Applied Soft Computing 10 (2010) 859–867

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

Sequential anomaly detection based on temporal-difference learning: Principles, models and case studies☆

Xin Xu

Institute of Automation, College of Mechatronics and Automation, National University of Defense Technology, Deya Road, Changsha 410073, PR China

ARTICLE INFO

Article history:
Received 3 January 2008
Received in revised form 11 May 2009
Accepted 3 October 2009
Available online 27 November 2009

Keywords:
Anomaly detection
Temporal-difference
Markov reward processes
Learning prediction
Computer security
Reinforcement learning

ABSTRACT

Anomaly detection is an important problem that has been popularly researched within diverse research areas and application domains. One of the open problems in anomaly detection is the modeling and prediction of complex sequential data, which consist of a series of temporally related behavior patterns. In this paper, a novel sequential anomaly detection method based on temporal-difference (TD) learning is proposed, where the anomaly detection problem of multi-stage cyber attacks is considered as an application case. A Markov reward process model is presented for the anomaly detection and alarming process of sequential data and it is verified that when the reward function is properly defined, the anomaly probabilities of sequential behaviors are equivalent to the value functions of the Markov reward process. Therefore, TD learning algorithms in the reinforcement learning literature can be used to efficiently construct anomaly detection models of complex sequential behaviors by estimating the value functions of the Markov reward process. Compared with other machine learning methods for anomaly detection, the proposed approach has the advantage of simplified labeling process using delayed evaluative signals and the prediction accuracy can be improved even if labeled training data are limited. Based on the experimental results on intrusion detection of host computers using system call data, it was shown that the proposed anomaly detection method can achieve higher or at least comparable detection accuracies than other approaches including SVMs, and HMMs.

1. Introduction

As a typical pattern recognition task, anomaly detection is to detect non-conforming or abnormal patterns from a given class of normal behaviors. These non-conforming patterns are often referred to as anomalies, outliers, exceptions, or surprises in different applications. Anomaly detection is widely used in a variety of domains, such as intrusion detection, fraud detection, fault detection, system health monitoring, and event detection in sensor networks. Although anomaly detection in data has been studied in the statistics community as early as the 19th century, there are still several open problems to be solved. As discussed in [1], one of the main challenges for anomaly detection techniques is that defining a normal region which encompasses every possible normal behavior is very difficult. The other challenge is that availability of labeled data for training/validation of models used by anomaly detection techniques is usually a major issue. In addition, the data usually contains noise which tends to be similar to the actual anomalies and hence is difficult to distinguish and remove. In recent years, aiming at the above challenges, a variety of anomaly detection techniques have been developed in the soft computing and machine learning communities. For a comprehensive survey on anomaly detection techniques based on machine learning, readers may refer to [1].

In general, existing soft computing approaches to anomaly detection can be grouped into three categories, i.e., supervised or classification-based, semi-supervised, and unsupervised anomaly detection methods. Supervised anomaly detection techniques learn a classifier using labeled instances belonging to normal or anomaly class, and then assign a normal or anomalous label to a test instance. Typical approach in such cases is to build a predictive model for normal vs. anomaly classes [2–4,28]. Semi-supervised anomaly detection techniques construct a model representing normal behavior from a given normal training data set, and then test the likelihood of a test instance to be generated by the learnt model. It is assumed that the training data have labeled instances for only the normal class. Since they do not require labels for the anomaly class, they are more widely applicable than supervised techniques [5,16]. Unsupervised anomaly detection techniques detect anomalies in an unlabeled test data set under the assumption that majority of the instances in the data set are

☆ This work is supported by the National Natural Science Foundation of China under Grant 60303012, 60774076, 90820302 the Fork Ying Tung Youth Teacher Foundation of China under Grant 114005, and the Natural Science Foundation of Hunan Province under Grant 07JJ3122.

E-mail addresses: xuxin_mail@263.net, xinxu@nudt.edu.cn.

1568-4946/$ – see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.asoc.2009.10.003

normal [10,11]. The techniques in this category make the implicit assumption that normal instances are far more frequent than anomalies in the test data. If this assumption is not true then such techniques suffer from high false alarm rate.

Although anomaly detection techniques have been widely studied and applied in a variety of areas, there are still many challenges for detecting anomalies in sequential data which is very common in a wide range of domains where a natural ordering is imposed on data instances by either time or position. In anomaly detection literature, two types of data sequences have been popularly studied, i.e., symbolic and continuous sequences. In this paper, we will mainly focus on symbolic data sequences but the methodology developed may also be extended to continuous data sequences or time series. Due to the temporally related nature of sequential data, detecting anomalous subsequences is more challenging than anomaly detection in static patterns. In this paper, a novel sequential anomaly detection method based on temporal-difference (TD) learning [19], which can be called TD_SAD, is presented, where intrusion detection of multi-stage computer attacks is considered as a special application case. For anomaly detection of multi-stage cyber attacks, Markovian modeling of sequences has been a popular approach in this category. However, in our approach, a new Markov reward model is established for sequential data, which is different from previous works in that reward functions are defined as a feedback signal to indicate whether a long sequence of observation patterns is normal or abnormal. Furthermore, it is analyzed in theory that the sequential anomaly detection task can be implemented by predicting the value functions of the Markov reward process. In the proposed TD_SAD approach, by incorporating reward signals for every observation pattern in data sequences, the anomaly probabilities of sequential behaviors are equivalent to the value functions of the Markov reward process. Therefore, the TD learning and prediction algorithms developed from the reinforcement learning [15] literature can be employed to detect multi-stage cyber attacks. According to the authors' knowledge, this is the first attempt to apply TD learning and prediction in sequential anomaly detection, which is different from previous supervised learning or statistical methods.

