SVM-DT-Based Adaptive and Collaborative Intrusion Detection

Shaohua Teng, *Member, IEEE*, Naiqi Wu, *Senior Member, IEEE*, Haibin Zhu, *Senior Member, IEEE*, Luyao Teng, and Wei Zhang

***Abstract*—As a primary defense technique, intrusion detection becomes more and more significant since the security of the networks is one of the most critical issues in the world. We present an adaptive collaboration intrusion detection method to improve the safety of a network. A self-adaptive and collaborative intrusion detection model is built by applying the Environments-classes, agents, roles, groups, and objects (E-CARGO) model. The objects, roles, agents, and groups are designed by using decision trees (DTs) and support vector machines (SVMs), and adaptive scheduling mechanisms are set up. The KDD CUP 1999 data set is used to verify the effectiveness of the method. The experimental results demonstrate the feasibility and efficiency of the proposed collaborative and adaptive intrusion detection method. Also, the proposed method is shown to be more predominant than the methods that use a set of single type support vector machine (SVM) in terms of detection precision rate and recall rate.**

***Index Terms*—Adaptive and collaborative, intrusion detection, decision tree (DT), support vector machines (SVM).**

I. Introduction

INTRUSION detection is an important means to guarantee the safety of a network to avoid illegal operations that are launched by intruders (such as attackers and hackers) via authentication identification [1]. An intrusion detection system (IDS) is the most significant tool to ensure the security of a network by analyzing the audit data and current state. There are many measures to protect a network system, however, most of the conventional methods are inefficient. Since some attacks are composed of a series of users' operations, the users' behavior should be analyzed to detect an intrusion. To do so, users' actions are divided into normal and abnormal ones to separate the data. Then, classification is used to justify the detection result.

With the explosive growth of transmission data and the wide application of high speed network, traditional intrusion detection methods are out of date and cannot meet the current requirements. Furthermore, an intrusion detection system should not affect the normal operation of a network system when it works, especially in the Big Data and high-speed network environment.

Support vector machines (SVMs) [2] are a powerful tool for machine learning which is widely utilized in many applications such as classification, intrusion detection, and pattern recognition. Since the current network is very complicated, an intrusion detection system needs to not only be an effective detection tool but also possess an adaptive mechanism. This work proposes an adaptive collaboration intrusion detection method and develops a corresponding intrusion detection model. The algorithms of SVMs and decision trees (DTs) are used in the model.

To build an effective and adaptive intrusion detection model, we introduce the environments-classes, agents, roles, groups, and objects (E-CARGO) model as a tool which helps us to design the detection system. Also, SVM classifiers and the DT algorithm are applied. Group role assignment is studied. Experiments on data set KDD CUP 1999 are done to illustrate the effectiveness and performance of the proposed method. The experimental results show that the proposed method can not only improve the accuracy of classification, but also save time and storage space.

The remainder of this paper is arranged as follows. Section II briefly reviews the related work. Section III introduces the E-CARGO model and intrusion detection. The major work is developed in Section IV, including the adaptive and collaborative intrusion detection method, its architecture and components. Section V exhibits the suspicious behavior detector based on SVMs and DTs. We build the whole model containing suspicious behavior detection roles and intrusion agents. Section VI shows the self-adaptive mechanism of cooperative intrusion detectors. We illustrate our work with experiments in Section VII. Section VIII summarizes the main contribution of this work.

II. Related Work

An SVM is a binary classifier and is applied to intrusion detection by many researchers [3]−[16].

Manuscript received August 14, 2016; accepted December 12, 2016. This work is supported in part by the National Natural Science Foundation of China (61772141, 61673123), Guangdong Provincial Science & Technology Project (2015B090901016, 2016B010108007), Guangdong Education Department Project (Guangdong Higher Education letter 2015[133]), the Guangzhou Science & Technology Project (201508010067, 201604020145, 201604046017, and 2016201604030034). Recommended by Associate Editor Mengchu Zhou. *(Corresponding author: Wei Zhang.)*

Citation: S. H. Teng, N. Q. Wu, H. B. Zhu, L. Y. Teng, and W. Zhang, "SVM-DT-based adaptive and collaborative intrusion detection," *IEEE/CAA J. of Autom. Sinica*, vol. 5, no. 1, pp. 108−118, Jan. 2018.

S. H. Teng and W. Zhang are with the School of Computer Science and Technology, Guangdong University of Technology, Guangzhou 510006, China (e-mails: shteng@gdut.edu.cn; weizhang@gdut.edu.cn).

N. Q. Wu is with the Institute of Systems Engineering, Macau University of Science and Technology, Macau 999078, China, and also with the school of Electro-Mechanical Engineering, Guangdong University of Technology, Guangzhou 510006, China (e-mail: nqwu@must.edu.mo).

H. B. Zhu is with the Collaborative Systems Laboratory, Nipissing University, North Bay, Ontario PICILZ, Canada (e-mail: haibinz@nipissingu.ca).

L. Y. Teng is with the College of Engineering and Science, Victoria University, Ballarat Rd, Footscray VIC 3011, Australia (e-mail: luyaot@student.unimelb.edu.au).

Digital Object Identifier 10.1109/JAS.2017.7510730

Teng *et al.* [3] introduce a fuzzy SVM into intrusion detection and propose a collaborative network intrusion detection model. It is a multi-agent model including three kinds of agents which are used to detect transmission control protocol (TCP) attacks, user datagram protocol (UDP) attacks, and internet control message protocol (ICMP) attacks, respectively. Each agent plays a role that is realized by a fuzzy SVM.

Kuang *et al.* [4] propose an SVM model for intrusion detection. This model is formed by a multi-layer SVM classifier by combining kernel principal component analysis (KPCA) with genetic algorithm (GA). By this method, KPCA is used as a preprocessor with a GA algorithm being embedded. With the KPCA, the dimension of feature vectors can be reduced and the training time can be shortened, while the GA algorithm is employed to optimize the kernel parameter σ, the punishment factor C, and the tube size ε of the SVM.

Li *et al.* [5] present a classifier for intrusion detection, which combines clustering, ant colony algorithm, and an SVM. This classifier can identify whether a network visit is normal or not. Bamakan *et al.* [6] use time-varying chaos particle swarm optimization and propose an intrusion detection framework based on multiple criteria linear programming (MCLP) and SVM. A weighted objective function is designed and implemented, and a particle swarm optimization (PSO) algorithm is introduced for searching the optimum. Aburomman and Reaz [7] use a PSO algorithm to calculate the weights of classifiers and generate an ensemble model for detecting intrusions. They use the local unimodal sampling approach as a meta-optimizer to obtain better parameters for PSO.

Lin *et al.* [8] propose an intelligent anomaly intrusion detection algorithm with feature selection and decision rules. The algorithm is formed by integrating SVM, DT, and simulated annealing (SA). By this algorithm, the DT and SA are used to produce decision rules for new attacks and improve the accuracy of classification, while the SVM and SA are used to find the best candidate features for detecting anomaly attacks. The best parameter settings for the DT and SVM are adjusted by SA.

Feng *et al.* [9] introduce SVMs and clustering into intrusion detection based on self-organized ant colony network (CSOACN). Classifiers are generated by combining two existing machine learning methods: SVMs and CSOACN. They illustrate the detection effectiveness by experiments.

Kim *et al.* [10] propose a hybrid intrusion detection model by hierarchically integrating a misuse detector and an anomaly detector in a decomposition structure. The C4.5 DT is used to build the misuse detector that divides the normal data into smaller subsets. Then, SVMs are used to create an anomaly detector in each decomposed region. With this integration, the anomaly detector can indirectly use the known attack information to enhance its detection ability when profiles of normal behaviors are built. Horng *et al.* [11], Ravale *et al.* [12] combine a hierarchical clustering algorithm and the SVM technique, and build an SVM-based intrusion detection model. This hierarchical cluster provides the SVM with a smaller, abstracted, and higher-qualified data set that is derived from the original data set. This method not only greatly shortens the modeling time, but also improves the performance of the resultant SVM.

