Computers, Materials & Continua CMC, vol.61, no.1, pp.255-273, 2019

An Intrusion Detection Algorithm Based on Feature Graph

Xiang Yu[1], Zhihong Tian[2], Jing Qiu[2, *], Shen Su[2, *] and Xiaoran Yan[3]

Abstract: With the development of Information technology and the popularization of Internet, whenever and wherever possible, people can connect to the Internet optionally. Meanwhile, the security of network traffic is threatened by various of online malicious behaviors. The aim of an intrusion detection system (IDS) is to detect the network behaviors which are diverse and malicious. Since a conventional firewall cannot detect most of the malicious behaviors, such as malicious network traffic or computer abuse, some advanced learning methods are introduced and integrated with intrusion detection approaches in order to improve the performance of detection approaches. However, there are very few related studies focusing on both the effective detection for attacks and the representation for malicious behaviors with graph. In this paper, a novel intrusion detection approach IDBFG (Intrusion Detection Based on Feature Graph) is proposed which first filters normal connections with grid partitions, and then records the patterns of various attacks with a novel graph structure, and the behaviors in accordance with the patterns in graph are detected as intrusion behaviors. The experimental results on KDD-Cup 99 dataset show that IDBFG performs better than SVM (Supprot Vector Machines) and Decision Tree which are trained and tested in original feature space in terms of detection rates, false alarm rates and run time.

Keywords: Intrusion detection, machine learning, ids, feature graph, grid partitions.

1 Introduction

With the development of Information technology and the popularization of Internet, whenever and wherever possible, people can connect to the Internet optionally. Meanwhile, the security of network traffic is threatened by various of online malicious behaviors. Although some corresponding techniques, such as firewalls, data encryption and user authentication, have been developed to protect the security of computer and network traffic, it is still hard to prevent a wide range of intrusion behaviors. The aim of intrusion detection system is to detect malicious intrusion behaviors timely and further identify the

[1] School of Electronics and Information Engineering, Taizhou University, Taizhou, 318000, China.

[2] Cyberspace Institute of Advanced Technology, Guangzhou University, Guangzhou, 510006, China.

[3] Indiana University Network Science Institute, Bloomington, Indiana, 47408, USA.

[*] Corresponding Authors: Jing Qiu. Email: qiujing@gzhu.edu.cn;
Shen Su. Email: johnsuhit@gmail.com.

CMC. doi:10.32604/cmc.2019.05821

sources where the attacks are from. And then attack warnings are sent to the network administrators by IDS [Milenkoski, Vieira, Kounev et al. (2015); Ganapathy, Kulothungan, Muthurajkumar et al. (2013); Wu, Zhang, Zhang et al. (2018)]. According to the survey, most IDS are based on anomaly detection and signature detection [Shin, Lee, Kim et al. (2013); Venkatesan, Ganesan and Selvakumar (2012)]. Anomaly detection justifies whether an observed behavior is malicious or not by comparing it with the behavior patterns. Other than anomaly detection, signature detection focuses on mapping the contents of logs or packets to the recorded sequences of attack events or commands. However, the results of conventional intrusion detection largely depend on the attacks recorded in signature databases, which lead to high false alarm rates in most cases. Recently, some innovative approaches, such as data mining and machine learning, have been employed to train the model for anomaly detection [Upadhyaya, Jain, HOD et al. (2012); Kang and Oh (2012)]. The paper is also inspired by Tian et al. [Tian, Su, Shi et al. (2019); Wang, Liu, Qiu et al. (2018); Cheng, Xu, Tang et al. (2018)], as the notion of the Internet of things [Yu, Tian, Qiu et al. (2018); Qiu, Xing, Zhu et al. (2019); Wang, Tian, Zhang et al. (2018); Tian, Cui, An et al. (2018); Tan, Gao, Shi et al. (2018); Chen, Tian, Cui et al. (2018); Tian, Shi, Wang et al. (2019); Tian, Gao, Su et al. (2019); Du, Xiao, Guizani et al. (2007)], sensor [Wang, Gu and Yan (2018); Wang, Gu, Ma et al. (2017); Li, Sun, Jiang et al. (2018)], and big data [Qiu, Chai, Liu et al. (2018); Qiu, Qi, Wang et al. (2017)] are no longer limited to its own scope.

As shown in recent studies, lots of advanced approaches based on machine learning have been integrated with intrusion detection model to improve the performance of malicious behavior detection [Depren, Topallar, Anarim et al. (2005); Agarwal, Purwar, Biswas et al. (2017); Modi and Patel (2013)]. According to the related literature, several common methods, such as detection rate, false positive, false negative, etc., are employed to evaluate the performance of intrusion detection model [Shiravi, Shiravi, Tavallaee et al. (2012); Nasr, Abou-El Kalam and Fraboul (2012); Chen, Abraham and Yang (2007)].

In this paper, we present IDBFG (Intrusion Detection Based on Feature Graph), an intrusion detection model based on feature graph, which can detect malicious network traffic effectively. In IDBFG, a graph structure with grid partitions on each dimension is adopted to represent network traffic behaviors, and then the patterns of network traffic behaviors are obtained by matching the features of each behavior to the grid partitions. Based on the partitions, a graph of malicious network traffic behaviors is formed. Further, whether a network traffic behavior is malicious or not can be identified by comparing the features of the behavior with the nodes in the graph of the intrusion detection model.

The remainder of this paper is organized as follows. First, the previous related work on intrusion detection are introduced in Section 2. Second, together with the corresponding partition method, matching and decision algorithms, the IDBFG model is presented in Section 3. In Section 4, the experiments conducted to evaluate the performance of IDBFG on KDD-CUP'99 dataset are discussed. Finally, we conclude the paper and discuss the directions of our future work.

