the University of New Brunswick. In our evaluation experiments, the Tuesday's sub-dataset (23.4GB) of the ISCX 2012 IDS dataset is considered, and its brief statistics is listed by Table 1. Due to existing only a few malicious network flows from the 1^{st} flow to the $375{,}664^{th}$ flow in the Tuesday's sub-dataset, we select 196034 network flows from the $375{,}665^{th}$ flow to the last flow in this sub-dataset to demonstrate the effectiveness of the proposed ADAID. The trained/tested network flows consist of 158576 benign flows and 37458 malicious attack flows, and Table 2 shows the distribution of malicious attack flows of the selected network flows. The 10 percent flows of the selected network flows are viewed as training samples in order to generate detectors, and the rest network flows of that are viewed as test samples in order to verify the detection capability of ADAID.

4.2 Dataset Preprocessing

The preprocessing operation for data samples of the given dataset plays an important role in the machine learning fields, and it mainly relates to feature selection and dimension reduction. Considering the common features of network flows, we extracted 10 typical features of the ISCX 2012 IDS dataset listed by Table 3 to analyze malicious behaviors of network flows in terms of the empirical methods of the existed literatures [15,23]. The aim of the preprocessing operation for network flows is that it may not only improve the anomaly detection precision but save the running costs both times and spaces in anomaly detection system.

As a key part of the preprocessing operation for the selected data sample, it's necessary that the key features of network flows are processed numerically. The minimum/maximum values of each selected feature is listed by Table 3. For the numeric range of these listed features, their default values are assigned according to the definitions and specifications of TCP/IP protocols. For instance, the fifth flag option of TCP header, SourceTCPFlags, is set to [0, 63]. For the rest features listed by Table 3, their maximum values aren't be limited, but they should be greater than the real values of any selected network flows. Take the third feature as an example, it is set to [0, 40,000] because the largest value of the transferred destination packets in any flows is not greater than 38,685.

Table 1: Tuesday's network flow statistics in the ISCX 2012 IDS dataset

Feature	**Value**	**Feature**	**Value**
Flows	571,698	Destination Bytes	22,842,855,364
Attack Flows	37,460	Source Bytes	1,905,193,956
Normal Flows	534,238	Destination Packets	21,746,115
ICMP Flows	6,073	Source Packets	13,254,945
TCP Flows	441,563	Destination IPs	26,780
UDP Flows	124,023	Source IPs	2,196

Table 2: Distribution of malicious network flows in the selected network flows

Network Flows	**Attacks of**	**Network Flows**	**Attacks of**
19,603 (10%)	79	117,620 (60%)	7,054
39,207 (20%)	82	137,224 (70%)	18,511
58,810 (30%)	83	156,827 (80%)	29,363
78,414 (40%)	84	176,431 (90%)	37,421
98,017 (50%)	85	196,034 (100%)	37,458

4.3 Evaluation Matrices

Anomaly detection is viewed as one kind of two-class problems. Network flow behaviors can be classified as benign behaviors or malicious behaviors by using anomaly detection algorithms. In this paper, we introduce three metrics to evaluate the performance of ADAID [25]:

1) Accuracy Rate (AR) that indicates the clustered correctly portion for all test samples of network flows, and its formal definition is shown in Equation (5);

2) Detection Rate (DR) that indicates the malicious attack flows which may be recognized correctly from test samples, and its formal definition is shown in Equation (6);

3) False Alarm Rate (FAR) that indicates the real benign flows which have been recognized as malicious attack flows from test samples, and its formal definition is shown in Equation (7).

In Equations (5) ,(6) and (7), TP(True Positive) indicates the cumulative number for the malicious attack flows which are labeled as real attack flows in test samples, FP(False Positive) indicates the cumulative number for the malicious attack flows which are labeled as benign flows in test samples, TN(True Negative) indicates the cumulative number for the benign flows which are labeled as normal network flows in test samples, and FN(False Negative) indicates the cumulative number for the benign flows which are labeled as malicious attack flows in test samples. To avoid bias, the final results of these evaluation metrics are given by the average results of Nr (=10) independent trials.

$$AR = \frac{TP+TN}{TP+FP+TN+FN} \tag{5}$$

$$DR = \frac{TP}{TP+FP} \tag{6}$$

$$FAR = \frac{FP}{TN+FP} \tag{7}$$

4.4 Parameter Settings

4.4.1 Evolution Parameters of aiNet

To demonstrate the effectiveness of ADAID, three clustering algorithms, namely aiNet model, aiNet based hierarchical clustering (aiNet_HC), and the proposed clustering algorithm combining aiNet with CDP (aiNet_DP), use same evolution parameter values listed by Table 4.