As we will analyzed in the following sections, the anomaly detection method proposed in this paper provides a new framework for detecting anomalies in multi-stage cyber attacks and can also be applied to other anomaly detection tasks in sequential data. The main contributions of this paper include the following two aspects. The first aspect of innovation is that reward functions are designed in Markovian modeling of sequential data so that the anomaly detection problem can be formulated as an equivalent value function prediction task. In previous works, the Markov models only focused on statistical models of state transitions and no reward functions were considered, which range from Finite State Automatons (FSAs) to Hidden Markov Models (HMMs) without reward functions. FSAs have been used to detect anomalies in network protocol data and operating system call intrusion detection [5], where anomalies are detected when a given sequence of events does not result in reaching one of the final states. In [12], HMM-based techniques were proposed to detect anomalous program traces in operating system call data. In our approach, the reward functions can be viewed as indicative signals for learning and the teacher signals in supervised learning are special cases of reward functions. So, the TD_SAD approach proposed in this paper provides a more flexible and efficient framework than FSA and HMMs and more prior information can be used to improve the performance of sequential anomaly detection. The second aspect of contributions is that temporal-difference learning was applied in anomaly detection of multi-stage cyber attacks and very promising results have been obtained. Until now, there have been few research works on applying TD learning in anomaly detection of complex sequential data. Thus, the proposed anomaly detection method based on Markov reward process and TD learning not only provides a new direction for research on intrusion detection using reinforcement learning but also has lots of potential extensions to anomaly detection tasks in other areas. The performance of the proposed anomaly detection method using TD learning was evaluated on system call data of host-based intrusion detection, from the MIT Lincoln Lab. and the University of New Mexico (UNM) [17]. The experimental results illustrate that the proposed method can achieve higher or at least comparable detection accuracies than previous approaches.

The paper is organized as follows. In Section 2, the research background of anomaly detection in computer security and related works are introduced. In Section 3, the anomaly detection problem in sequential data is formulated and analyzed by using intrusion detection of multi-stage cyber attacks as an application example. In Section 4, the anomaly detection method based on Markov reward models and TD learning is presented. It is proved that by appropriately selecting the reward functions of the Markov reward model, there is equivalence between the estimation of anomaly probabilities and the learning prediction of value functions. In Section 5, experimental results on the system call data from the MIT Lincoln Lab. and the UNM are described to illustrate the effectiveness of the proposed method. And in Section 6, conclusions and discussions are provided.

2. Background and related works

Since the detection problem of multi-stage cyber attacks will be used as an application case for the proposed anomaly detection method, some research background and related works will be introduced in the following. The purpose of intrusion detection [6] is to find cyber attacks or non-permitted deviations of the characteristic properties in a computer system or monitored networks. Earlier intrusion detection techniques commonly made use of extracted signatures of known attacks and made decisions by comparing observation data with the signatures. This kind of detection strategy is usually called misuse detection. Nevertheless, it is almost impossible for misuse detection systems to find new attacks with unknown or deformed signatures. To overcome the shortcomings of misuse detection, anomaly detection techniques in computer security have attracted lots of research interests in the literature [3,4]. Anomaly detection is different from misuse detection techniques in that little prior knowledge on precise signatures of computer attacks is needed. So, one advantage of anomaly detection is the ability to detect novel attacks. However, since conventional anomaly detection techniques have to deal with a complete set of normal behaviors, which usually have large uncertainties and observation noises, it is very difficult to for anomaly detection systems to have high detection rates and low false alarm rates simultaneously. In addition, in order to use training data from recorded attack behaviors to improve performance, it is also desirable to develop systematic methods to incorporate attack behaviors in the framework of anomaly detection, i.e., to construct hybrid anomaly detection models using both normal behavior data and attack data.

Aiming at the above problems, soft computing methods have been widely studied for anomaly detection in computer security applications in the past decade [7–11]. In [7,8], several efforts have been devoted to designing anomaly detection algorithms using supervised learning algorithms, such as neural networks, etc. Some recent works have been focused on using supervised learning methods to construct hybrid anomaly detection models, i.e., models that trained both on normal data and attack data, such as the multi-class classifier approach based on support vector machines (SVMs) [10]. Another approach to anomaly detection is to use unsupervised

learning methods [9]. Unlike supervised learning methods, whose models are built by careful labeling of observation data, unsupervised anomaly detection tries to detect anomalous behaviors with little *a priori* knowledge about the training data. However, as studied in [7], the performance of pure unsupervised anomaly detection approaches is usually unsatisfactory, e.g., it was demonstrated in [7] that supervised learning methods significantly outperformed the unsupervised ones if the test data only contain known attacks.

Despite of the great potentials of applying machine learning methods to improve the performance of anomaly detection, previous approaches still do not have satisfactory results in terms of detection accuracy and low false alarm rates, especially for complex multi-stage attacks, which consist of long sequences of observation features. Essentially, there are two important challenges need to be considered for anomaly detection of multi-stage cyber attacks. One challenge is the training data collection and labeling problem, which is crucial to the successful applications of supervised learning methods. Since the observation data in computer security are very huge and training data may be a mixture of normal usage and attack behaviors, it will be very expensive to get precisely labeled data. And it will be necessary to develop new hybrid anomaly detection models that can realize model training both on normal and abnormal data. The other challenge of applying machine learning methods in anomaly detection is to model dynamic sequential behaviors for complex multi-stage attacks, since many multi-stage attacks consist of sequences of temporally related observation features and it will be more difficult to give precise labels to such kind of attacks. Therefore, dynamic behavior modeling approaches [12] for anomaly detection become very important to improve the performance of intrusion detection in complex environments.

To simplify the process of data labeling and realize efficient modeling of sequential behavior patterns, semi-supervised learning methods have received increasing attention in recent years [14]. Unlike supervised and unsupervised learning algorithms, semi-supervised learning can make use of unlabeled data to solve complex sequential prediction and decision problems. In [14], a semi-supervised approach to anomaly detection was proposed, where a partially observable Markov decision process (POMDP) model was given as the decision model of intrusion detection problems. Nevertheless, the POMDP model is computationally expensive and it is very difficult to solve real-world POMDP problems with large state spaces. In [14], it was suggested that simpler models and cost functions may be used.