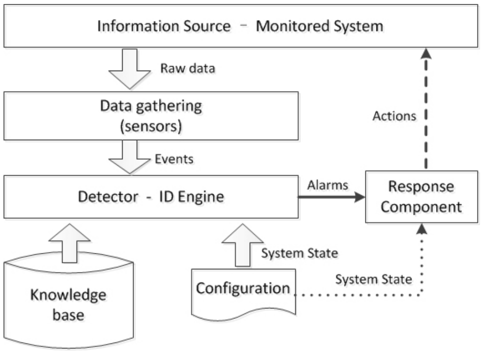

Figure 1: Basic architecture of IDS

2 Related work

In this section, we review the conventional and the advanced intrusion detection approaches which are integrated with supervised learning and unsupervised learning respectively.

2.1 Conventional intrusion detection approaches

As is well known, the objective of early approaches for securing computer systems is to design security mechanisms, which can protect computer systems from attacks by creating a shield around them. Nevertheless, these typical security mechanisms, including firewalls, virtual private networks and authentication mechanisms, cannot provide complete security, especially under continual attacks. Due to the vulnerabilities of security mechanisms, intrusion detection is introduced as a complement [Baig, Sait and Shaheen (2013); Shittu, Healing, Ghanea-Hercock et al. (2015)].

Since Dorothy Denning proposed the first intrusion detection model in 1987 [Jeya, Ravichandran and Ravichandran (2012)], lots of IDSs have been presented in both research and commercial fields [Pan, Morris and Adhikari (2015); Shon, Kovah and Moon (2006); Saad, Manickam and Ramadass (2013); Khan, Awad and Thuraisingham (2007)]. Most of these conventional IDSs are based on a general architecture with five main components involved as shown in Fig. 1 [Sultana and Jabbar (2016)].

From host-based to network-based, the focus of early intrusion detection research shifts from insider [Sarasamma, Zhu and Huff (2005); Ashraf and Habaebi (2015)] threats to the attacks which are from networks [Lin and Lee (2013); Özyer, Alhajj and Barker (2007)]. However, neither the host-based IDS nor the network-based IDS can identify unknown network traffic behaviors effectively without the analysis of past network traffic activities.

2.2 Advanced intrusion detection approaches

In general, methods of machine learning are usually used to collect information and extract knowledge from the information autonomously. When integrated with intrusion detection approaches, machine learning methods learn from previous experiences and train the intrusion detection model for the detection of malicious network traffic behaviors. In addition, the model can be improved by autonomous learning when there exist additional training data or knowledge. In summary, machine learning methods can be classified into two types, the supervised learning and the unsupervised learning, which are described as follows.

2.2.1 Intrusion detection with supervised learning

In most cases, the supervised learning trains a model with priori knowledge, such as training data. And the training data are generally represented as a set of vectors, each vector is composed of the values of condition attributes and decision attributes. When training model, the given training data are fed into a function, whose objective is to establish a approximate mapping model between the inputs and the outputs of training data. Based on the constructed model, the data with the values of condition attributes only, whose category is unknown, can be classified into the right category [Subaira and Anitha (2014)].

After years of research, lots of well-known supervised learning methods have been integrated with intrusion detection approaches, which include support vector machines (SVM), decision tree (DT), artificial neural network (ANN), genetic algorithm (GA) and so on [Liu, Yi and Yang (2007)]. Gisung Kim et al. present a novel hybrid intrusion detection model based on C4.5 decision tree in 2014. By analyzing packets, alarming system administrator and blocking attack connections, the IDS based on the model can effectively detect malicious network activities and prevent further damage from attacks [Kim, Lee and Kim (2014)]. Eesa et al. propose a new feature selection approach on the basis of decision tree for intrusion detection systems in 2015 [Islam, Seera and Loo (2017)]. WY Feng et al. propose a classification algorithm which combines support vector machine with ant colony network to improve run-time efficiency and classification rate in 2013 [Zhang, Jiang and Kamel (2005)]. T Shon et al. propose a hybrid approach for intrusion detection, it employs genetic algorithm to extract features for SVM learning [Shon and Moon (2007)]. HA Sonawane et al. evaluate the comparative performance of intrusion detection based on neural network and principal component analysis (PCA) in 2015 [Lin, Ke and Tsai (2015)]. In addition, there are also some intrusion detection approaches based on supervised learning, such as DT, GA, ANN, SVM [Sindhu, Geetha and Kannan (2012); Xiang, Yong and Meng (2008); Feng, Zhang, Hu et al. (2014); Tajbakhsh, Rahmati and Mirzaei (2009)].

2.2.2 Intrusion detection with unsupervised learning

Other than supervised learning, the unsupervised learning is a machine learning method which constructs model according to observations, such as clustering. Due to the absence of priori knowledge in unsupervised learning, the input data are not marked in advance and need to be processed or classified according to the similarity between each other. When the predefined objective function converges, the unsupervised learning process terminates and the learning results are identified as the outputs [Tong, Wang and Yu (2009)].

After years of research, some well-known unsupervised learning methods have been integrated with intrusion detection approaches, among which the representatives are k-means, kNN, fuzzy clustering and so on [Peddabachigari, Abraham, Grosan et al. (2007)]. Giorgio Giacinto et al. propose an intrusion detection approach based on k-means clustering, which can detect the attacks never observed before [Gaffney and Ulvila (2001)]. Y Li et al. propose, TCM-KNN, an active learning algorithm for intrusion detection based on kNN, which performs well with high detection rate and low false positives [Li and Guo (2007)]. G Wang et al. propose an intrusion detection approach using artificial neural networks and fuzzy clustering, which helps IDS achieve stronger stability, higher detection rate and less positive rate [Wang, Hao, Ma et al. (2010)]. In addition, there are also some intrusion detection approaches based on unsupervised learning, such as k-means, kNN [Julisch (2003); Huang, Ye, Xiong et al. (2016)].

2.3 Comparison of related work

In general, the existing intrusion detection approaches based on machine learning can be classified into three categories, which consist of the intrusion detection approaches combining with unsupervised learning methods, supervised learning methods and hybrid methods respectively.

As shown in Tab. 1, most of the proposed intrusion detection approaches select SVM, DT and kNN as the benchmarks, and KDD-CUP99 is the most commonly used data set for simulation. In addition, a significant part of related work put focus on advanced approaches in order to improve the performance of intrusion detection. Some of the advanced approaches tend to extract more representative features to improve the efficiency, however, it is hard to verify the effectiveness of feature selection on intrusion detection. In this paper, we use a graph structure to record the features of malicious behaviors which has never been mentioned before.