4.4.2 Parameter Settings of CDP

The cutoff distance dc and the cluster number nc are two key parameters of CDP, and can improve the clustering precision of network flows. The parameter dc represents a border region of each cluster. For the cluster centroid of each cluster, if the distance between this cluster centroid and one of data/vector points of the clustered dataset is not greater than dc, this data/vector point will be assigned to this cluster. Therefore, dc is an important parameter to discriminate correctly different clusters. Known from Reference [22], supposing nd represents the number of data/vector points of the clustered dataset, $n = \lceil(0.5 * (nd - 1) * nd)\rceil$ represents the total number of points by calculating distance between any two different data/vector points of the clustered dataset, and the value of dc can be chosen any one point around the former 1-2% of the total number of points after these points are sorted in ascending order. The larger dc is, the lesser the number of clusters are; conversely, the smaller dc is, the more the number of clusters are. The dc in this paper is obtained from one point around 1.5% of the total number of points in the clustered dataset.

To obtained reasonable nc of dataset, we firstly need calculate $ri = pi * di$ in Equation (4) after choosing a suitable dc, and ri is used for exhibiting a power law distribution of all data points, and then all elements in r are re-sorted in descend order, where pi denotes the local

Table 3: Flow feature description for the ISCX 2012 IDS dataset

Feature name	**Description**	**Minimum Value**	**Maximum Value**
TotalDestinationBytes	Transferred destination octets	0	60,000,000
TotalSourceBytes	Transferred source octets	0	2,000,000
TotalDestinationPackets	Transferred destination packets	0	40,000
TotalSourcePackets	Transferred source packets	0	20,000
DestinationTCPFlags	Destination TCP flags	0	63
SourceTCPFlags	Source TCP flags	0	63
DestinationPort	Destination port number	0	65,535
SourcePort	Source port number	0	65,535
ProtocolName	IP protocol number	0	255
Duration	Duration of flow (in seconds)	0	864,000

Table 4: Evolution parameters of the aiNet model

Parameter	**Value**	**Parameter**	**Value**
Number of Runs Nr	10	Re-selection Rate Rr	0.2
Number of Generations Ng	10	Hypermutation Rate Hr	4
Population Size Ps	10	Natural Death Threshold Nt	1
Taken Best-matching Cells Tbc	4	Suppression Threshold St	0.1

density of each data point i, and di denotes its distance from points with higher density. The i-th data point with corresponding to ri has more chance as a cluster centroid if ri is more bigger [22]. The nc will be set to 35% of the total number of r in this paper, and the total number of r depend on the output results of the coarse-grained clustering stage.

4.4.3 Parameter Settings of CLA

The recognition threshold Rt is an important parameter of CLA, and it is used for labeling normal/abnormal clusters. A reasonable selected Rt can increase the DRs and decrease the $FARs$ of anomaly detection system. There are two strategies to obtain the reasonable value of Rt. The first strategy is that the ratio, which is 10 percent of all samples of training dataset, may be considered as the value of Rt. The second strategy is that the ratio between the existing real attacks and the total amount samples in training dataset also may be considered as the value of Rt. Known from Table 2, there are 79 real attack flows in all 19603 network flows of training dataset, so the highest attack ratio in training dataset is about 0.004. According to the first strategy, one kind of network flows is regarded as normal if its amount of network flows isn't less than 10 percent of all samples in training dataset, namely Rt=0.1. Therefore, the value of Rt may be defined from 0.0040 to 0.1 according to the above-mentioned two strategies, but the reasonable value of Rt should close to 0.0040 in order to detect effectively anomaly network flows. The experimental results, which are AR, DR, and FAR, of the proposed ADAID are shown by Figure 3. Known from Figure 3, AR, DR and FAR of ADAID have got different results according to the change of Rt that ranges from 0.0040 to 0.0051. Rt in ADAID is set to 0.0046 in this paper, and the corresponding AR, DR, and FAR are 85.93%, 100% and 14.64%, respectively.