In order to overcome the weakness of previous anomaly detection approaches, especially to solve the data labeling and sequential behavior detection problem, in this paper, a novel anomaly detection method based on TD learning is proposed. After establishing a new Markov reward model for sequential anomaly detection, it is proved that with appropriately defined reward functions, the estimation of anomaly probabilities for sequential behaviors can be equivalently solved by learning the value functions of the Markov reward process. Therefore, TD learning algorithms [23,26] can be efficiently used for model construction and prediction in anomaly detection systems. The Markov reward model in this paper can be viewed as a practical extension and simplification of the POMDP model in [14]. More importantly, the main advantage of the proposed method is that it needs very little *a priori* knowledge on the precise labeling of training samples and only evaluative labeling on complete data sequences is required, which is more practical in real-world applications for detecting multi-stage cyber attacks.

3. The anomaly detection problem in sequential data

In the following, intrusion detection of multi-stage cyber attacks will be employed as an application case to formulate and analyze the sequential anomaly detection problem. Fig. 1 shows the basic structure of an intrusion detection system, which includes four main procedures, i.e., training data labeling, feature extraction, detection model training and online detection. In the data collection and labeling procedure, training data are selected and labeled from the observation data, which either come from host computers or network traffic. Then, a feature extraction process is performed to transform the data into representations that are suitable for model training. After that, detection models can be constructed and optimized by various machine learning algorithms. At last, online detection is performed and alarms can be sent out based on the outputs of the detection model.Among the above four procedures, training data labeling is one of the most important and difficult tasks since it is hard to extract signatures precisely even for known attacks and there are still increasing amounts of unknown multi-stage attacks with complex sequential signatures. Until now, although there have been many research works on intrusion detection based on machine learning methods, little attention has been put on the labeling problem. In most of the previous anomaly detection methods based on supervised learning algorithms, every single sample in the training data was either labeled as normal or abnormal. However, the distinctions between normal and abnormal behaviors are usually very vague and inappropriate labeling may limit or worsen the detection performance of supervised learning methods. This difficulty becomes more severe when the observed data consist of temporally related sequential patterns, i.e., the behaviors are described as temporal traces of single observation features. For example, in host-based intrusion detection, most user-to-root attacks are multi-stage attacks and they are consisted of sequences of system calls or shell commands [12]. The execution trajectories of different processes form different traces of system calls. Each trace is defined as the list of system calls issued by a single process from the beginning of its execution to the end. For intrusion detection using other types of observation data such as network connections, similar sequential behaviors can also be observed for complex multi-stage attacks. For example, Fig. 2 shows a sequential state transition model for a simplified IP protocol [5].

For intrusion detection based on various observation data, the observation elements are raw data from the sensors of an intrusion detection system (IDS), which can be formally defined as follows:

Definition 1 (Basic observation element).
A basic observation element in an IDS is an observation feature o_t that is obtained from a corresponding sensor at a given time.

From Definition 1, a basic observation element o_t can be either a system call in host-based IDSs or a connection feature vector in network-based IDSs. Let O denote the set of all possible symbols of observation elements o_t. A complete observation sequence for anomaly detection can be formally defined as follows.

Definition 2 (Complete observation sequence).
A complete observation sequence for intrusion detection is a time series of basic observation elements $\{o_1, o_2, \ldots, o_T\}$ with $o_i \in O$, which can be accurately determined either as normal or abnormal.

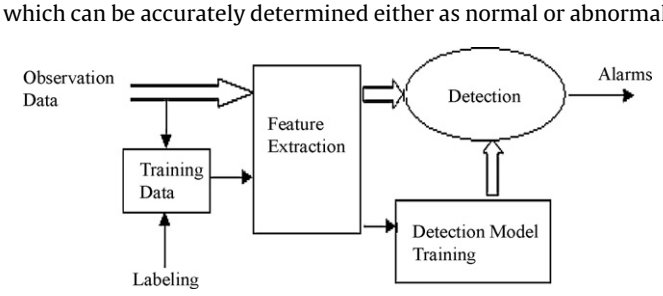

Fig. 1. Basic structure of an anomaly detection system.

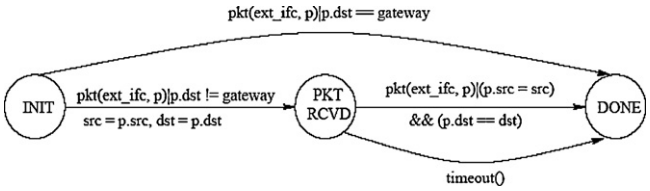

Fig. 2. A simplified IP state machine model [5].

For host-based intrusion detection, a complete observation sequence is typically a process in a host computer, but the meaning of a process, or trace, varies from program to program. For some programs, a process corresponds to a single task; for example, in *lpr*, a SunOS program, each print job generates a separate trace. In other programs, multiple processes are required to complete a task. A simple case of a process trace consisting of seven system calls is shown as follows:

open, *read*, *mmap*, *mmap*, *open*, *read*, *mmap*

According to Definition 2, a complete observation sequence is a series of observation elements, which has enough length to be specified either as a normal or abnormal behavior.

To detect sequential abnormal behaviors or attacks with temporally related patterns, state transition models can be used to distinguish normal sequences from abnormal sequences, where a state can be defined as a short sequence of observation elements. In the following, a formal description of the state in sequential data is given.

Definition 3 (State and state sequence for anomaly detection). A state $x_i = (o_{i+1}.o_{i+2}, \ldots, o_{i+n})$ in sequential anomaly detection is a short sequence or combination of several temporally successive observation elements, which is a part of a complete observation sequence. Based on the definition of states, a state sequence $S = \{x_1, x_2, \ldots, x_T\}$ can be obtained from the original complete observation sequence $\{o_1, o_2, \ldots, o_N\}$ by selecting a sliding window length l, i.e., $x_{i+1} = (o_{i+l+1}, o_{i+l+2}, \ldots, o_{i+l+n})$.