The indicators, which include detection rate (DR), false Alarm Rate (FAR), false negative (FN), false positive (FP) and true positive (TP), are the most commonly employed in evaluating the performance of intrusion detection approaches. However, the run time indicator, which is ignored frequently, is also important especially for online detection. Hence, when evaluating the performance of our approach, not only DR, FAR but also run time are considered simultaneously.

Based on the discussion above, we propose the approach described below, which takes all features into account, and further records the features of malicious behaviors as a graph structure. Since DT and SVM are conveniently used as benchmarks for all the other classifiers and are likely to provide reasonable classification performance in most applications, they are also adopted in the paper.

3 IDBFG model

The proposed approach, IDBFG, consists of two phases, the training phase and the detection phase, and the training phase is further divided into two steps. In the first step of training

Table 1: Comparison of intrusion detection approaches

Related work	Dataset	Baseline	Evaluation
Kim et al. [Kim, Lee and Kim (2014)].	KDD-Cup'99 DARPA1999	DT,SVM	DR,ROC
de la Hoz et al. [De la Hoz, de la Hoz, Ortiz et al. (2014)].	KDD-Cup'99	DT,NB,RF	DR,ROC
Liu et al. [Liu, Yi and Yang (2007)].	KDD-Cup'99	DT,SVM,SOM	DR,FA,FP
Peddabachigari et al. [Peddabachigari, Abraham, Grosan et al. (2007)].	KDD-Cup'99	DT,SVM	Accuracy
Lin et al. [Lin, Ying, Lee et al. (2012)].	KDD-Cup'99	DT,SVM	DR
Wang et al. [Wang, Hao, Ma et al. (2010)].	KDD-Cup'99	DT,NB,ANN	Precision,Recall F-measure
Xiang et al. [Xiang, Yong and Meng (2008)].	KDD-Cup'99	DT	DR,Run time
Feng et al. [Feng, Zhang, Hu et al. (2014)].	KDD-Cup'99	SVM	DR,FP,FN
Tajbakhsh et al. [Tajbakhsh, Rahmati and Mirzaei (2009)].	KDD-Cup'99	SVM,k-NN	DR,FP
Khan et al. [Khan, Awad and Thuraisingham (2007)].	DARPA1998	SVM	FP,FN,Accuracy
Li et al. [Li and Guo (2007)].	KDD-Cup'99	SVM,ANN,k-NN	TP,FP
Shon et al. [Shon and Moon (2007)].	DARPA1999	SVM	DR,FP,FN
Shon et al. [Shon, Kovah and Moon (2006)].	DARPA1998	SVM,ANN,k-NN	DR,FP,FN
Baig et al. [Baig, Sait and Shaheen (2013)].	KDD-Cup'99	ANN,NB	Accuracy,FP,FN, Precision,Recall
Zhang et al. [Zhang, Jiang and Kamel (2005)].	KDD-Cup'99	ANN	DR,FP
Tong et al. [Tong, Wang and Yu (2009)].	DARPA1999	ANN,SOM	DR,FP
Chen et al. [Chen, Abraham and Yang (2007)].	DARPA1998	ANN	FP,FN
Ozyer et al. [Özyer, Alhajj and Barker (2007)].	KDD-Cup'99	GA	DR
Sarasamma et al. [Sarasamma, Zhu and Huff (2005)].	KDD-Cup'99	SOM	DR,FP
Shin et al. [Shin, Lee, Kim et al. (2013)].	DARPA2000	Markov chain	DR,FP,ROC

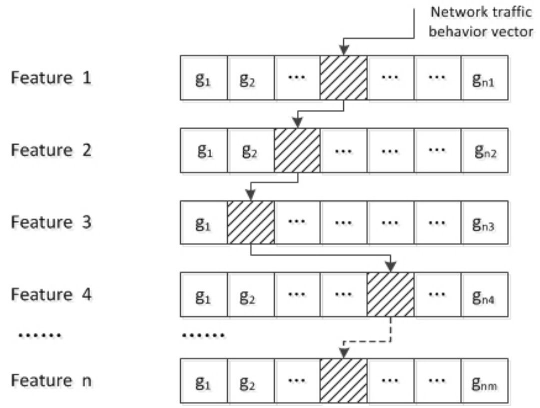

Figure 2: The structure of grid cells on each dimension

phase, the data space is partitioned into grid cells where each network traffic behavior can be recorded according to the value of each feature on the corresponding dimension. After all network traffic behaviors are recorded, including both the malicious behaviors and the normal ones, the grid partitions of malicious behaviors which differ from the normal behaviors are extracted and further form the feature graph in the second step of training phase. A network traffic behavior whose corresponding values of features are involved in the nodes of the graph, is identified as malicious behavior. Whenever there exist malicious behaviors of new type, the nodes of corresponding grid cells which record new malicious behaviors are added to the graph.

3.1 The first step of training phase

Generally, a network traffic behavior can be represented as a vector in feature space. For each vector, the value on each dimension denotes the magnitude of the corresponding feature of the behavior. Thus, the vector of a network traffic behavior is described in definition 1, where V_i denotes the value on ith dimension of the vector and d is the total number of features.

Definition 3.1. The vector of a network traffic behavior:
$Vector\_of\_Behavior = \{V_1, V_2, ..., V_d\}$ is the vector whose value on each dimension denotes the magnitude of the corresponding feature in a network traffic behavior.

In the first phase, a strategy based on grid partition is adopted which is commonly employed in a kind of grid based clustering algorithms in data mining. The feature space is first partitioned into grid cells on each dimension, and then the vector of network traffic behavior can be mapped into the grid cells according to the dimensional values involved in the vector as shown in Fig. 2. After all training network traffic behavior vectors are mapped into the grid cells on each dimension, a graph of malicious network traffic behaviors can be extracted from the grid structure.