Lin *et al.* [13] develop intrusion detection techniques by extracting a feature representation approach to the malicious network traffic data. They present a method for how to extract more representative features for normal connections and effective detection of attacks. In their paper, two distances are measured. One is the distance between each sample and its cluster center, and the other is that between the data and its nearest neighbor in the same cluster.

Mitrokotsa *et al.* [14], [15] study the intrusion detection problem in mobile ad-hoc networks (MANET) by evaluating the performance of classification methods. They select varied traffic conditions and mobility patterns, then examine the classifier's performances on a database. Catania *et al.* [16] focus on how to deal with imbalanced data and present an approach for autonomously labeling the routine traffic based on an SVM. These labeling processes are described in SNORT [1].

There are some limitations of SVMs, such as high dependency on parameters, huge number of support vectors in the calculating process, and a long training time. In many cases, an SVM model does not behave flexibly as we need. Hence, we cannot directly apply this method in different network scales. Especially, traditional intrusion detection methods and technologies are difficult to utilize in the current high-speed network environment with tremendous amounts of data. For intrusion classification in complicated network environments, we put forward an adaptive cooperation method based on the E-CARGO model and test it with the KDD CUP 1999 data set.

Teng *et al.* [17] and Zhang *et al.* [18] introduce collaborative computing and granular computing into intrusion detection, and present a cooperative multi-agent intrusion detection model, where every agent plays a detector role and these agents form a distributed intrusion detection model. This model improves the performance of detecting intrusions. However, designing and implementing an agent are very complicated, and a distributed model needs to hold and manage a lot of agents, agents' actions, and their communications [17]–[19]. Note that the problem for building an intrusion detection model falls in the software engineering field. An E-CARGO model is a tool in software engineering and includes a set of software components. They are groups, roles, agents, environments, objects, and classes [20]–[22]. Therefore, an E-CARGO model is applied in the proposed method and its components are used to describe the architecture, the modules, and their relationship.

Role-based collaboration [20]–[22] uses roles as underlying mechanisms to realize abstraction, collaboration, allocation and assignment, and interaction. A role (r) in an E-CARGO model can be defined as a requirement and assigned to some intrusion detection agents. A current agent (a) is currently playing a detection role and the potential ones possess the detecting ability, but they are not currently playing that role. A group (g) which often contains many agents performs a function of an intrusion detection component. An environment (e) can include many roles and it confines a number range [l, u] for each role. The role with range [l, u] means that at least

l detection agents are required to play it and can be played by at most u detection agents. A group g is workable when each role in g has enough (l) agents to play it currently.

III. Intrusion Detection and E-CARGO

A common intrusion detection framework (CIDF) was proposed by the defense advanced research projects agency (DARPA) [1], [2], [23]. The CIDF divides an intrusion detection system into four components: event generator, detector, response unit, and database. These four components are used in our model.

In this paper, components and their collaboration are specified by environments, classes, agents, roles, groups, and objects in the E-CARGO model [17], [20]−[22]. In the E-CARGO, requirements are regarded as roles. All the intrusion detection components are taken as requirements such that they are defined as roles. A role r realizes a function about intrusion detection and is assigned to some agents that play role r. Agents that play the same role form a group g. g then composes detection results of the agents in the same group.

The three fundamental components in the E-CARGO are agent, role, and group [17], [20]−[22]. These components are defined as follows.

Definition 1 [20]−[22]: ***class***: $c ::= \langle n, D, F, X\rangle$, where

1) n is the ID of c;

2) D is a data structure which describes the state of an object;

3) F is a set of mapping functions or methods;

4) X is a unified interface of all the objects in the class, which is a set of message patterns that tells how to send a message to invoke a function.

Note that c represents a class. It can be an event generator class, an event detector class, or a response unit class. C is the set of all classes.

Definition 2 [20]−[22]: ***role***: $r ::= \langle n, I, N_a, N_o\rangle$, where

1) n is the ID of r;

2) $I ::= \langle M_{\text{in}}, M_{\text{out}}\rangle$ denotes a set of messages with M_{in} being the input messages and M_{out} the output messages;

3) N_a is an ID set of agents that are playing r;

4) N_o is an ID set of objects including classes, environments, roles, and groups which can be accessed by agents playing r.

Symbol r refers to a role and R represents the set of all the roles. There are three kinds of roles and they are event generator role EvGenRole, event detector role EvDetRole, and response unit role ResUnitRole. EvGenRole generates suspicious events that come from users' behaviors, EvDetRole detects attacks, and ResUnitRole makes a response according to the detected attacks.

An event generator role EvGenRole is defined as follows.

Definition 2' [1]: ***RoleEvGenRole***: $r ::= \langle n, I, N_a, N_o\rangle$, where

1) n is the ID of r;

2) $I ::= \langle M_{\text{in}}, M_{\text{out}}\rangle$ denotes a set of messages with M_{in} being the collected network messages and M_{out} output messages to role EvDetRole;

3) N_a is an ID set of agents that are playing EvGenRole r for generating suspicious events;

(4) N_o is an ID set of objects, including detecting environment, detecting roles, and detecting groups which can be accessed by agents playing EvGenRole r.

Others roles can be similarly defined as follows.

Definition 3 [20]−[22]: ***group***: $g ::= \langle n, e, J\rangle$, where

1) n is the ID of g;

2) e is an intrusion detection environment, all agents in g work under the same e;

3) J is a set of tuples for identifying an agent and role, i.e.,

$$J = \{\langle n_a, n_r, n_o\rangle | \exists q, n_o (n_o \in N_o) \wedge (\langle n_r, q, N_o\rangle \in e.B)\}.$$

In this paper, G denotes the set of all groups, and g is a specific group. There are two kinds of groups: big and small ones. A big group is denoted by BigGroup, while a small one is denoted by SmallGroup. There are three BigGroups and they are EvGenGroup, EvDetGroup, and ResUnitGroup. EvGenGroup is used to produce suspicious intrusion events, EvDetGroup detects these suspicious events, and ResUnitGroup responds to detection results.

A BigGroup contains some SmallGroups. In this paper, EvGenGroup contains four SmallGroups and they are sensor SmallGroup EvGenGroup$_{\text{sen}}$, decoding SmallGroup EvGenGroup$_{\text{decod}}$, filtering SmallGroup EvGenGroup$_{\text{filt}}$, and generating suspicious intrusion event SmallGroup EvGenGroup$_{\text{gen}}$. EvDetGroup has four SmallGroups and they are SmallGroup EvDetGroup$_{\text{tcp}}$ for detecting TCP attacks, SmallGroup EvDetGroup$_{\text{udp}}$ for detecting UDP attacks, SmallGroup EvDetGroup$_{\text{icmp}}$ for detecting ICMP attacks, and SmallGroup EvDetGroup$_{\text{cont}}$ for detecting contents. ResUnitGroup can have one or more groups according to network traffic.

Definition 4 [20]−[22]: ***agent***: $a ::= \langle n, c_a, s, N_r, N_{bg}, N_{sg}\rangle$, where

1) n is the ID of agent a;

2) c_a is a class describing the properties of agent a;

3) s is a data structure that includes attributes or states;

4) N_r denotes a set of roles that the agent is playing;

5) N_{bg} denotes a set of IDs of BigGroups;

6) N_{sg} denotes a set of IDs of SmallGroups.

It should be noticed that the potential abilities of an agent in a BigGroup are stronger than that of an agent in a SmallGroup.

There are two classes of agents. One belongs to a BigGroup and the other belongs to a SmallGroup. The former can play different roles in the same BigGroup and the latter can enact only the role in the same SmallGroup.