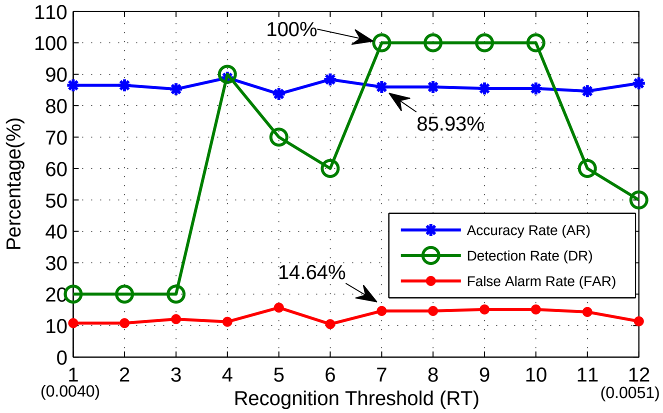

Figure 3: Performance comparison of ADAID with different Rt

4.4.4 Performance Evaluation of ADAID

Known from the proposed ADAID, the clustering algorithm is reviewed as a vital part of anomaly detection strategy. In this paper, we discuss the performances of three different clustering algorithms, which are aiNet, aiNet_HC and the proposed aiNet_DP, to detect anomaly behaviors of network flows. After running clustering operation for network flows, CLA and FAD are used for recognizing malicious clusters of network flows and detecting anomaly behaviors of network flows, respectively. Table 5 shows the experimental results of three different anomaly detection approaches.

Known from Table 5, compared with the aiNet based anomaly detection approach, the accuracy rates (ARs) of the aiNet_DP based anomaly detection approach in training stage and test stage are reach to 85.93% and 85.78%, respectively. And the corresponding false alarm rates ($FARs$) are only 14.64% and 15.28%, respectively. Therefore, the aiNet_DP based anomaly detection approach possesses higher ARs and lower $FARs$ than the aiNet based anomaly detection approach. Although the aiNet_HC based anomaly detection approach possesses higher ARs and lower $FARs$ than ADAID, its detection rates (DRs) in training stage and test stage are only reach to 70% and 70.75%, respectively. Obviously, the DRs of ADAID are about 30% higher than the aiNet_HC based anomaly detection approach. The deviation of the ARs between ADAID and the aiNet_HC based anomaly detection approach in training stage and test stage do not exceed 5%, and meanwhile the deviation of $FARs$ of them do not exceed 6%.

The aiNet based unsupervised clustering is regarded as an effective strategy for detecting network anomaly behaviors in anomaly detection system. The experimental results show that the aiNet based anomaly detection approach has more improvement space to enhance its ARs and reduce its $FARs$. Therefore, the improved clustering algorithm combining aiNet with other clustering algorithm is considered as more radical method to improve the effectiveness of clustering algorithm, such as aiNet_HC and aiNet_DP listed by Table 5. Compared with the aiNet based anomaly detection approach, the DRs of the aiNet_HC based anomaly detection approach decline even if its ARs and $FARs$ are improved. However, compared with two anomaly detection approaches which are respectively based on aiNet and aiNet_HC, the proposed ADAID combining aiNet with density peaks is more ideal approach for detecting anomaly behaviors of network flows because it possesses precise DRs, higher ARs and reasonable $FARs$.

5 Conclusions

An anomaly detection approach for network flow using artificial immune network and density peak (ADAID) in this paper is proposed to detect malicious attack behaviors and benign activities of network flows. In ADAID, its clustering algorithm consists of aiNet and CDP, where aiNet and CDP are viewed as coarse-grained clustering and fine-grained clustering, respectively. The aim of this clustering algorithm is to cluster similar values of common features from massive network flows and finish the classification of network flows. The anomaly detection of ADAID comprises of CLA and FAD, where CLA is to label clusters as abnormal or normal by learning network flows of training dataset, and the identified clusters are viewed as detectors; FAD can be used for detecting malicious attack behaviors from network flows of test dataset.

To demonstrate the effectiveness of ADAID, we firstly introduce three different clustering algorithms, namely, aiNet, aiNet_HC and the proposed aiNet_DP, to classify network flows of training dataset, respectively. The output clusters generated by three clustering algorithms all are labeled by CLA. And then the labeled clusters use FAD to detect network flows of test dataset. To improve the performance of ADAID, we analyzed the parameters of CDP, namely cutoff distance dc and cluster number nc, to obtain more precise clusters of network flows, and meanwhile we discussed the recognition threshold Rt of CLA to distinguish reasonably malicious flows and benign flows. In our experiments, the ISCX 2012 IDS dataset is adopted to evaluate ADAID. To avoid bias, the final experimental results are given by the average experimental results of Nr independent trials, and show that ADAID is a radical anomaly detection approach for network flows.

We will further improve ADAID in our future works that relates to unsupervised clustering, automatic detection, running costs *etc.* We will try to adopt more efficient immune optimizing strategies and parallel computing approaches to improve ADAID for detecting anomalies of network flows.