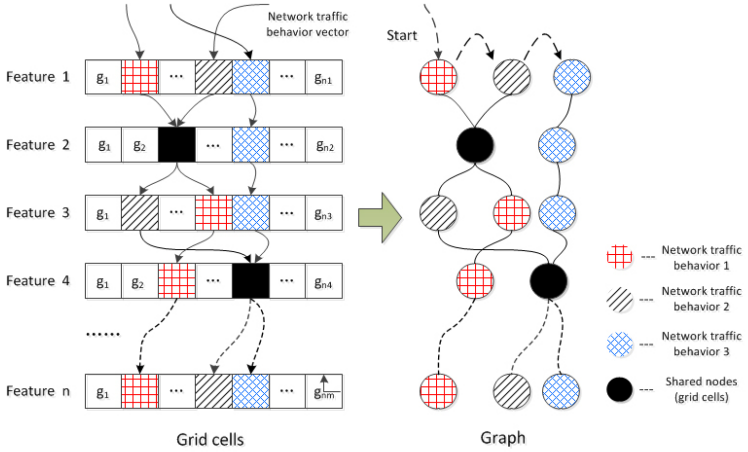
Network traffic behavior vector
Start
Feature 1
Feature 2
Feature 3
Feature 4
Feature n
g_1
g_2
g_{n1}
g_{n2}
g_{n3}
g_{n4}
g_{nm}
Network traffic behavior 1
Network traffic behavior 2
Network traffic behavior 3
Shared nodes (grid cells)
Grid cells
Graph

Figure 3: The feature graph

3.2 The second step of training phase

Based on the work of the first step, the grid cells, which are on the path of malicious vector, together with the malicious path are extracted. And the extracted grid cells serve as the nodes of the feature graph, the path between two nodes serves as the edge of the feature graph. As shown in Fig. 3, a graph of malicious network traffic behaviors can be obtained by extracting the grid cells which includes most malicious network traffic behaviors at a certain proportion. Similarly, the graph of normal network traffic behaviors or all network traffic behaviors can also be obtained in the same way.

To the graph of malicious network traffic behaviors, if a network traffic behavior to be detected can be mapped into a path which is from the top level (feature 1) to the bottom level (feature n), then it is detected as malicious.

3.3 Detection of malicious network traffic behaviors

Based on the training phase of IDBFG, a security model consists of training phase and detection phase is proposed as shown in Fig. 4. To justify whether a network traffic behavior is malicious or not, the network traffic behavior, represented as a vector, is first mapped into the corresponding grid cells on each dimension according to its value in the vector. Since a vector has only one value on each dimension, the behavior vector can be further represented as a series of grid cells according to the dimensional values. If the grid cells involved in a behavior vector can be exactly matched to the pattern of a malicious behavior which is represented as a chain of connected nodes at different levels in the feature graph, the behavior vector is identified as malicious network traffic behavior. The procedure of IDBFG is as described in Algorithm 1.

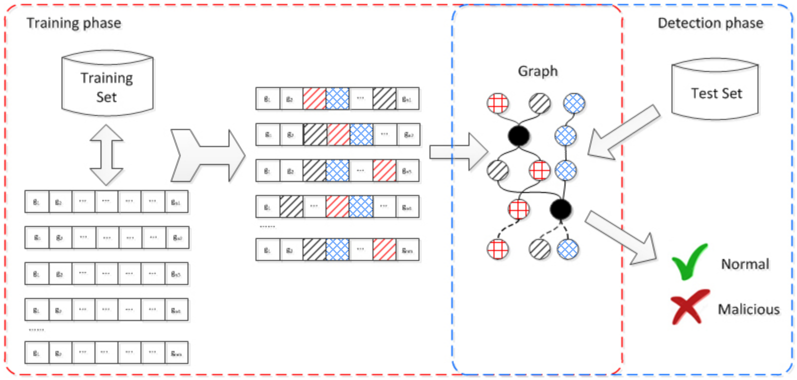

Figure 4: The security model

4 Experiments

4.1 The dataset

The experiments in the paper are based on the KDD-CUP'99 dataset, which is popular and widely used as standard dataset for intrusion detection. In KDD-CUP'99 dataset, 494020 samples are involved, each of which represents a network connection and each network connection is represented as a 41 dimensional feature vector. The 41 features are of various types, which include content type, intrinsic type and traffic type. Besides the normal network traffic behaviors, the remaining classes are of four different types of attacks, such as denial of service (Dos), user to root (U2R), probe and remote to local (R2L).

4.2 The baseline classifiers

Decision Tree is a widely used model for classification and prediction in machine learning, which can deal with not only numerical attribute but also the attribute of other types such as nominal attribute, binary attribute, etc. In addition, when processing large amount of data, it is reliable to get the feasible results with DT in a short period of time. The support vector machine is also one of the most widely used classification methods which applies the statistics of support vectors to categorize unlabeled data with high accuracy and efficiency.

Since SVM and DT have been proved to be excellent classifiers with high accuracy and have been widely employed as the baseline algorithms in lots of comparison tasks, we compare the performance of IDBFG, SVM and DT in the scenario that all malicious network traffic behaviors to be detected have already appeared in the training set.

4.3 The evaluation

In this paper, we choose the rates of detection, false alarm rate and run time as the evaluation index, which are widely employed for the experiments of intrusion detection. The rates of detection, false alarm can be calculated by Eq. (1) to Eq. (2), where TN (true negatives) denotes the number of normal behaviors correctly classified as normal, FN (false negatives)

Algorithm 1 IDBFG

```
Input: $Tr$ - Training set
       $\delta$ - The number of partitions
Output: $Graph$ - The feature graph
        $Grid$ - grid structure
Train_model($Tr$, $\delta$)
$Grid \leftarrow partition(\delta)$    %The grid structure are saved in $Grid$
for (each $tr$ in $Tr$)
   { map $tr$ into the corresponding grid cells in $Grid$
      if ($tr$ is malicious)
      { $g \leftarrow$ finds the grid cells of $tr$ in $Grid$
            $Graph \leftarrow Graph \bigcup g$ }
   }
Input: $Te$ - Test set
Output: $Mal$ - The set of malicious network behaviors in $Te$
        $Nor$ - The set of normal network behaviors
Detect_malicious($Te$)
for (each $t$ in $Te$)
   if ($t \in Graph$)
      $Mal \leftarrow Mal \bigcup t$
   else
      $Nor \leftarrow Nor \bigcup t$
```