Definition 5 [20]−[22]: ***environment***: $e ::= \langle n, B\rangle$, where

(1) n is the ID of e;

(2) B is a set of tuples, i.e., $B = \{\langle n_r, q, N_o\rangle\}$, where n_r is the ID of a role, q is described by (l, u) and tells that how many agents can play this role in e and how many agents may play the same role r in g. Set N_o consists of the objects accessed by the agents that play r. $|N_o|$ denotes the number of resources visited by agents.

Here, e denotes an intrusion detection environment. The symbol q is a range, such as (3, 8), (10, 20), (7, 7), or (2, 60).

IV. Intrusion Detection Components

According to the CIDF, an intrusion detection system (IDS) is composed of an event generator, an event detector, a response unit, and an event database. Except the event database, the others are defined as three kinds of roles: EvGenRole, EvDetRole, and ResUnitRole. Role EvGenRole corresponds to the event generator, role EvDetRole corresponds to the suspicious event detector, and role ResUnitRole states the response unit.

A. Event Generator and Event Generation Role

The function of an event generator in CIDF is mainly composed of a sensor, a decoder, a filter, and an event generator. Because event generation role EvGenRole expresses functions of the event generator, it includes sensing, decoding, filtering, and generating suspicious events. Then, role EvGenRole is divided into role $\text{EvGenRole}_{\text{sen}}$, role $\text{EvGenRole}_{\text{decod}}$, role $\text{EvGenRole}_{\text{filt}}$, and role $\text{EvGenRole}_{\text{gen}}$. They are in charge of gathering network packages, decoding, filtering, and generating suspicious events, respectively. Their construction is shown in Fig. 1.

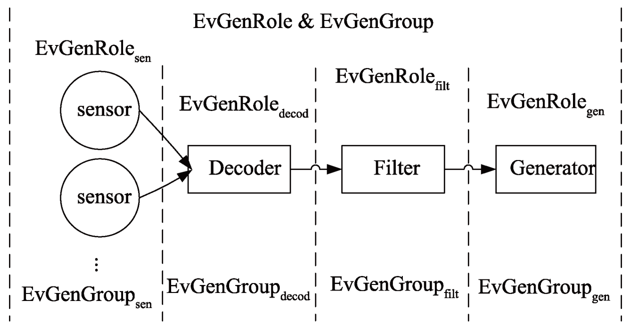


Fig. 1. Role, group about event generator.

In Fig. 1, every role is played by some agents. All of the agents that play the same role form a group. EvGenGroup contains four small groups: $\text{EvGenGroup}_{\text{sen}}$, $\text{EvGenGroup}_{\text{decod}}$, $\text{EvGenGroup}_{\text{filt}}$, and $\text{EvGenGroup}_{\text{gen}}$. Because agent a in $\text{EvGenGroup}_{\text{sen}}$ collects data from the Internet, its data structure includes all fields of the network packet. In this paper, a small group includes a role and its agents, e.g., $\text{EvGenGroup}_{\text{sen}}$ has many sensors.

B. Event Detector and Event Detection Role

The event detector in CIDF is designed to detect suspicious events generated by event generators. The function of the event detector is defined as role EvDetRole such that Role EvDetRole is responsible for detecting suspicious events.

Three kinds of suspicious event detector roles are generated according to the network protocols TCP, UDP, and ICMP. These three roles are role $\text{EvDetRole}_{\text{tcp}}$, role $\text{EvDetRole}_{\text{udp}}$, and role $\text{EvDetRole}_{\text{icmp}}$. They are responsible for detecting TCP suspicious events, UDP suspicious attacks, and ICMP suspicious attacks, respectively. In addition, role $\text{EvDetRole}_{\text{cont}}$ detects the content of the network packet extracted. The content feature denotes data content in a network packet, including a user's messages, the protocol of the application layer, the root user or administrator information, and the important data. The structure of event detectors is shown in Fig. 2.

In Fig. 2, EvDetGroup includes four SmallGroups: $\text{EvDetGroup}_{\text{tcp}}$, $\text{EvDetGroup}_{\text{udp}}$, $\text{EvDetGroup}_{\text{icmp}}$, and $\text{EvDetGroup}_{\text{cont}}$. The detection role is played by some detection agents and a SmallGroup is composed of agents that play the same detector role. For example, $\text{EvDetGroup}_{\text{tcp}}$ has many agents that detect TCP attacks.

The creation of event detectors is presented in Section V.

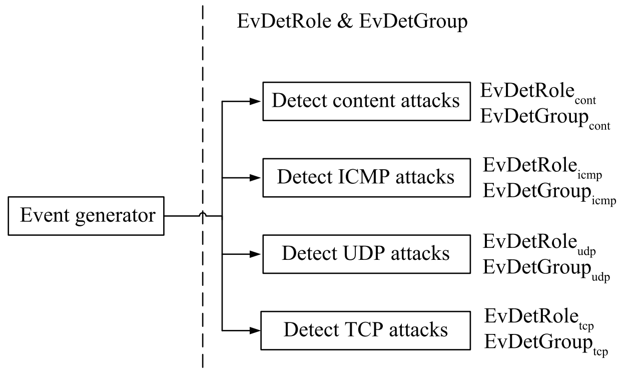


Fig. 2. Role, group about event detector.

C. Response Unit and Its Role

The response unit is a component in the CIDF and makes corresponding responses according to the detection results. Role ResUnitRole is defined as the requirement of the response unit. Many agents are assigned to play the role ResUnitRole.

D. Role and Group

The roles defined in this paper are listed in Fig. 3 and groups are given in Fig. 4.

E. The Collaboration Detection Architecture

A new collaboration intrusion detection framework is proposed in this section and a multi-agent cooperative detection model is designed and implemented. This model includes four components: event generators, event detectors, a response unit, and a database. An event generator contains a sensor, a decoder, a filter, and a generator. An event detector includes a TCP detector, a UDP detector, an ICMP detector, and a content detector. The adaptive collaboration detection model is shown in Fig. 5. It is abbreviated as CAIDM.

According to the E-CARGO model, these components are all taken as requirements and are defined as roles. There are three kinds of roles: event generator role EvGenRole, event detector role EvDetRole, and response unit role ResUnitRole. In Fig. 5, there are five kinds of detector roles: role $\text{EvDetRole}_{\text{tcp}}$, role $\text{EvDetRole}_{\text{udp}}$, role $\text{EvDetRole}_{\text{icmp}}$, role $\text{EvDetRole}_{\text{cont}}$, and role $\text{EvDetRole}_{\text{fc}}$. These roles are used to detect TCP attacks, UDP attacks, ICMP attacks, the content of network packages, and fusion detection, respectively. The resultant detection messages are sent to the response unit and the database.

Each role r is played by some agents and the agents that perform the same task (a role) form a small group. In order to cope with different network traffics and environments, three kinds of detecting scales are introduced in the system, i.e., small, middle, and large modes. These modes are used for diverse detection instances for various network traffics and an adaptive detection mechanism is required and implemented.

F. Role and Agent

There are three kinds of roles and they are EvGenRole, EvDetRole, and ResUnitRole.

Role EvGenRole is divided into role EvGenRole$_{\rm sen}$, role EvGenRole$_{\rm decod}$, role EvGenRole$_{\rm filt}$, and EvGenRole$_{\rm gen}$; Role EvDetRole falls into role EvDetRole$_{\rm tcp}$, role EvDetRole$_{\rm udp}$, role EvDetRole$_{\rm icmp}$, role EvDetRole$_{\rm cont}$, and role EvDetRole$_{\rm fc}$.

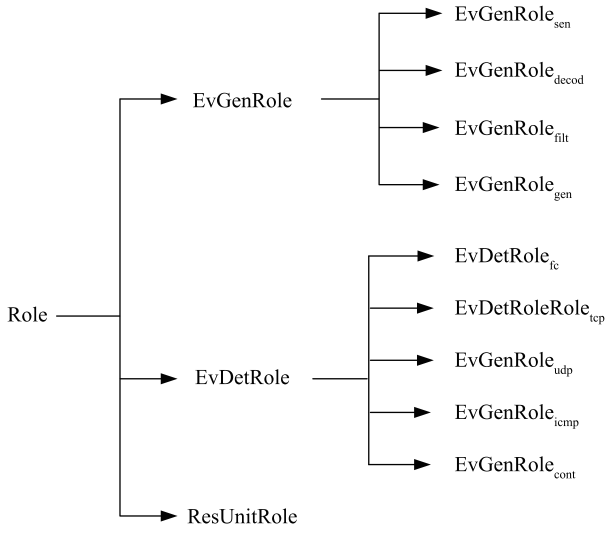


Fig. 3. Roles and their relations.