From the above definitions, it can be seen that the state sequences are transformed from the original observation sequences by appropriately selecting a sliding time window. For example, if we select a sequence of 4 system calls as one state and the sliding length between sequences is 1, the state transitions corresponding to the trace {*open*, *read*, *mmap*, *mmap*, *open*, *read*, *mmap*} are:

- State 1: *open*, *read*, *mmap*, *mmap*
- State 2: *read*, *mmap*, *mmap*, *open*
- State 3: *mmap*, *mmap*, *open*, *read*
- State 4: *mmap*, *open*, *read*, *mmap*

Based on the definition of states, the task of anomaly detection is to make decisions of whether a state sequence is normal or abnormal, which can be illustrated in Fig. 3. As we will explain in the sequel, one of the main benefits for anomaly detection using state transition models is the simplification of the labeling process before machine learning algorithms can be used for model training. In Fig. 4, there are four complete observation traces x_1–x_2–x_3–x_4–x_5, x_1–x_6–x_3–x_4–x_5, x_1–x_2–x_3–x_7–x_8 and x_1–x_6–x_3–x_7–x_8, where the former two traces are normal traces and the latter two are abnormal. x_i ($i = 1, 2, \ldots, 8$) are states at different time steps, which have been defined in the above. To determine the label of a complete observation trace is usually easy since much *a priori* information and knowledge can be obtained to distinguish a complete normal trace from a complete abnormal one. For example, in the training data, we usually know that whether there is an attack during the observation of a complete data sequence. Nevertheless, it is very difficult to determine whether a single state x_i ($i = 1, 2, \ldots, 8$) is normal or abnormal. In many supervised learning approaches to IDSs, the states in a normal trace are all assumed to be normal and those states in an abnormal trace are all labeled as abnormal. Although this kind of labeling is simple to be implemented, it cannot describe the temporal behaviors and sequential relationships among states and as illustrated in Fig. 4, some states such as x_1 and x_2, cannot be simply labeled as normal or abnormal.

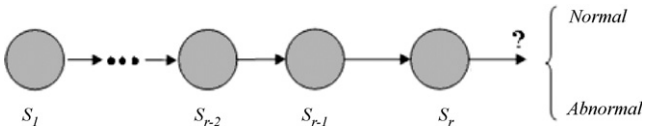

Fig. 3. The anomaly detection problem for state sequences.

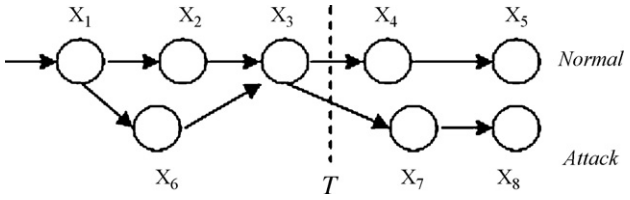

Fig. 4. The time to alarm (TTA) problem in sequential anomaly detection.

Moreover, to detect sequential patterns in multi-stage attacks more precisely, it is necessary to find a proper time point to determine whether there is an attack and raise alarms during the observation process. In Fig. 4, it can be seen that it is sufficient to raise alarms at the state transitions after state x_3 since it is obvious that the transition from x_3 to x_4 will lead to normal traces while the transition from x_3 to x_7 will lead to attacks. But at the states before x_3, it is not appropriate to determine whether a trace is normal or abnormal. Thus, there is a proper time to alarm for intrusion detection of sequential data, which has been rarely explored by previous supervised learning methods. In the following section, we will present an anomaly detection approach based on TD learning, which will provide an efficient framework to select an appropriate time to alarm during the detection of complex sequential behaviors.

4. Anomaly detection based on Markov reward model and TD learning

4.1. Markov reward model for anomaly detection in sequential data

Markov reward processes are popular stochastic models for sequential modeling and decision making. A Markov reward process can be denoted as a tuple $\{S, R, P\}$, where S is the state space, R is the reward function, P is the state transition probability. Let $\{x_t \mid t = 0, 1, 2, \ldots; x_t \in S\}$ denote a trajectory generated by the Markov reward process. For each state transition from x_t to x_{t+1}, a scalar reward r_t is defined. The state transition probabilities satisfy the following Markov property:

$$P\{x_{t+1}|x_t, x_{t-1}, \ldots, x_1, x_0\} = P\{x_{t+1}|x_t\} \quad (1)$$

For the anomaly detection problem discussed above, a Markov reward model will be established in the following. In the Markov reward model, the state x_i and the state sequence $S = \{x_1, x_2, \ldots, x_T\}$ are defined by Definition 3 and the state transition probability can be defined as follows.

Let $N(x_i)$ and $A(x_i)$ denote the sets of all possible normal and abnormal state sequences starting from state x_i, respectively. Let

$C(x_i)$ be the total number of state transitions that start from x_i. For any two states x_i and x_j, let $C(x_i, x_j)$ denote the total number of state sequences which start from x_i and have a state transition from x_i to x_j.

Definition 4 (State transition probability).
The state transition probability between two states x_i, x_j is defined as

$$P(x_i, x_j) = \frac{C(x_i, x_j)}{C(x_i)} \tag{2}$$

Then, the Markov reward model for anomaly detection of sequential behaviors can be formally described as follows:

Definition 5 (Markov reward model for anomaly detection).
A Markov reward model M of a complete observation sequence $S = \{x_1, x_2, \ldots, x_T\}$ is defined as a triple $\{X, R, P\}$, where X is the set of all possible states based on Definition 3, P is the state transition probability given by Definition 4, and the reward function R: $x \rightarrow r(x)$ is defined as

$$r(x) = \begin{cases} 0, & \text{if } x = x_T \text{ and } S \in N(x_1) \\ 1, & \text{if } x = x_T \text{ and } S \in A(x_1) \\ 0, & \text{if } x \neq x_T \end{cases} \tag{3}$$

For every state, a probability of anomaly $P_a(x)$ is defined as the probability of a complete observation sequence, which starts from x, being an abnormal sequence, i.e.

$$P_a(x) = P\{(x_1, x_2, \ldots, x_T) \in A(x) | x_1 = x\} \tag{4}$$

Then, for every state sequences $S = \{x_i\}$ ($i = 1, 2, \ldots, n$), an accumulated probability $P(S)$ of anomaly can be computed as

$$P(s) = \sum_{i=1}^{n} P_a(x_i) \tag{5}$$

Therefore, the anomaly detection problem can be solved by estimating the probability of anomaly for every state and comparing the accumulated anomaly probability $P(s)$ of a state sequence $S = \{x_i\}$ ($i = 1, 2, \ldots, n$) with a predefined threshold μ. If $P(s) > \mu$, then an alarm is generated to indicate the anomaly of the sequence.