Table 2: Confusion matrix

Actual \ Predicted	Normal	Malicious
Normal	True Negative(TN)	False Positive(FP)
Malicious	False Negative(FN)	True Positive(TP)

denotes the number of malicious behaviors falsely classified as normal, TP (true positives) denotes the number malicious behaviors correctly classified as malicious and FP (false positives) denotes the number of normal behaviors falsely classified as malicious as shown in Tab. 2. In addition, the scale of the grid granularity, which indirectly influences the performance of detection, is also evaluated.

$$Detection\ Rate = \frac{TP}{TP + FP} \tag{1}$$

$$False\ Alarm\ Rate = \frac{FP}{FP + TN} \tag{2}$$

4.4 Performance comparison with 19-dimensional features

In the experiments, we first compare the DR (detection rate) and FAR (false alarm rate) of IDBFG, SVM and DT with 19-dimensional features, which are based on the information gain ratio values of various features. As shown in Fig. 5, we evaluate the DR and FAR of the approaches in this paper with ROC curve, and our goal is to maximize DR and minimize FAR concurrently. In addition, the run time of IDBFG, SVM and DT is evaluated next. As the results shown in Fig. 5, when dealing with the KDD dataset with 19 dimensions involved, IDBFG and DT perform better than SVM. The results show that IDBFG provides higher detection rate and lower false alarm rate than SVM, and compared to IDBFG, DT provides nearly the same performance of DR and FAR. The results may be because SVM depends much on its statistics nature instead of the features of behavior when justifying a network traffic behavior is malicious or not, however, to IDBFG and DT, whether a network traffic behavior is detected as malicious just depends on the models they build.

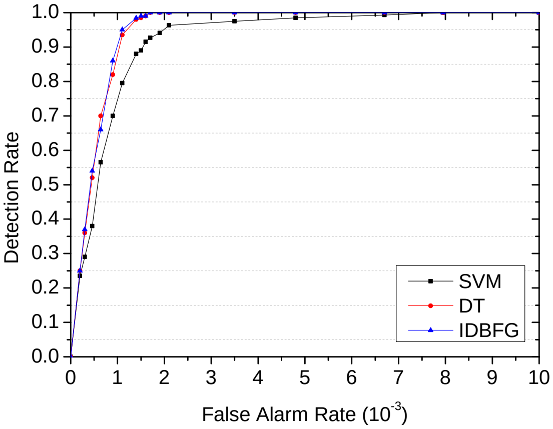

Figure 5: ROC performance of IDBFG, SVM and DT with 19 features

Although having the similar performance of DR and FAR, IDBFG is more efficient than DT as shown in Fig. 6, which can be explained as that the nodes in feature graph of IDBFG are far less than the nodes in the DT, which may lead to low computational effort. In addition, the run time of both IDBFG and DT are lower than that of SVM.

4.5 Performance comparison with all features

We next compare the DR, FAR and run time of IDBFG, SVM and DT with all features. As shown in Fig. 7, the DR and FAR of the approaches are evaluated by ROC curve respectively, and our goal is to maximize DR and minimize FAR concurrently. As the results shown in Fig. 7, when dealing with the KDD dataset with all features involved, the performance of IDBFG, DT and SVM deteriorate slightly. And the results show that both

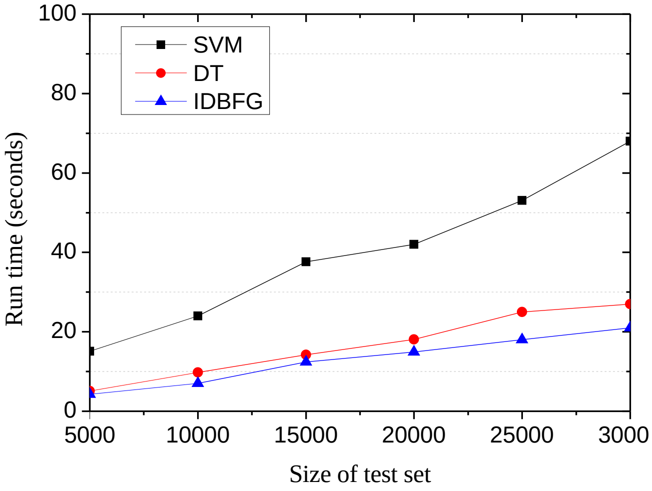

Figure 6: Run time of IDBFG, SVM and DT with 19 features

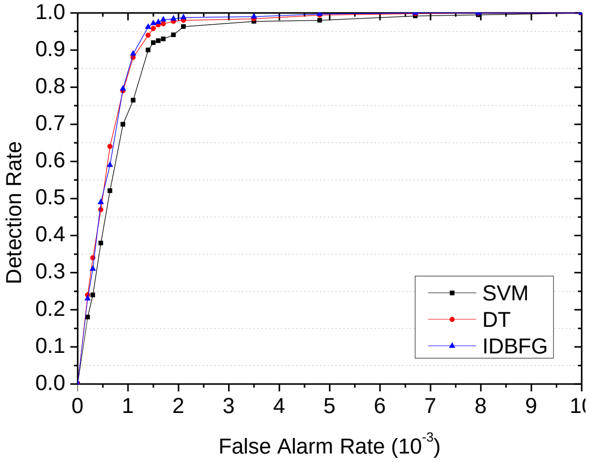

Figure 7: ROC performance of IDBFG, SVM and DT with all features

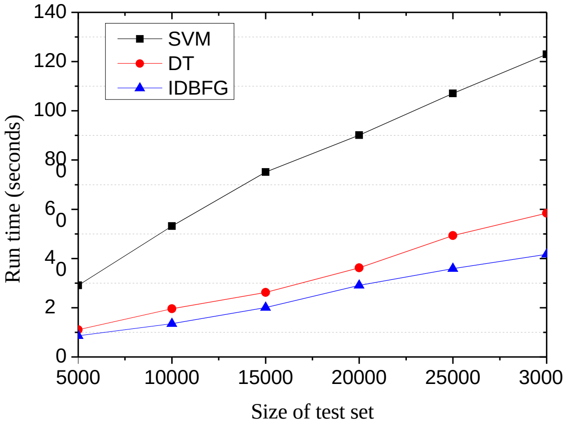

Figure 8: Run time of IDBFG, SVM and DT with all features

IDBFG and DT provide higher detection rate and lower false alarm rate than SVM, and DT provides nearly the same performance of DR and FAR as IDBFG does. The results are mainly because SVM depends much on its statistics nature instead of the features of behavior. Other than SVM, IDBFG and DT put more focus on the models they build.