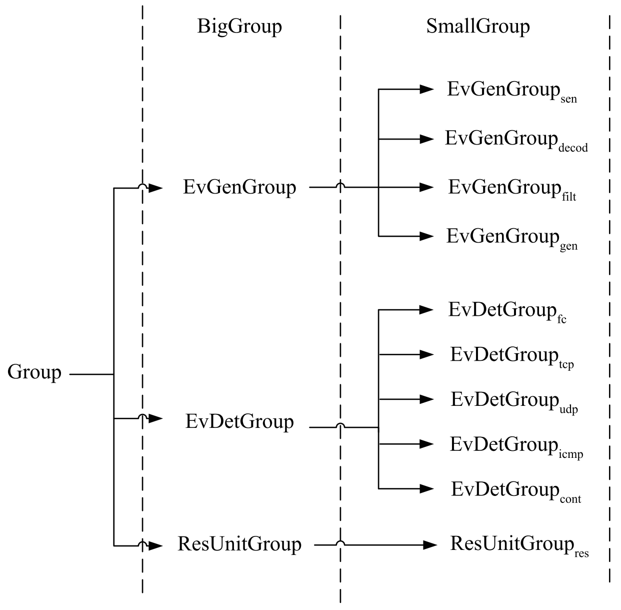


Fig. 4. Groups and their relations.

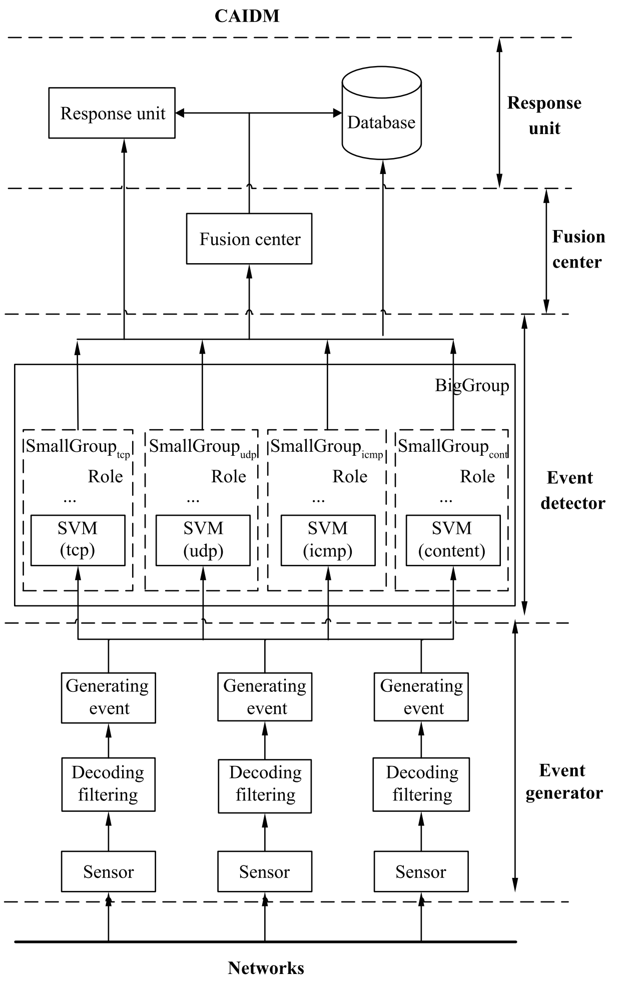


Fig. 5. The collaborative and adaptive intrusion detection model (CAIDM).

A role r can be assigned to some agents. Agents that play the same role r form a small group. For example, an EvGenRole$_{\rm sen}r_{\rm sen}$ is defined as for collecting data from network packet and all agents that play EvGenRole$_{\rm sen}$ $r_{\rm sen}$ must extract data from network packet. These agents form SmallGroup EvGenGroup$_{\rm sen}$ $g_{\rm sen}$.

In the next section, the procedure of creating TCP detector is discussed.

V. BUILDING INTRUSION DETECTOR

Because TCP attacks are more frequent than others, building TCP detector is discussed in this paper.

There are five classes of suspicious event detector roles which are used to detect TCP attacks, UDP attacks, ICMP attacks, and contents of packages, and synthesize detections in the fusion center, respectively. These suspicious event detector roles are described below. Agents are created with each of them being composed of SVMs and DT.

A. SVM and Multi-Class SVMs

There are two major categories [2] of SVMs for solving multi-class problems: 1) establishing a group of 2-class classifiers and 2) establishing a multi-class classifier. The former has many types, including 1-v-r (one-versus-rest), 1-v-1 (one-versus-one), directed acyclic graph (DAG) SVM (large interval

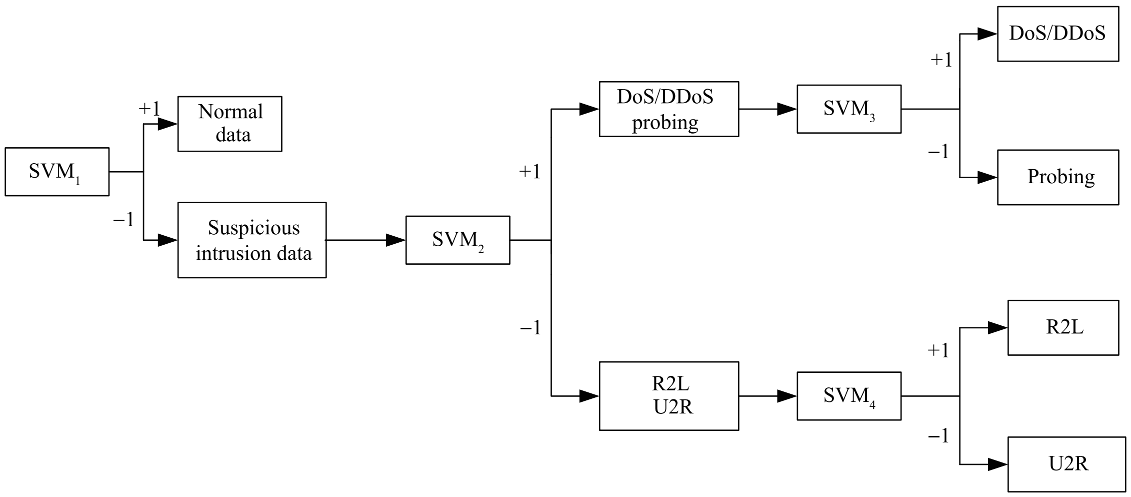


Fig. 6. The collaborative intrusion detector based on SVMs and DT for TCP.

multi-class SVM classifier based on directed acyclic graph), the binary SVM, etc.

In this paper, a multi-class classifier of the 1-v-r type is used to perform the intrusion detection as a detector. It needs to establish many 2-class classifiers. After network packets are decomposed into four parts corresponding to the TCP, UDP, ICMP protocols, and content, a group of 2-class classifiers based on SVMs and DTs are built to detect intrusions. Four 2-class classifiers are created to implement such a detector for TCP protocol.

B. Data and Suspicious Event Detector

A network data stream can be decomposed into three sets: the TCP data, the UDP data, and the ICMP data. Four kinds of detector roles are designed and implemented, and they are used to detect TCP, UDP, ICMP, and content-based attacks, respectively. According to the DT method, building these intrusion detectors includes two steps: modeling and testing. The preprocessed historical data (network data or experimental data) is decomposed into training data and testing data. The former (training data) is applied to produce the detection agents based on SVMs and DTs, and the latter (testing data) is used to assess these detection agents.