Theorem 1 shows that based on the reward function defined in (3), there is equivalence between the estimation of anomaly probability of states and the value function prediction of the Markov reward model.

Theorem 1. *The state value function $V(x)$ of the Markov reward model M in Definition 5 is equal to the probability of state anomaly $P(x)$, i.e., $V(x) = P(x)$.*

Proof. The value function V(x) of a Markov reward process is given by

$$V(x) = E\{\sum_t \gamma^t r_t(x_t) | x_1 = x\} \tag{6}$$

□

In our discussion, due to the finite length of observation sequences, the discount factor γ is set to 1. By replacing the expectation $E\{.\}$ with weighted sum of probabilities, we get

$$V(x) = \sum_{i=1,N} P(x_{i1}, x_{i2}, \ldots, x_{iT(i)} | x_{i1} = x) \sum_{t=1,T(i)} r(x_t) \tag{7}$$

where $P(x_{i1}, x_{i2}, \ldots, x_{iT(i)} | x_{i1} = x)$ denotes the probability of the observation sequence $\{x_{i1}, x_{i2}, \ldots, x_{T(i)}\}$ that starts from x, N is the total number of possible observation sequences starting from x, and $T(i)$ is the length of the observation sequence.

Based on the definition of reward function in (3), the rewards are all zeros except for the terminal states, where the reward is either +1 or 0. Thus, the value function can be expressed as

$$V(x) = \sum_{i=1,N} P(x_{i1}, x_{i2}, \ldots, x_{iT(i)} | x_{i1} = x) r(x_{iT(i)}) \tag{8}$$

For all the possible observation sequences starting from x, they can be divided into two parts, i.e., abnormal and normal sequences. By the definition of $A(x)$ and $N(x)$, we get

$$\begin{aligned} V(x) &= \sum_{i \in A(x)} P(x_{i1}, x_{i2}, \ldots, x_{iT(i)} | x_{i1} \\ &= x) r(x_{iT(i)}) + \sum_{i \in N(x)} P(x_{i1}, x_{i2}, \ldots, x_{iT(i)} | x_{i1} = x) r(x_{iT(i)}) \\ &= \sum_{i \in A(x)} P(x_{i1}, x_{i2}, \ldots, x_{iT(i)} | x_{i1} = x) r(x_{iT(i)}) \end{aligned} \tag{9}$$

Since the anomaly probability of state x can be computed as

$$\begin{aligned} P(x) &= P\{(x_1, \ldots, x_T) \in A(x_1) | x_1 = x\} \\ &= \sum_{i \in A(x)} P\{(x_{i1}, x_{i2}, \ldots, x_{iT}) | x_1 = x\} \end{aligned} \tag{10}$$

According to (9) and (10), we can directly get

$$V(x) = P(x) \tag{11}$$

Thus, it is proved in Theorem 1 that the learning prediction of value functions for the Markov reward process is equivalent to estimate the anomaly probabilities of the corresponding states. More importantly, the result makes it possible to apply the TD learning and prediction methods from the RL literature to intrusion detection so that complex sequential behaviors can be detected efficiently.

4.2. The sequential anomaly detection algorithm using TD learning

Until now, temporal-difference learning [19] has been considered as one of the most efficient approaches to value function prediction without any *a priori* model information about Markov reward processes. Different from supervised learning for sequential prediction such as Monte Carlo estimation methods, TD learning is to update the estimations based on the differences between two temporally successive estimations, which constitutes the main ideas of a popular class of TD learning algorithms called TD(λ) [19]. The aim of TD(λ) is to estimate the value functions of a Markov reward process by observing the state transition sequences, where a reward signal is given after each state transition. The temporal difference is defined as the difference between two successive estimations and has the following form

$$\delta_t = r_t + \gamma \tilde{V}_t(x_{t+1}) - \tilde{V}_t(x_t) \tag{12}$$

where x_{t+1} is the successive state of x_t, $\tilde{V}(x)$ denotes the estimate of the value function $V(x)$ and r_t is the reward received after the state transition from x_t to x_{t+1}.

For Markov reward processes with large or continuous state spaces, linear function approximators are commonly used [22]. In linear TD(λ) algorithms, value functions are represented as

$$\tilde{V}(x) = \phi^T(x) W = \sum_{j=1}^{n} \phi_j(x) w_j \tag{13}$$

where $\phi(x) = [\phi_1(x), \phi_2(x), \ldots, \phi_n(x)]$ is a vector of linear basis functions and $W = [w_1, w_2, \ldots, w_n]$ is the weight vector.

In [22], linear TD(λ) algorithms are proved to converge with probability 1 under certain assumptions and the limit of convergence W^* is also derived, which satisfies the following equation.

$$E_0[A(X_t)] W^* - E_0[b(X_t)] = 0 \tag{14}$$

$$A(X_t) = \vec{z}_t(\phi^T(x_t) - \gamma\phi^T(x_{t+1})) \tag{15}$$

$$b(X_t) = \vec{z}_t r_t \tag{16}$$

$$\vec{z}_t = \gamma\lambda\vec{z}_{t-1} + \phi(x_t) \tag{17}$$

where $X_t = (x_t, x_{t+1}, z_{t+1})$ $(t = 1, 2, \ldots)$ form a new Markov process, x_t and x_{t+1} are two successive states, r_t is the corresponding reward, $E_0[\cdot]$ stands for the expectation with respect to the unique invariant distribution of $\{X_t\}$, λ is a constant for eligibility traces $z_t(x)$, and γ is the discount factor.