Similar to the experimental results with 19-dimensional features, the run time of both IDBFG and DT with all features are lower than that of SVM, and IDBFG is more efficient than DT as shown in Fig. 8. The results can be explained as that only the nodes on the pathes of malicious behaviors are extracted to build feature graph which are far less than the nodes in the DT. Consequently, IDBFG needs lower computational effort than DT does.

4.6 Comparison of different grid granularity

Since the number of partitions on each dimension is determined by the partition parameter δ, which has great influence on the performance of IDBFG, in this experiment, the influence of partition parameter δ is discussed according to the performance of IDBFG with different partition parameter.

In this experiment, the number of partitions on each dimension, δ, is initially set to be the square root of n, where n denotes the total number of samples in training set and $\delta_0 = \sqrt{n}$. Then, we adjust δ by increasing or decreasing it at a certain ratio η, which is set to be 10%. When δ changes, the DR and FAR performance of IDBFG also change. As shown in Fig. (9), the performance of IDBFG is optimal at the point around $\delta = 0.7\delta_0$, where DR and FAR have almost the highest and lowest value respectively.

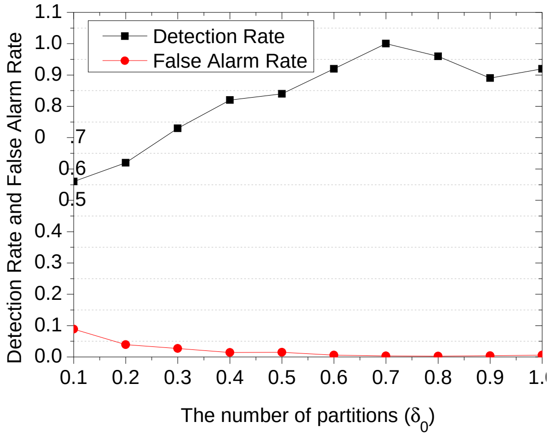

Figure 9: The DR and FAR performance of IDBFG with different grid granularity

5 Conclusion and future work

Up to now, very few intrusion detection approaches are based on graph structure. In this paper, we propose a new approach for intrusion detection which is based on IDBFG model, and the advantages of the approach are as follows:

1. Compared to normal network traffic behaviors, the proportion of malicious behaviors is relatively small, which leads to the high efficiency of building the graph of malicious behaviors.

2. The graph can be updated conveniently by extracting the grid cells involving malicious behaviors from the grid structure.

3. When detecting malicious behaviors, the proposed approach is efficient and effective according to the results of experiments in different scenarios.

Since malicious network traffic behaviors are of various, it is impossible to identify all malicious behaviors. Actually, the malicious network traffic behaviors, which have not been recorded or have not even appeared before, still threaten the security of various systems. Once this kind of malicious network traffic behaviors appear, it is necessary to detect them timely and take corresponding measures.

Hence, our future work will focus on intrusion detection in two directions. First, try to improve the efficiency and effectiveness of IDBFG on malicious network traffic behaviors detection. Second, try to detect the malicious network traffic behaviors that have never appeared before.

Acknowledgement: This research was funded in part by the National Natural Science Foundation of China (61871140, 61872100, 61572153, U1636215, 61572492, 61672020), the National Key research and Development Plan (Grant No. 2018YFB0803504), and Open Fund of Beijing Key Laboratory of IOT Information Security Technology (J6V0011104).

References

Agarwal, M.; Purwar, S.; Biswas, S.; Nandi, S. (2017): Intrusion detection system for ps-poll dos attack in 802.11 networks using real time discrete event system. *IEEE/CAA Journal of Automatica Sinica*, vol. 4, no. 4, pp. 792-808.

Ashraf, Q. M.; Habaebi, M. H. (2015): Autonomic schemes for threat mitigation in internet of things. *Journal of Network and Computer Applications*, vol. 49, pp. 112-127.

Baig, Z. A.; Sait, S. M.; Shaheen, A. (2013): Gmdh-based networks for intelligent intrusion detection. *Engineering Applications of Artificial Intelligence*, vol. 26, no. 7, pp. 1731-1740.

Chen, J.; Tian, Z.; Cui, X.; Yin, L.; Wang, X. (2018): Trust architecture and reputation evaluation for internet of things. *Journal of Ambient Intelligence and Humanized Computing*, pp. 1-9.

Chen, Y.; Abraham, A.; Yang, B. (2007): Hybrid flexible neural-tree-based intrusion detection systems. *International Journal of Intelligent Systems*, vol. 22, no. 4, pp. 337-352.

Cheng, R.; Xu, R.; Tang, X.; Sheng, V. S.; Cai, C. (2018): An abnormal network flow feature sequence prediction approach for ddos attacks detection in big data environment. *Computers, Materials & Continua*, vol. 55, no. 1, pp. 95-119.

De la Hoz, E.; de la Hoz, E.; Ortiz, A.; Ortega, J.; Martínez-Álvarez, A. (2014): Feature selection by multi-objective optimisation: application to network anomaly detection by hierarchical self-organising maps. *Knowledge-Based Systems*, vol. 71, pp. 322-338.

Depren, O.; Topallar, M.; Anarim, E.; Ciliz, M. K. (2005): An intelligent intrusion detection system (ids) for anomaly and misuse detection in computer networks. *Expert Systems with Applications*, vol. 29, no. 4, pp. 713-722.