KDD CUP 1999 [3], [17], [18] decomposes attacks into four types, including probing, denial of service (DoS), remote-to-local (R2L), and the user-to-root (U2R). A probing attack acquires objective services provided by the object, including possible bugs, related system information, and so on [17], [18]. DoS attacks damage the target system and stop normal services by interrupting normal service delivery, causing a system collapse or shutdown [17], [18]. R2L and U2R are attacks that illegally promote users' privileges [17], [18]. The former obtains the permission to enter the target system by stealing user information. The latter upgrades one's permission by changing common users' privileges in the system or changing root authority. Thus, tasks for ensuring the security of a system are very complicated, especially when attacks are combined to form collaborative attacks. Each record in KDD CUP 1999 has 41 attributes, of which 34 attributes are continuous and seven are discrete. Before the experiments, the discrete attributes are converted into numerical ones by counting the frequency of their values. All data is turned into the standard and available format.

To build TCP detectors, a collaborative and adaptive intrusion detection model is designed based on SVMs and DTs and implemented in this text. A TCP detector has three layers and the detector at each layer can be defined as a role. Every role asks agents to perform it. Then, many 2-class SVM explorers are set up, and each 2-class SVM is a 1-to-rest classifier. The number of classifiers is related to the network traffics and the detection task load. Agents that detect TCP attacks are adaptive and are activated by the agent playing the manager role.

In order to make detection faster and more efficient, normal data is first separated from suspicious data. By doing so, one can reduce the amount of data to be further dealt with. In addition, we try to decrease the output of a 2-class SVM detector. Then, an optimized, cooperative, and adaptive intrusion detection model based on SVMs and DT is designed and implemented for a TCP detector as shown in Fig. 6. At each layer, a detecting role is required and we build a 2-class SVM or more to play the role. When the network traffic is heavy, many agents are required to perform the same event detector role at the same layer. These agents form a small group. In Fig. 6, SVM_1 decomposes the whole data set into normal and suspicious attack data, and performs the role at layer 1. At layer 2, SVM_2 separates the suspicious intrusion data into two parts. One of them includes DoS, DDoS, and probing attacks, and the other contains R2L and U2R attacks. Here, SVM_2 plays the role at layer 2. There are SVM3 and SVM_4 at layer 3. SVM_3 is assigned to detect probing and DoS/DDoS attacks, while SVM_4 is used to detect U2R and R2L attacks. Similarly, SVM_3 and SVM_4 play two roles at layer 3, respectively. An optimized DT-based detector is shown in Fig. 6.

C. Building SVM and DT-Based Detection Agent

Building a detection agent based on SVM needs to train and test it iteratively. After preprocessing, data is decomposed into four data sets corresponding to the network protocols TCP, UDP, ICMP, and content. Each data set is also divided into training and testing data.

Building a TCP detector is as follows. A group of SVM detectors is set up by using the training samples with the class label, i.e., four 2-class classifiers which are used to detect the normal data, the DoS/DDoS attacks, the probing attacks, and R2L or U2R attacks. In this paper, these detectors are all defined as roles in the E-CARGO model, and these roles require agents to play. When SVMs are used to implement these agents, some support vectors and the homologous parameters in the model should be brought out and be produced.

When testing, the testing data is delivered to four 2-class SVM classifiers according to TCP, UDP, ICMP protocols, and content, respectively. By comparing the classification results with testing data with the known class label, we can evaluate the detection accuracy of these SVM detectors. The SVM that meets the detection accuracy requirements is the expected detection agent. Hence, we build four 2-class SVMs: SVM_1, SVM_2, SVM_3, and SVM_4 as shown in Fig. 6.

VI. Adaptive Mechanisms

Due to different network environments, huge varieties of intrusions, high-speed traffics, and the big data generated from the network and so on, to be effective, an intrusion detection system should be flexible. The intrusion detection system needs to gather data from different networks and hosts. Moreover, it is required that the system should detect attacks on time and respond in a very short time. Because a single intrusion detector is difficult to provide enough security protection for a complicated network, a multi-agent collaborative and adaptive intrusion detection method is used to overcome the disadvantage of a single detector system.

In addition, to make such a system efficient within an acceptable cost, intrusion detection needs to be adaptive according to different network conditions. An adaptive mechanism of a collaborative intrusion detection model is proposed in this paper. All the components of CAIDM are defined as roles of the E-CARGO model and each role is assigned to some agents. The agents playing the same role form a group.

A. Local and Global Agent

Definition 6: Local agent: suppose p is an agent and belongs to a small group, then p is a local one and can play the role corresponding to this small group only.

Definition 7: Global agent: suppose p is an agent and belongs to a big group, then p is a global one and can play any role corresponding to the big group.

Here, two theorems are given below.

Theorem 1: Let p be an agent, SG is a small group and BG is a big group. Then, we have

1) $\forall p \in \text{EvGenGroup} \Rightarrow p \in \text{EvGenGroup}_{\text{sen}} \wedge p \in \text{EvGenGroup}_{\text{decod}} \wedge p \in \text{EvGenGroup}_{\text{filt}} \wedge p \in \text{EvGenGroup}_{\text{gen}}$.

That is:

p can play any role in EvGenRole.

2) $\forall p \in \text{EvDetGroup} \Rightarrow p \in \text{EvDetGroup}_{\text{tcp}} \wedge p \in \text{EvDetGroup}_{\text{udp}} \wedge p \in \text{EvDetGroup}_{\text{icmp}} \wedge p \in \text{EvDetGroup}_{\text{cont}}$.

That is:

p can perform any role in EvDetRole.

Theorem 2: Let q be an agent and suppose BG be a big group, SG_1 and SG_2 be two small groups. Then, if $\text{SG}_1 \neq \text{SG}_2$, we have

$$\forall q \in \text{SG}_1 \Rightarrow q \notin \text{SG}_2.$$

For example, $\forall a_2 \in \text{EvGenGroup}_{\text{sen}}$, we have $a_2 \notin \text{EvGenGroup}_{\text{decod}} \wedge a_2 \notin \text{EvGenGroup}_{\text{filt}} \wedge a_2 \notin \text{EvGenGroup}_{\text{gen}}$. The others are similar.

B. Network Traffic and Levels

For convenience, we set three different levels to detect attacks under the current network condition in this paper. In fact, because E-CARGO model is referred to our model, any scheduling can be applied by agent evaluation and role assignment.

There are three levels and they are called idle, normal, and busy level.

If the network condition is normal, we use a group of agents to detect attacks. If the network condition is busy, more agents than that in normal condition are generated to detect attacks. While the network condition is idle, fewer agents are required.

T_l and T_h are set as thresholds to define the levels. We have the definition below.

Definition 8: Network traffic: suppose T_l, T_h and X be real numbers, X denotes the network traffic, and $0 \leq \text{T}_\text{l} < \text{T}_\text{h}$, then,

1) Normal: The network traffic is normal, if $\text{T}_\text{l} \leq X < \text{T}_\text{h}$;

2) Idle: If $X < \text{T}_\text{l}$, then the network traffic is idle;

3) Busy: If $X \geq \text{T}_\text{h}$, then the network traffic is busy.

There are three scheduling cases in this paper and they are described by the following theorem.

Theorem 3. Suppose T_l, T_h and X be real number, X denotes the network traffic, and $0 \leq \text{T}_\text{l} < \text{T}_\text{h}$. We have

Case 1: When X is less than T_l, a few of roles are required, and a small scale scheduling is executed.

Case 2: When X is between T_l and T_h, a medium number of roles is required, and a middle scale scheduling is performed.

Case 3: When X is greater than T_h, plenty of roles are required, and a big scale scheduling is activated and carried out.

Therefore, network conditions and suspicious behaviors determine the number of agents being active at a layer, which makes it possible for the E-CARGO model to activate or shut down an agent independently, leading to a self-adaptive agent action.