As studied in [23], LS-TD(λ) algorithms have better data efficiency than conventional linear TD(λ) algorithms. The least-squares TD(λ) algorithm computes the weight vector W by solving Eq. (14) directly, i.e.,

$$W_{\text{LS-TD}(\lambda)} = A_T^{-1}b_T = \left(\sum_{t=1}^{T} A(X_t)\right)^{-1}\left(\sum_{t=1}^{T} b(X_t)\right) \tag{18}$$

where T is the length of the state trajectories.

Based on the analysis in Theorem 1, the LS-TD(λ) algorithm can be applied to estimate the anomaly probabilities by predicting the value functions of the Markov reward process defined above. At first, the observation data for training need to be transformed to a state transition model, which has been discussed in Section 3. Then, the reward function defined in (3) is used to make the observation sequences become the state sequences of a Markov reward model. After preparing the training data, the following anomaly detection algorithm (Algorithm 1), called TD_SAD (TD-based Sequential Anomaly Detection), can be employed to construct the detection model by approximating the state value functions of the Markov reward process.

Algorithm 1. TD_SAD—The sequential anomaly detection algorithm based on TD learning

1: Given:
- State transition data $\{(x_t, x_{t+1}, r_t)\}$ $(t = 1, 2, \ldots, T)$ for training, where every state transition trace with length T was evaluated as normal or abnormal and the reward function was designed by (3).
- A termination criterion for the algorithm.
- The linear basis functions for LS-TD(λ), and the eligibility parameter λ.
- Sequential data $S_n = \{x_i'\}$ $(i = 1, 2, \ldots, n)$ for testing, and the threshold parameter μ.

2: Initialize:
(2.1) Let $t = 0$.
(2.2) Set the initial state x_0.
3: Training: loop for maximum iteration number n:
Loop for every state sequences:
(3.1) For the current state x_t,
- If x_t is an absorbing state, $r(x_t) = r_T$, where r_T is the terminal reward.
- Otherwise, observe the state transition from x_t to x_{t+1} and the reward $r(x_t, x_{t+1})$, use Eq. (17) to update z_t, and use (15), (16) to update $A(X_t)$, $B(X_t)$.

(3.2) If x_t is an absorbing state, i.e., the end of a state sequence, re-initialize the process by setting x_{t+1} to a starting state of a observation sequence.
(3.3) Whenever updated estimations are desired, use Eq. (18) to compute the coefficients and value function estimations.
(3.4) $t = t + 1$.
4: Output the detection model $\{W_{\text{LS-TD}}, \phi(x)\}$ for testing.
5: Testing: for every state x' in testing sequence, the anomaly probability can be estimated by

$$(19)\quad P_a(x') = \tilde{V}(x') = \varphi^T(x')W_{\text{LS-TD}} = \sum_{j=1}^{n}\varphi_j(x')w_j$$

6: Alarming: the accumulated anomaly probability of the testing sequence $S_n = \{x_i\}$ $(i = 1, 2, \ldots, n)$ can be computed by (5). The decision output of the anomaly detection system can be determined as follows:

If $P(S_n) > \mu$, then raise alarms,
else no alarms.

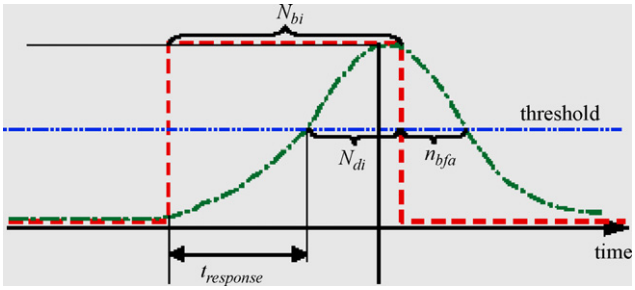

Fig. 5. The response curve and its relationship with performance measures of multi-stage attacks [29].

Based on the above detection strategy, the time to alarm as well as the detection accuracy of the sequential anomaly detection system is determined by the value function estimation of TD learning and the threshold parameter. This conclusion can be obtained by analyzing the response curve of detection outputs and its relationship with the performance measures of the detection system, which is illustrated in Fig. 5 [29]. In Fig. 5, the green curve is the curve of the detection outputs, e.g., the value function estimation of the proposed method and the red dotted line shows the time interval that a multi-stage attack occurs. The blue dotted line is the threshold and alarms are raised when the detection outputs are greater than the threshold. N_{bi} denotes the total number of observation states in the time interval of a multi-stage attack, n_{di} denotes the number of states that the attack is correctly detected by the detection system, and n_{bfa} is the number of states that false alarms occur. $T_{response}$ can be viewed as the time to alarm discussed in Section 3.

From Fig. 5, it can be seen that the detection accuracy of multi-stage anomalous behaviors can be guaranteed by regulating the output response of the detection model, i.e., to make the ratio n_{di}/N_{bi} be closer to 1 and make n_{bfa}/N_{bi} be closer to zero. In our case, the output response is completely determined by the value function prediction model of the Markov reward process. Thus, there are two direct ways to improve the performance of the proposed approach. One way is to increase the estimation accuracy of the TD learning prediction algorithm. For example, it may be beneficial to use TD(λ) with nonlinear approximation abilities such as the kernel LS-TD(λ) algorithm in [26]. However, in our following experiments, linear LS-TD(λ) has satisfactory performance when compared with previous approaches. Another way to improve the detection performance is to select the optimal threshold parameter by a threshold determination procedure, which will be used in the experimental section of this paper.