Du, X.; Xiao, Y.; Guizani, M.; Chen, H. H. (2007): An effective key management scheme for heterogeneous sensor networks. *Ad Hoc Networks*, vol. 5, no. 1, pp. 24-34.

Feng, W.; Zhang, Q.; Hu, G.; Huang, J. X. (2014): Mining network data for intrusion detection through combining svms with ant colony networks. *Future Generation Computer Systems*, vol. 37, pp. 127-140.

Gaffney, J. E.; Ulvila, J. W. (2001): Evaluation of intrusion detectors: a decision theory approach. *Proceedings of the 2001 IEEE Symposium on Security and Privacy*, pp. 50-61.

Ganapathy, S.; Kulothungan, K.; Muthurajkumar, S.; Vijayalakshmi, M.; Yogesh, P. et al. (2013): Intelligent feature selection and classification techniques for intrusion detection in networks: a survey. *EURASIP Journal on Wireless Communications and Networking*, vol. 2013, no. 1, pp. 271.

Huang, X.; Ye, Y.; Xiong, L.; Wang, S.; Yang, X. (2016): Clustering time-stamped data using multiple nonnegative matrices factorization. *Knowledge-Based Systems*, vol. 114, pp. 88-98.

Islam, M. N.; Seera, M.; Loo, C. K. (2017): A robust incremental clustering-based facial feature tracking. *Applied Soft Computing*, vol. 53, pp. 34-44.

Jeya, P. G.; Ravichandran, M.; Ravichandran, C. (2012): Efficient classifier for r2l and u2r attacks. *International Journal of Computer Applications*, vol. 45, no. 21, pp. 28-32.

Julisch, K. (2003): Clustering intrusion detection alarms to support root cause analysis. *ACM Transactions on Information and System Security*, vol. 6, no. 4, pp. 443-471.

Kang, J.; Oh, S. (2012): Anomaly intrusion detection based on clustering a data stream. *International Journal of Future Computer and Communication*, vol. 1, no. 1, pp. 17-20.

Khan, L.; Awad, M.; Thuraisingham, B. (2007): A new intrusion detection system using support vector machines and hierarchical clustering. *VLDB Journal*, vol. 16, no. 4, pp. 507-521.

Kim, G.; Lee, S.; Kim, S. (2014): A novel hybrid intrusion detection method integrating anomaly detection with misuse detection. *Expert Systems with Applications*, vol. 41, no. 4, pp. 1690-1700.

Li, M.; Sun, Y.; Jiang, Y.; Tian, Z. (2018): Answering the min-cost quality-aware query on multi-sources in sensor-cloud systems. *Sensors*, vol. 18, no. 12, pp. 4486.

Li, Y.; Guo, L. (2007): An active learning based tcm-knn algorithm for supervised network intrusion detection. *Computers & Security*, vol. 26, no. 7, pp. 459-467.

Lin, P. C.; Lee, J. H. (2013): Re-examining the performance bottleneck in a nids with detailed profiling. *Journal of Network and Computer Applications*, vol. 36, no. 2, pp. 768-780.

Lin, S. W.; Ying, K. C.; Lee, C. Y.; Lee, Z. J. (2012): An intelligent algorithm with feature selection and decision rules applied to anomaly intrusion detection. *Applied Soft Computing*, vol. 12, no. 10, pp. 3285-3290.

Lin, W. C.; Ke, S. W.; Tsai, C. F. (2015): Cann: an intrusion detection system based on combining cluster centers and nearest neighbors. *Knowledge-Based Systems*, vol. 78, pp. 13-21.

Liu, G.; Yi, Z.; Yang, S. (2007): A hierarchical intrusion detection model based on the pca neural networks. *Neurocomputing*, vol. 70, no. 7, pp. 1561-1568.

Milenkoski, A.; Vieira, M.; Kounev, S.; Avritzer, A.; Payne, B. (2015): Evaluating computer intrusion detection systems: A survey of common practices. *ACM Computing Surveys*, vol. 48, no. 1, pp. 1-41.

Modi, C. N.; Patel, D. (2013): A novel hybrid-network intrusion detection system (h-nids) in cloud computing. *Proceedings of the IEEE Symposium on Computational Intelligence in Cyber Security*, pp. 23-30.

Nasr, K.; Abou-El Kalam, A.; Fraboul, C. (2012): Performance analysis of wireless intrusion detection systems. *International Conference on Internet and Distributed Computing Systems*, pp. 238-252.

Özyer, T.; Alhajj, R.; Barker, K. (2007): Intrusion detection by integrating boosting genetic fuzzy classifier and data mining criteria for rule pre-screening. *Journal of Network and Computer Applications*, vol. 30, no. 1, pp. 99-113.

Pan, S.; Morris, T.; Adhikari, U. (2015): Developing a hybrid intrusion detection system using data mining for power systems. *IEEE Transactions on Smart Grid*, vol. 6, no. 6, pp. 3104-3113.

Peddabachigari, S.; Abraham, A.; Grosan, C.; Thomas, J. (2007): Modeling intrusion detection system using hybrid intelligent systems. *Journal of Network and Computer Applications*, vol. 30, no. 1, pp. 114-132.

Qiu, J.; Chai, Y.; Liu, Y.; Gu, Z.; Li, S. et al. (2018): Automatic non-taxonomic relation extraction from big data in smart city. *IEEE Access*, vol. 6, pp. 74854-74864.

Qiu, J.; Qi, L.; Wang, J.; Zhang, G. (2017): A hybrid-based method for chinese domain lightweight ontology construction. *International Journal of Machine Learning and Cybernetics*, pp. 1-13.

Qiu, J.; Xing, Z.; Zhu, C.; Lu, K.; He, J. et al. (2019): Centralized fusion based on interacting multiple model and adaptive kalman filter for target tracking in underwater acoustic sensor networks. *IEEE Access*, vol. 7, pp. 25948-25958.