C. Scheduling Policy

Because there are big and small groups, some agents belong to big groups, while some others belong to small ones. An adaptive scheduling plan is proposed as follows.

Let SG_1 denote a small group, BG_l a big group, SG_l-idle a set of agents that are idle, SR_l a role in SG_l, and a_1 an agent. Then, we have the scheduling policies as follows.

Policy 1: If SG_1 is busy, then

if $\exists a_1 \in \text{SG}_\text{l}$-idle and a_1 can play SR_l, then a_1 is assigned to SR_l and $a_1 \in \text{SG}_\text{l}$;

else if SG_l-idle$=\phi$, and $\exists\, a_1 \in \text{BG}_\text{l}$ and a_1 can play SR_l, then a_1 is assigned to SR_l and $a_1 \in \text{SG}_\text{l}$;

else no assignment happens.

Policy 2: If $\mathrm{SG_l}$ is not busy, then

if $\exists a_l \in \mathrm{SG_l}|\mathrm{BG_l}$, a_l is idle, then a_l is recovered, and $a_l \in$ $\mathrm{SG_l}$-idle or $a_l \in \mathrm{BG_l}$.

Policy 3: If q denotes aggregation requirements, $\mathrm{T_l}$ and $\mathrm{T_h}$ are two thresholds such that $0 \leq \mathrm{T_l} < \mathrm{T_h}$. We have:

1) if $q < \mathrm{T_l}$, then the small scale scheduling is performed;

2) if $\mathrm{T_l} \leq q < \mathrm{T_h}$, then the middle scale scheduling is executed; and

3) if $q \geq \mathrm{T_h}$, then the big scale scheduling is carried out.

D. EvGenRole and EvGenGroup

EvGenRole acts as an event generator role that requires agents to perform. EvGenRole contains a sensor role, a decoding role, a filtering role, and a generating event role. EvGenGroup denotes an active generator group and is a big group. It includes four members: $\mathrm{EvGenGroup_{sen}}$, $\mathrm{EvGenGroup_{decod}}$, $\mathrm{EvGenGroup_{filt}}$, and $\mathrm{EvGenGroup_{gen}}$. $\mathrm{EvGenGroup_{sen}}$ requires agents playing the sensor role, $\mathrm{EvGenGroup_{decod}}$ the decoding role, $\mathrm{EvGenGroup_{filt}}$ the filtering role, and $\mathrm{EvGenGroup_{gen}}$ the generating event role.

Some agents are global and belong to a big group EvGenGroup, while the others are local and belong to a small group. One agent in a small group must be in one of the four small groups: $\mathrm{EvGenGroup_{sen}}$, $\mathrm{EvGenGroup_{decod}}$, $\mathrm{EvGenGroup_{filt}}$, and $\mathrm{EvGenGroup_{gen}}$.

E. EvDetRole and EvDetGroup

EvDetRole is a suspicious event detector role, and EvDetRole includes four kinds of roles. Agents playing EvDetRole form a big group EvDetGroup. There are five detector small groups in big group EvDetGroup. These groups are $\mathrm{EvDetGroup_{tcp}}$, $\mathrm{EvDetGroup_{udp}}$, $\mathrm{EvDetGroup_{icmp}}$, $\mathrm{EvDetGroup_{cont}}$, and $\mathrm{EvDetGroup_{fc}}$. $\mathrm{EvDetGroup_{tcp}}$ holds agents that are in charge of detecting TCP attacks, $\mathrm{EvDetGroup_{udp}}$ owns agents that detect UDP attacks, $\mathrm{EvDetGroup_{icmp}}$ accommodates agents that find out ICMP attacks, $\mathrm{EvDetGroup_{cont}}$ contains agents that detect content attacks, and $\mathrm{EvDetGroup_{fc}}$ includes fusion agents that fuse the result of detectors.

F. Agent and Group

Agents for the same role form a small group. The suspicious behavior detector role is assigned to agents. Especially, Fig. 6 indicates that the data processed by group $\mathrm{SVM_1}$ is far more than that by group $\mathrm{SVM_2}$, the data processed by group $\mathrm{SVM_2}$ is more than that by group $\mathrm{SVM_3}$ or group $\mathrm{SVM_4}$, and the data processed by group $\mathrm{SVM_4}$ is the least. Therefore, we decompose the layers into three levels as listed in Fig. 6. From the top to bottom, each layer represents the data dealt with by groups $\mathrm{SVM_1}$, $\mathrm{SVM_2}$, $\mathrm{SVM_3}$, and $\mathrm{SVM_4}$, respectively. At the top, group $\mathrm{SVM_1}$ possesses the most number of agents for the largest gathered data. The next layer is group $\mathrm{SVM_2}$, then group $\mathrm{SVM_3}$ and group $\mathrm{SVM_4}$ is at the bottom.

The actions of agents are all adaptive. The group evaluation and role assignment are carried out by the scheduling process. The scheduling process executes the scheduling plan made in advance according to Definition 8 and the three policies.

Moreover, some agents are created to perform the response unit role. These agents implement the functions of the response unit role and fulfill the response tasks.

G. Role, Group, and Communication

Communications among components in CIDF are implemented based on groups. Communications are divided into three types that happen between two SGs, between an SG and a BG, and between two BGs. Communication between two SGs is from one small group to another small group, such as from SG $\mathrm{EvGenGroup_{sen}}$ to SG $\mathrm{EvGenGroup_{decod}}$, from SG $\mathrm{EvGenGroup_{decod}}$ to SG $\mathrm{EvGenGroup_{filt}}$, and from SG $\mathrm{EvGenGroup_{filt}}$ to SG $\mathrm{EvGenGroup_{gen}}$. The second type is from a small group to a big group or from a big group to a small group, such as from SG $\mathrm{EvGenGroup_{gen}}$ to BG EvDetGroup, or from BG EvDetGroup to SG $\mathrm{EvDetGroup_{tcp}}$. The last one is from a big group to another big group, such as from EvDetGroup to ResUnitGroup.

The data format of communication is user datagram in TCP/IP. The specification, rule, and mode of the communication are defined in the specification of roles, and the input and output specification of roles express the requirement of the communication. The input specification describes the criterion that a receiver receives and the output depicts the specification that a sender sends. All of the communications happen between two groups.

This kind of communication efficiently decreases transferring data among components. Hence, the traffic in IDS is reduced.

VII. Experiment and Result Analysis

The operating system of the experimental computer is Windows 7 with 64 bytes. The CPU is Intel(R) Core(TM) i3-2130 CPU @ 3.40 GHz, and its RAM is 4.00 GB. The experiment software is MATLAB 2012a.

A. Data Source and Feature Extraction

The experimental data is extracted from KDD CUP 1999 [3], [17], [18]. There are over 212960 records for testing, while there are over 467420 records for training. There are 24 types of attacks in the training set and 14 new attacks are added into the testing set to reinforce the effort. These total 38 attacks are divided into four categories according to their attack type: Probing, DoS, R2L, and U2R. A whole TCP session is regarded as a connection record. Each record contains four types of attribution collections: basic features, host-based traffic features, time-based traffic features, and content features.

In the training stage, the training data in the KDD CUP 1999 is divided into 46742 groups, each group possesses 10 records. A record is extracted from a group and is used as a training sample. The testing data is similar and 21296 groups are generated. 21296 records are extracted from the testing data. Especially, a multi-feature classification method is adopted to build TCP, UDP, ICMP, and content-based detection agents in this paper, since a single attribute is difficult to discriminate complicated attacks. We have the following:

1) 32 selected attributes are used to generate a TCP attack detection agent;

2) 21 extracted attributes are used to produce a UDP attack detection agent;

3) 18 chosen attributes are used to detect ICMP attacks.

B. The Results and Analysis

The candidates of the training and testing data are exhibited in Table I. Experimental results list in Table II.