Acknowledgments

This work was funded by China Scholarship Council, Australian Research Council Discovery Project DP150104871, the China Postdoctoral Science Foundation under Grant No.2014M562102, Hunan Provincial Natural Science Foundation of China under Grant No.2015JJ2112, the Scientific Research Fund of Hunan Provincial Education Department of China under Grant No.18A449. The authors gratefully acknowledge the anonymous reviewers for their valuable comments.

References

[1] M. Ahmed and A. N. Mahmood, "Network traffic analysis based on collective anomaly detection," in *The 9th IEEE Conference on Industrial Electronics and Applications*, vol. 2014, pp. 1141–1146, 2014.

[2] M. Ahmed, A. N. Mahmood, and J. Hu, "A survey of network anomaly detection techniques," *Journal of Network and Computer Applications*, vol. 60, pp. 19–31, 2016.

[3] A. L. Buczak and E. Guven, "A survey of data mining and machine learning methods for cyber security intrusion detection," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 2, pp. 1153–1176, 2016.

[4] L. N. de Castro and F. J. Von Zuben, "aiNet: An artificial immune network for data analysis," *Data Mining: A Heuristic Approach*, vol. 2001, no. 1, pp. 231–259, 2001.

Table 5: Accuracy comparison for clustering algorithm based anomaly detection

Anomaly Detection	Training phase			Test phase		
	AR(%)	DR(%)	FAR(%)	AR(%)	DR(%)	FAR(%)
aiNet Based	76.43	100	24.53	76.39	100	23.71
aiNet_HC Based	90.47	70	8.70	89.37	70.75	10.55
ADAID	85.93	100	14.64	84.78	100	15.28

[5] J. Erman, A. Mahanti, M. Arlitt, I. Cohen, and C. Williamson, "Offline/realtime traffic classification using semi-supervised learning," *Performance Evaluation*, vol. 64, no. 9-12, pp. 1194–1213, 2007.

[6] L. Fang and L. Le-Ping, "Unsupervised anomaly detection based on an evolutionary artificial immune network," in *Workshops on Applications of Evolutionary Computation*, pp. 166–174, 2005.

[7] Y. Hamid, V. R. Balasaraswathi, L. Journaux, and M. Sugumaran, "Benchmark datasets for network intrusion detection: A review," *International Journal of Network Security*, vol. 20, no. 4, pp. 645–654, 2018.

[8] Y. He, "Identification and processing of network abnormal events based on network intrusion detection algorithm," *International Journal of Network Security*, vol. 21, no. 1, pp. 153–159, 2019.

[9] M. S. Hwang, S. K. Chong, and H. H. Ou, "On the security of an enhanced umts authentication and key agreement protocol," *European Transactions on Telecommunications*, vol. 22, no. 3, pp. 99–112, 2011.

[10] N. K. Jerne, "Towards a network theory of the immune system," in *Annales d'immunologie*, vol. 125, pp. 373–389, 1974.

[11] D. Kwon, H. Kim, J. Kim, C. S. Sang, I. Kim, and K. J. Kim, "A survey of deep learning-based network anomaly detection," *Cluster Computing*, no. 5, pp. 1–13, 2017.

[12] H. Lau, J. Timmis, and I. Bate, "Anomaly detection inspired by immune network theory: A proposal," in *IEEE Congress on Evolutionary Computation*, pp. 3045–3051, 2009.

[13] K. Leung and C. Leckie, "Unsupervised anomaly detection in network intrusion detection using clusters," *Proceedings of the Twenty-eighth Australasian conference on Computer Science*, vol. 8, pp. 333–342, 2005.

[14] B. Li, J. Springer, G. Bebis, and M. H. Gunes, "A survey of network flow applications," *Journal of Network and Computer Applications*, vol. 36, no. 2, pp. 567–581, 2013.

[15] W. Li, M. Canini, A. W. Moore, and R. Bolla, "Efficient application identification and the temporal and spatial stability of classification schema," *Computer Networks*, vol. 53, no. 6, pp. 790–809, 2009.

[16] D. S. A. Minaam and E. Amer, "Survey on machine learning techniques: Concepts and algorithms," *International Journal of Electronics and Information Engineering*, vol. 10, no. 1, pp. 34–44, 2019.

[17] G. Munz, S. Li, and G. Carle, "Traffic anomaly detection using k-means clustering," *GI/ITG Workshop MMBnet*, 2007. (`http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.323.6870&rep=rep1&type=pdf`)

[18] A. W. Moore, Z. Denis, and M. L. Crogan, "Discriminators for use in flow-based classification," *Queen Mary and Westfield College, Department of Computer Science*, 2005. (`https://www.cl.cam.ac.uk/~awm22/publications/RR-05-13.pdf`)

[19] S. Petrovic, G. Alvarez, A. Orfila, and J. Carbo, "Labelling clusters in an intrusion detection system using a combination of clustering evaluation techniques," *Proceedings of the 39th Annual Hawaii International Conference on System Sciences (HICSS'06)*, vol. 6, pp. 129b–129b, 2006.