Saad, R. M.; Manickam, S.; Ramadass, S. (2013): Utilizing data mining approches in the detection of intrusion in ipv6 network: review & analysis. *International Journal on Network Security*, vol. 4, no. 1, pp. 35-39.

Sarasamma, S. T.; Zhu, Q. A.; Huff, J. (2005): Hierarchical kohonenen net for anomaly detection in network security. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 35, no. 2, pp. 302-312.

Shin, S.; Lee, S.; Kim, H.; Kim, S. (2013): Advanced probabilistic approach for network intrusion forecasting and detection. *Expert Systems with Applications*, vol. 40, no. 1, pp. 315-322.

Shiravi, A.; Shiravi, H.; Tavallaee, M.; Ghorbani, A. A. (2012): Toward developing a systematic approach to generate benchmark datasets for intrusion detection. *Computers & Security*, vol. 31, no. 3, pp. 357-374.

Shittu, R.; Healing, A.; Ghanea-Hercock, R.; Bloomfield, R.; Rajarajan, M. (2015): Intrusion alert prioritisation and attack detection using post-correlation analysis. *Computers & Security*, vol. 50, pp. 1-15.

Shon, T.; Kovah, X.; Moon, J. (2006): Applying genetic algorithm for classifying anomalous tcp/ip packets. *Neurocomputing*, vol. 69, no. 16, pp. 2429-2433.

Shon, T.; Moon, J. (2007): A hybrid machine learning approach to network anomaly detection. *Information Sciences*, vol. 177, no. 18, pp. 3799-3821.

Sindhu, S. S. S.; Geetha, S.; Kannan, A. (2012): Decision tree based light weight intrusion detection using a wrapper approach. *Expert Systems with Applications*, vol. 39, no. 1, pp. 129-141.

Subaira, A. S.; Anitha, P. (2014): Efficient classification mechanism for network intrusion detection system based on data mining techniques: a survey. *IEEE 8th International Conference on Intelligent Systems and Control*, pp. 274-280.

Sultana, A.; Jabbar, M. (2016): Intelligent network intrusion detection system using data mining techniques. *2nd International Conference on Applied and Theoretical Computing and Communication Technology*, pp. 329-333.

Tajbakhsh, A.; Rahmati, M.; Mirzaei, A. (2009): Intrusion detection using fuzzy association rules. *Applied Soft Computing*, vol. 9, no. 2, pp. 462-469.

Tan, Q.; Gao, Y.; Shi, J.; Wang, X.; Fang, B. et al. (2018): Towards a comprehensive insight into the eclipse attacks of tor hidden services. *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 1584-1593.

Tian, Z.; Cui, Y.; An, L.; Su, S.; Yin, X. et al. (2018): A real-time correlation of host-level events in cyber range service for smart campus. *IEEE Access*, vol. 6, pp. 35355-35364.

Tian, Z.; Gao, X.; Su, S.; Qiu, J.; Du, X. et al. (2019): Evaluating reputation management schemes of internet of vehicles based on evolutionary game theory. arxiv:1902.04667.

Tian, Z.; Shi, W.; Wang, Y.; Zhu, C.; Du, X. et al. (2019): Real time lateral movement detection based on evidence reasoning network for edge computing environment. arxiv:1902.04387.

Tian, Z.; Su, S.; Shi, W.; Du, X.; Guizani, M. et al. (2019): A data-driven method for future internet route decision modeling. *Future Generation Computer Systems*.

Tong, X.; Wang, Z.; Yu, H. (2009): A research using hybrid rbf/elman neural networks for intrusion detection system secure model. *Computer Physics Communications*, vol. 180, no. 10, pp. 1795-1801.

Upadhyaya, D.; Jain, S.; HOD, C.; KIT, K. (2012): Model for intrusion detection system with data mining. *International Journal of Advanced Research in Computer Engineering and Technology*, vol. 1, no. 4, pp. 145-148.

Venkatesan, R.; Ganesan, R.; Selvakumar, A. A. L. (2012): A survey on intrusion detection using data mining techniques. *International Journal of Computers & Distributed Systems*, vol. 2, no. 1.

Wang, B.; Gu, X.; Ma, L.; Yan, S. (2017): Temperature error correction based on bp neural network in meteorological wireless sensor network. *International Journal of Sensor Networks*, vol. 23, no. 4, pp. 265-278.

Wang, B.; Gu, X.; Yan, S. (2018): Stcs: a practical solar radiation based temperature correction scheme in meteorological wsn. *International Journal of Sensor Networks*, vol. 28, no. 1, pp. 22-33.

Wang, G.; Hao, J.; Ma, J.; Huang, L. (2010): A new approach to intrusion detection using artificial neural networks and fuzzy clustering. *Expert Systems with Applications*, vol. 37, no. 9, pp. 6225-6232.

Wang, Y.; Tian, Z.; Zhang, H.; Su, S.; Shi, W. (2018): A privacy preserving scheme for nearest neighbor query. *Sensors*, vol. 18, no. 8, pp. 2440.

Wang, Z.; Liu, C.; Qiu, J.; Tian, Z.; Cui, X. et al. (2018): Automatically traceback rdp-based targeted ransomware attacks. *Wireless Communications and Mobile Computing*, vol. 2018.

Wu, X.; Zhang, C.; Zhang, R.; Wang, Y.; Cui, J. (2018): A distributed intrusion detection model via nondestructive partitioning and balanced allocation for big data. *Computers, Materials & Continua*, vol. 56, no. 1, pp. 61-72.

Xiang, C.; Yong, P. C.; Meng, L. S. (2008): Design of multiple-level hybrid classifier for intrusion detection system using bayesian clustering and decision trees. *Pattern Recognition Letters*, vol. 29, no. 7, pp. 918-924.

Yu, X.; Tian, Z.; Qiu, J.; Jiang, F. (2018): A data leakage prevention method based on the reduction of confidential and context terms for smart mobile devices. *Wireless Communications and Mobile Computing*, vol. 2018, pp. 1-11.

Zhang, C.; Jiang, J.; Kamel, M. (2005): Intrusion detection using hierarchical neural networks. *Pattern Recognition Letters*, vol. 26, no. 6, pp. 779-791.