TABLE I
DISTRIBUTION OF THE RECORDS

Protocol	TCP protocol		UDP protocol		ICMP protocol	
Data set	Test	Train	Test	Train	Test	Train
Total	4700	14 313	11 295	9047	5012	23 382
Normal	2561	5110	10 987	8182	84	50
DoS	1995	8939	255	723	4833	23 300
R2L	1	100	0	0	0	0
U2R	16	5	0	0	0	0
Probing	127	159	53	142	95	32

TABLE II
EXPERIMENTAL RESULTS

Protocol	TCP protocol		UDP protocol		ICMP protocol	
Training time	5.641s		0.771s		1.1130s	
Result	Correct records	Accuracy (%)	Correct records	Accuracy (%)	Correct records	Accuracy (%)
normal	2544	99.30	8809	80.18	83	98.81
DoS	1994	99.95	196	76.86	4823	99.79
R2L	1	100.00	0	0	0	0
U2R	12	75.00	0	0	0	0
Probing	120	94.49	35	66.04	84	88.42

From Tables I and II, we have the following conclusions:

1) The training time of UDP detector is the shortest, that of ICMP goes after, and that of TCP is the longest.

2) The number of detecting UDP attacks is the lowest, that of TCP goes after, and that of ICMP is the highest.

3) The detecting accuracy is directly related to the number of attack samples in the training data set.

4) 23 382 ICMP training samples contain 23 332 attack samples. The proportion of ICMP attack samples is 99.7862 %.

5) 14 313 TCP training samples contain 9203 attack samples. The proportion of TCP attack samples is 64.2982 %.

6) 9047 UDP training samples only have 865 attack samples. The proportion of UDP attack samples is 9.5612 %.

7) The detecting accuracy of new, unknown, and abnormal attacks is low. When the attack samples of the testing data set are greater than that of the training data set, the detecting rate is lower.

Here, we compare the detection accuracy of the collaborative and adaptive intrusion detection model (CAIDM) with that of a set of a single SVM detector (they are composed of a set of SVM detectors, every SVM detector solely detects suspicious data in different levels.) Their detection results are listed in Table III.

TABLE III
COMPARISON BETWEEN CAIDM AND SINGLE SVM DETECTOR

Algorithm	Count	Average-accuracy (%)	Error (%)	Training (s)
SingleType-SVM	17 166	81.716	19.39	29.730
CAIDM	18 701	89.02	12.19	7.247

The total records correctly detected by CAIDM are: 2544+1994+1+12+120+8809+196+35+83+4823+84=18 701.

The total test records are: 4700+11 295+5012 = 21 007 and the total detection accuracy is: 18 701/21 007=89.02 %.

The total records correctly detected by a set of a single SVM detector are 17 166; and the detection accuracy rate is: 17 166/21 007=81.716 %.

In addition, the training time of a set of a single SVM detector is about 29.730 s, while the training time of the CAIDM only requires 7.247 s. The above results reveal that the CAIDM based on 2-class SVMs and DTs is better than a set of a single SVM detector in terms of both the training time and the detection accuracy.

C. Live Detection

In a real network environment, detection data comes from current network packets. TCP/IP network packets can be decomposed into four parts: TCP, UDP, ICMP and application layer protocol. TCP means connection-oriented services, UDP means connectionless datagram services, and ICMP is the internet control message protocol. Three intrusion detectors are built according to TCP, UDP, and ICMP protocols. Furthermore, the content-based detector is built to cope with application layer protocols. A group of agents that play detector roles are created, and these data are detected by three intrusion detectors.

VIII. CONCLUSION

In this paper, a collaborative and adaptive intrusion detection method based on 2-class SVMs and DTs is proposed. A detection model called CAIDM is created and implemented. The E-CARGO model is used as a tool for describing the intrusion detection and modeling. In this paper, roles, groups, and agents are all studied and applied, for instance, the response unit role, the suspicious event detection role, the generating suspicious event role, etc. A role is assigned to some agents. A group (SmallGroup) contains many agents that perform the same role. TCP/IP protocols can be decomposed into four categories: TCP, UDP, ICMP, and application layer protocols. These protocols include different attributes. A variety of intrusion detectors are designed and implemented. There are four types of SVM identification functions designed and implemented, and related agents are created. These agents built by using different properties are applied to find out attacks for TCP, UDP, ICMP, and application layer protocols, respectively. The TCP detection agent is used as one example to explain the agent creation process. At last, the proposed method's effectiveness is confirmed by experiments with the KDD CUP 1999 data set. Experimental results reveal that the optimized collaborative and adaptive intrusion detection model (CAIDM) based on 2-class SVMs and DTs is more accurate and efficient than the detector system with a set of single type SVMs.

REFERENCES

[1] S. H. Teng, N. Q. Wu, W. Zhang, and X. F. Fu, "Cooperative intrusion detection based on object monitoring," *Acta Sci. Nat. Univ. Suny.*, vol. 47, no. 6, pp. 76−81, Nov. 2008.

[2] E. Alpaydin, *Introduction to Machine Learning*. 3rd ed. New York, NY, USA: The MIT Press, 2014.

[3] S. H. Teng, H. L. Du, N. Q. Wu, W. Zhang, and J. Y. Su, "A cooperative network intrusion detection based on fuzzy SVMs," *J. Netw.*, vol. 5 no. 4, pp. 475−483, Jan. 2010.

[4] F. J. Kuang, W. H. Xu, and S. Y. Zhang, "A novel hybrid KPCA and SVM with GA model for intrusion detection," *Appl. Soft Comput.*, vol. 18, pp. 178−184, May 2014.

[5] Y. H. Li, J. B. Xia, S. L. Zhang, J. K. Yan, X. C. Ai, and K. B. Dai, "An efficient intrusion detection system based on support vector machines and gradually feature removal method," *Expert Syst. Appl.*, vol. 39, no. 1, pp. 424−430, Jan. 2012.

[6] S. M. H. Bamakan, H. D. Wang, Y. J. Tian, and Y. Shi, "An effective intrusion detection framework based on MCLP/SVM optimized by time-varying chaos particle swarm optimization," *Neurocomputing*, vol. 199, pp. 90−102, Jul. 2016.

[7] A. A. Aburomman and M. B. Reaz, "A novel SVM-kNN-PSO ensemble method for intrusion detection system," *Appl. Soft Comput.*, vol. 38, pp. 360−372, Jan. 2016.

[8] S. W. Lin, K. C. Ying, C. Y. Lee, and Z. J. Lee, "An intelligent algorithm with feature selection and decision rules applied to anomaly intrusion detection," *Appl. Soft Comput.*, vol. 12, no. 10, pp. 3285−3290, Oct. 2012.

[9] W. Y. Feng, Q. L. Zhang, G. Z. Hu, and J. X. Huang, "Mining network data for intrusion detection through combining SVMs with ant colony networks," *Future Generation Comput. Syst.*, vol. 37, pp. 127−140, Jul. 2014.

[10] G. Kim, S. Lee, and S. Kim, "A novel hybrid intrusion detection method integrating anomaly detection with misuse detection," *Expert Syst. Appl.*, vol. 41, no. 4, pp. 1690−1700, Mar. 2014.

[11] S. J. Horng, M. Y. Su, Y. H. Chen, T. W. Kao, R. J. Chen, J. L. Lai, and C. D. Perkasa, "A novel intrusion detection system based on hierarchical clustering and support vector machines," *Expert Syst. Appl.*, vol. 38, no. 1, pp. 306−313, Jan. 2011.

[12] U. Ravale, N. Marathe, and P. Padiya, "Feature selection based hybrid anomaly intrusion detection system using K means and RBF kernel function," *Procedia Comput. Sci.*, vol. 45, pp. 428−435, Dec. 2015.

[13] W. C. Lin, S. W. Ke, and C. F. Tsai, "CANN: An intrusion detection system based on combining cluster centers and nearest neighbors," *Knowl.-Based Syst.*, vol. 78, pp. 13−21, Apr. 2015.