4.3. Analysis and discussions

The sequential anomaly detection method based on TD learning transforms the estimation of anomaly probability of states to the learning prediction of state value functions of a Markov reward process. Compared with previous efforts on anomaly detection based on machine learning, the approach proposed in this paper has advantages in the following aspects. Firstly, it does not need precise labeling of training data and only evaluative reward signals on complete observation sequences are needed. The TD learning algorithms developed in the RL literature can be applied so that the sequential nature of complex behaviors can be modeled in an efficient way. Secondly, the computational complexity of TD learning algorithms is linear with respect to the number k of state features and the length m of traces, i.e., it has time complexity of $O(km)$, which is much lower than other statistical modeling approaches based on Markov models [12,18]. For example, the training algorithm for HMMs is very expensive, i.e., it runs in time

$O(nm^2)$, where n is the number of states in the HMM and m is the size of the trace. When TD learning prediction methods using function approximators are used, the number k of state features can become much smaller than n. Moreover, compared with other statistical modeling approaches such as HMMs and general Markov chain models [13], our method only implicitly constructs the probabilistic model and the detection of anomalies is based on the estimation of value functions, which has been proved to be equivalent to the estimation of anomaly probabilities. In [13], the robustness of Markov chain modeling techniques was studied and it was shown that when explicitly estimating the probabilistic structure of the Markov chain model for normal data, the detection accuracy was very sensitive to the noise of data, i.e., when the intrusion data were mixed with normal data, the performance of the Markov chain model would become worse. Nevertheless, in our approach, the detection accuracy is not influenced by the mixing of normal and abnormal data due to the hybrid modeling strategy.

The semi-supervised learning and reinforcement learning methods for intrusion detection [14,16] are closely related to the research work in this paper. However, there are some major differences among these approaches. In the semi-supervised model based on POMDP [14], both actions and rewards are considered, where the actions correspond to whether to raise alarms or not and the rewards reflect the costs of certain security policies, e.g., time to alarm,—the elapsed time between the beginning of a real attack and its first detection, etc. Moreover, the estimation of a POMDP model is computationally expensive. Furthermore, another major difference between the proposed Markov reward model and the semi-supervised POMDP model in [14] is that in our model, no action policies are considered and the anomaly detection problem is modeled as an equivalent learning prediction task for an appropriately defined Markov reward process. This can greatly reduce the computational complexity and realize the main goal of anomaly detection in an efficient way. An earlier work on reinforcement learning methods for intrusion detection was presented in [16], where a CMAC neural network was used to detect denial-of-service (DOS) attacks based on the feedback signals of protected systems. The results showed that reinforcement learning was suitable to realize autonomous intrusion detection systems without much work on labeling of training samples. However, the method in [16] only used very simple update rules of reinforcement learning, which lack rigorous theoretical analysis on convergence and generalization ability, and the application was limited to a particular type of Denial-of-Service (DOS) attacks.

5. Experimental results

To evaluate the effectiveness of the proposed approach, experiments on anomaly detection for host computers using system call data were conducted. In the experiments, a wide variety of data sets were used, which include "live" normal data, i.e., traces of programs collected during normal usage of a production computer system, and different kinds of multi-stage cyber attacks including buffer overflows, symbolic link attacks, Trojan programs, etc. Table 1 shows some of the details of the data, where four kinds of attack data as well as normal data are considered. The data used in the experiments were typical data of sequential behaviors collected from real or simulated environments and they have been widely studied by other researchers to evaluate the performance of intelligent modeling and prediction methods for intrusion detection [20,21]. All of these data sets are publicly available at the website of Department of Computer Science, the University of New Mexico [17].

In the data sets, each trace is a sequence of system calls generated by a single process from the beginning of its execution to the end. Since the traces were generated by different programs under different environments, the number of system calls per trace varies widely. In the experiments, four different classes of system call traces were used, which correspond to four types of intrusive program behaviors, i.e., *MIT live lpr*, *sendmail*, *ps*, and *login*. Here, "*live*" is defined to be traces of programs collected during normal usage of a production computer system, and "*synthetic*" is defined to be traces collected by running a prepared script, i.e., the program options were chosen solely for the purpose of exercising the program, and not to meet any real user's requests. For detailed discussion of the properties of the data sets, please refer to [20].

As shown in Table 1, the four types of system call traces were divided into two parts. One part is for model training and threshold determination and the other part is for performance evaluation. Table 1 shows the numbers of normal and attack traces for training and testing. As can be seen in the table, the numbers of testing traces are usually larger than those of training traces. During the threshold determination process, the same data sets are used as the training process, i.e., the training data sets and the data sets for threshold determination are the same. In the testing stage, two criteria for performance evaluation are used, which include the detection rate *Dr* and the false alarm or false positive rate *Fp*, and they are computed as follows:

$$Dr = \frac{n_d}{n_a} \tag{20}$$

$$Fp = \frac{N_a}{N} \tag{21}$$

where n_d is the number of abnormal traces that have been correctly identified by the detection model and n_a is the total number of abnormal traces, N_a is the number of normal states that have been incorrectly identified as anomaly by the detection model, and N is the total number of normal states. In the computation of false alarm rates, all false alarms during a long state trace are all counted and the total sum of false alarms is divided by the number of all states in the traces.

For the four types of program data, i.e., *MIT-lpr*, *ps*, *login*, and *sendmail*, four different anomaly detection models were separately trained using the TD-based sequential anomaly detection (TD_SAD) method (Algorithm 1). Every state in the Markov reward model has a sequence length of 6, which has been optimally selected using information-theoretic measures in previous work [27]. The reward function is defined by (3). To compare the performance of the TD_SAD method developed in this paper with previous approaches, the experimental results in [12,25], where

Table 1
System call data for performance evaluation.

		lpr	*sendmail*	*ps*	*login*
Training and Threshold Selection	Normal Trace Number	10	13	12	10
	Attack Trace Number	20	5	3	1
Testing	Normal Trace Number	2,703	67	11	11
	Attack Trace Number	1,001	7	8	3
Total system call number		1,023,950	223,733	8333	10,948

Table 2
Performance comparisons between TD and HMM methods.