[20] L. Portnoy, E. Eskin, and S. Stolfo, "Intrusion detection with unlabeled data using clustering," in *Proceedings of ACM CSS Workshop on Data Mining Applied to Security*, 2001. (`http://citeseerx.ist.psu.edu/viewdoc/citations?doi=10.1.1.126.2131`)

[21] M. A. Rassam and M. A. Maarof, "Artificial immune network clustering approach for anomaly intrusion detection," *Journal of Advances in Information Technology*, vol. 3, no. 3, pp. 147–154, 2012.

[22] A. Rodriguez and A. Laio, "Clustering by fast search and find of density peaks," *Science*, vol. 344, no. 6191, pp. 1492–1496, 2014.

[23] M. Sheikhan and Z. Jadidi, "Flow-based anomaly detection in high-speed links using modified gsa-optimized neural network," *Neural Computing and Applications*, vol. 24, no. 3-4, pp. 599–611, 2014.

[24] Y. Shi, R. Li, X. Peng, and G. Yue, "Network security situation prediction approach based on clonal selection and scgm(1,1)c model," *Journal of Internet Technology*, vol. 17, no. 3, pp. 421–429, 2016.

[25] Y. Shi, X. Peng, R. Li, and Y. Zhang, "Unsupervised anomaly detection for network flow using immune network based k-means clustering," in *International Conference of Pioneering Computer Scientists, Engineers and Educators*, pp. 386–399, 2017.

[26] A. Shiravi, H. Shiravi, M. Tavallaee, and A. A. Ghorbani, "Toward developing a systematic approach to generate benchmark datasets for intrusion detection," *Computers & Security*, vol. 31, no. 3, pp. 357–374, 2012.

[27] N. Shone, T. N. Ngoc, V. D. Phai, and Q. Shi, "A deep learning approach to network intrusion detection," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 2, no. 1, pp. 41–50, 2018.

[28] I. Syarif, A. Prugel-Bennett, and G. Wills, "Unsupervised clustering approach for network anomaly detection," in *International Conference on Networked Digital Technologies*, pp. 135–145, 2012.

[29] P. Winter, E. Hermann, and M. Zeilinger, "Inductive intrusion detection in flow-based network data using one-class support vector machines," *IEEE 4th IFIP International Conference on New Technologies, Mobility and Security (NTMS'11)*, vol. 2011, pp. 1–5, 2011.

[30] R. Xu and D. Wunsch, "Survey of clustering algorithms," *IEEE Transactions on Neural Networks*, vol. 16, no. 3, pp. 645–678, 2005.

[31] J. Zhang, "Application of artificial intelligence technology in computer network security," *International Journal of Network Security*, vol. 20, no. 6, pp. 1016–1021, 2018.

Biography

Yuanquan Shi received the B. S. degree from Hunan Normal University, Changsha, China, in 2000, the M.E. degree from National University of Defense Technology, Changsha, China, in 2005, the Ph.D. Degree from Sichuan University, Chengdu, China, in 2011, all in computer science. Currently, he is Professor in the School of Computer Science and Engineering, Huaihua University, China, and also a visiting scholar in the School of Computer Sciences at the University of Adelaide, Australia. His research interests include network security, intelligent computing, time series prediction, and parallel computing.

Hong Shen is Professor (Chair) of Computer Science in University of Adelaide, Australia. He received the B.Eng. degree from Beijing University of Science and Technology, M.Eng. degree from University of Science and Technology of China, Ph.Lic. and Ph.D. degrees from Abo Akademi University, Finland, all in Computer Science. He was Professor and Chair of the Computer Networks Laboratory in Japan Advanced Institute of Science and Technology (JAIST) during 2001∼2006, and Professor (Chair) of Compute Science at Griffith University, Australia, where he taught 9 years since 1992. With main research interests in parallel and distributed computing, algorithms, data mining, privacy preserving computing, network security, high performance networks and multimedia systems, he has published more than 300 papers including over 100 papers in international journals such as a variety of IEEE and ACM transactions. Prof. Shen received many honours/awards including China National Endowed Expert of Thousand Talents, Chinese Academy of Sciences Hundred Talents, National Education Commission Science and Technology Progress Award, and Chinese Academy of Sciences Natural Sciences Award. He served on the editorial board of numerous journals and chaired several conferences.