[14] A. Mitrokotsa, and C. Dimitrakakis, "Intrusion detection in MANET using classification algorithms: The effects of cost and model selection," *Ad Hoc Netw.*, vol. 11, no. 1, pp. 226−237, Jan. 2013.

[15] S. Pastrana, A. Mitrokotsa, A. Orfila, P. Peris-Lopez, "Evaluation of classification algorithms for intrusion detection in MANETs," *Knowl.-Based Syst.*, vol. 36, pp. 217−225, Dec. 2012.

[16] C. A. Catania, F. Bromberg, and C. G. Garino, "An autonomous labeling approach to support vector machines algorithms for network traffic anomaly detection," *Expert Syst. Appl.*, vol. 39, no. 2, pp. 1822−1829, Feb. 2012.

[17] S. H. Teng, C. Y. Zheng, H. B. Zhu, D. N. Liu, and W. Zhang, "A cooperative intrusion detection model for cloud computing networks," *Int. J. Sec. Appl.*, vol. 8 no. 3, pp. 107−118, May 2014.

[18] W. Zhang, S. H. Teng, H. B. Zhu, and D. N. Liu, "A cooperative intrusion detection model based on granular computing and agent technologies," *Int. J. Agent Technol. Syst.*, vol. 5, no. 3, pp. 54−74, Jul. 2013.

[19] W. Zhang, S. H. Teng, X. F. Fu, J. H. Fan, Y. Teng, and H. B. Zhu, "A cooperative intrusion detection model based on granular computing," in *Proc. 17th Int. Conf. Computer Supported Cooperative Work in Design*, Whistler, BC, Canada, 2013, pp. 325−331.

[20] H. B. Zhu and M. C. Zhou, "Efficient role transfer based on Kuhn-Munkres algorithm," *IEEE Trans. Syst. Man Cybern. A Syst. Hum.*, vol. 42 no. 2 pp. 491−496, Mar. 2012.

[21] H. B. Zhu, W. Zhang, Y. Wang, J. X. Zhu, D. N. Liu, and S. H. Teng, "A role-permission assignment method of RBAC involved conflicting constraints under E-CARGO," *Int. J. Cognit. Inf. Nat. Intell.*, vol. 9 no. 4, pp. 49−64, Oct. 2015.

[22] H. B. Zhu, D. N. Liu, S. Q. Zhang, Y. Zhu, L. Y. Teng, and S. H. Teng, "Solving the Many to Many assignment problem by improving the Kuhn-Munkres algorithm with backtracking," *Theor. Comput. Sci.*, vol. 618, no. C, pp. 30−41, Mar. 2016.

[23] X. J. Zhu. "Anomaly detection through statistics-based machine learning for computer networks," Ph. D. dissertation, Univ. Arizona, Arizona, USA, 2006.

Shaohua Teng (M'15) received the Ph. D. Degrees in industry engineering from Guangdong University of Technology, Guangzhou, China, in 2008. From 1982 to 1998, he was with the Jiangxi Normal University. Since 2005, he has been a Professor at Guangdong University of Technology.

He is a Senior Member of Chinese Association of Automation and China Computer Federation. His research interests include big data, data mining, network security, cooperative work, and Petri net theory and applications. He has applied for 14 patents on his invention. He has published 200 papers on computer magazines and international conferences and 2 books. He earned a Provincial Science and Technology Award, and Guangdong outstanding teacher.

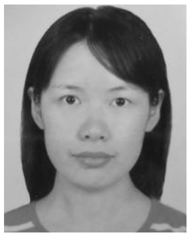
Naiqi Wu (M'04-SM'05) received the B. S. Degree in electrical engineering from Anhui University of Technology, Huainan, China, in 1982, the M.S. and Ph.D. Degrees in systems engineering both from Xi'an Jiaotong University, Xi'an, China in 1985 and 1988, respectively. From 1988 to 1995, he was with Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang, China, and from 1995 to 1998, with Shantou University, Shantou, China. He moved to Guangdong University of Technology, Guangzhou, China in 1998. He joined Macau University of Science and Technology in 2013. He is currently a Professor at the Institute of Systems Engineering, Macau University of Science and Technology. His research interests include production planning and scheduling, manufacturing system modeling and control, discrete event systems, Petri net theory and applications, intelligent transportation systems, information assurance, and energy systems. He is the author or coauthor of one book, five book chapters, and 130+ peer-reviewed journal papers. Dr. Wu was an Associate Editor of the *IEEE Transactions on Systems, Man, & Cybernetics, Part C, IEEE Transactions on Automation Science and Engineering, IEEE Transactions on Systems, Man, & Cybernetics: Systems*, and Editor in Chief of *Industrial Engineering Journal*. He is serving as an associate editor for Information Sciences and guest editor of *IEEE/CAA Journal of Automatica Sinica*.

Haibin Zhu (M'02-SM'04) received the B.S. degree in computer engineering from Institute of Engineering and Technology, Zhengzhou, China (1983), and M.S. (1988) and Ph.D. (1997) degrees in computer science from the National University of Defense Technology (NUDT), Changsha, China.

He is a Full Professor of the Department of Computer Science and Mathematics, Founder and Director of Collaborative Systems Laboratory, Nipissing University, Canada, and a Visiting Professor of the Department of Control Science and Engineering, Nanjing University, China. He was a Visiting Professor and a Special Lecturer in the College of Computing Sciences, New Jersey Institute of Technology, Newark, USA (1999−2002) and a Lecturer, an Associate Professor, and a Full Professor at NUDT (1988−2000). He has published 150+ research papers,4 books, and 4 book chapters on object-oriented programming, distributed systems, collaborative systems, and computer architecture.

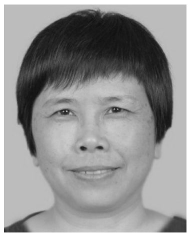
He is serving and served as co-chair of the technical committee of Distributed Intelligent Systems of IEEE SMC Society, Associate Editor of *IEEE SMC Magazine*, Associate Editor of *Int'l J. of Agent Technologies and Systems*, Associate Editor-in-Chief for *Int'l J. of Advances in Information and Service Sciences*, member of the editorial board of Int'l J. of Software Science and Computational Intelligence, Guest Editor for *IEEE Transactions on SMC(A)*, organizer of the workshops and special sessions on Role-Based Collaboration (RBC) for 10+ int'l conferences, and program committee member for more than 50 int'l conferences.

He is the recipient of the 2011 Chancellors' award for excellence in research and 2006−2007/2011−2012 research achievement awards from Nipissing University, the 2004 and 2005 IBM Eclipse Innovation Grant Awards, the Best Paper Award from the ISPE Int'l Conf. on Concurrent Engineering (ISPE/CE2004), the Educator's Fellowship of OOPSLA'03, a 2nd Class National Award of Education Achievement from Ministry of Education of China (1997), a 2nd Class National Award of Excellent Textbook from the Ministry of Education of China (2002), three 1st Class Ministerial Research Achievement Awards from The Commission of Science Technology and Industry for National Defense of China (1997, 1994, and 1991), and a 2nd Class Excellent Textbook Award of the Ministry of Electronics Industry of China (1996). He is a senior member of IEEE, a member of ACM, and a life member of the Chinese Association for Science and Technology, USA.



Luyao Teng received the Master degree in Melbourne University, Victoria, Australia, in 2014. Since 2015, she has been working for her Ph.D. degree in Victoria University, Victoria, Australia.

Her current research interests include data mining, data base, image recognition.



Wei Zhang received the M.S. Degree in software engineering from the South China University of Technology, Guangzhou, China, in 2005. Since 2006, she is an Associate Professor at Guangdong University of Technology, Guangzhou, China.

She is a Senior Member of China Computer Federation. She is responsible for teaching data mining in School of Computer Science and Technology. Her research interests include big data, network security, machine learning and statistical pattern recognition. She has published 150 papers on computer magazines and international conferences. She earned a Provincial Science and Technology Award.