	TD_SAD		HMM [12]	
	Detection rate	False alarm rate	Detection rate	False alarm rate
lpr	100%	0.00749	100%	3E−4
sendmail	100%	0.002951	61.5% 84.6% 92.3%	0.05[a] 0.10[a] 0.20[a]
ps	100%	0.003815	100%	–[b]
login	100%	0	77.8%	–[b]

[a] The false alarm rates were computed based on the percentages of incorrectly identified normal traces.
[b] No normal traces were used as testing data.

Table 3
Performance comparisons between TD and supervised pattern classification methods.

	TD_SAD	SVM	Naïve Bayes	C4.5	RIPPER	Logistic regression
live lpr MIT detection rate false alarm rate	100% 0.75%	99.80% 0.14%	100% 62.31%	99.90% 0.11%	99.80% 0.11%	99.90% 0%
sendmail detection rate false alarm rate	100% 0.29%	40% 0.28%	92% 84.97%	40% 1.15%	48.00% 2.31%	64% 2.31%

HMMs and other supervised learning methods were applied to the same data sets, are also shown in the following tables.

In Table 2, the detection rates and false alarm rates of TD_SAD and HMMs are compared with respect to the four types of program traces. It is illustrated that for all the data sets, TD_SAD has a detection rate of 100% with very low false alarm rates. While HMMs have lower detection rates and the corresponding false alarm rates are usually higher than TD_SAD. In addition, as indicated in the previous section, HMMs have larger computational complexity than the proposed TD_SAD method. In Table 2, the false alarm rates of HMMs in the data sets of *ps*, and *login* were not computed since no normal data were used in the testing stage of [12].

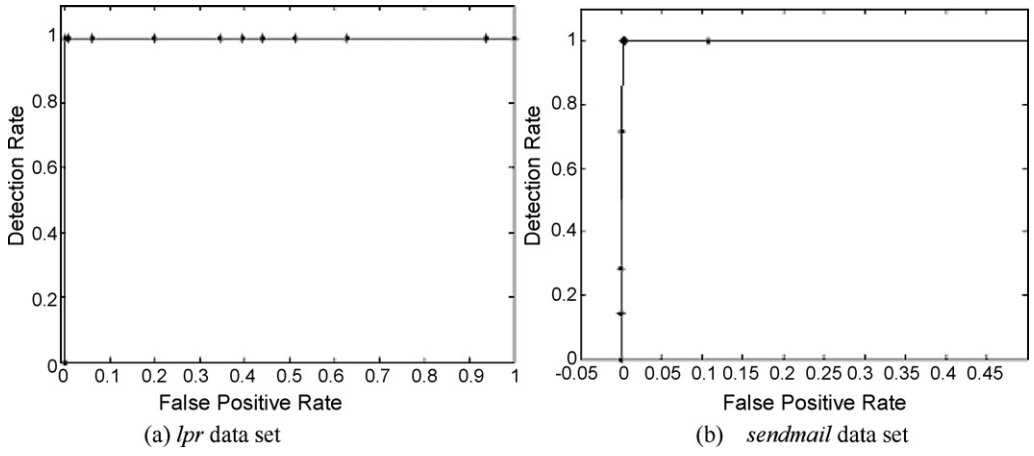

Fig. 6. ROC curves of TD_SAD for the *lpr* data set (a) and the *sendmail* data set (b).

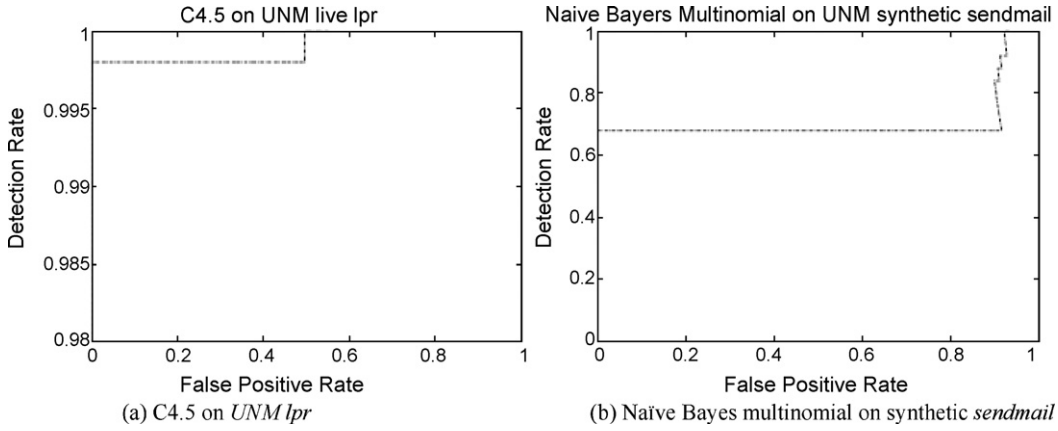

Fig. 7. ROC curves of C4.5 and Naïve Bayes on the *lpr* data (a) and the *sendmail* data (b) [25].

By making use of the data sets of two program behaviors, i.e., *MIT live lpr* and *sendmail*, Table 3 makes comparisons between the TD_SAD method and some popular supervised pattern classification methods including support vector machines (SVMs), Naïve Bayes methods, C4.5 decision trees, RIPPER and logistic regression, etc. [25]. In [25], a 'bag of system calls' representation for intrusion detection of system call sequences was proposed so that the intrusion detection problem was solved as a static pattern classification task and various supervised learning methods were employed.

From Table 3, it is also shown that TD_SAD can usually obtain higher detection rates than other supervised learning methods and the false alarm rates of TD_SAD are relatively low.

Fig. 6 depicts the ROC curves [24] obtained from the performance evaluation of TD_SAD on the testing data of *MIT live lpr* and *sendmail*, where different thresholds were selected and the corresponding detection rates and false alarm rates were computed. In Fig. 7, the ROC curves of C4.5 and Naïve Bayes on the *lpr* data and the *sendmail* data are plotted, which were obtained in the research work in [25]. From Figs. 6 and 7, it is shown that compared with other supervised learning methods such as C4.5 and Naïve Bayes, the TD_SAD model can obtain very low false alarm rates while having detection rates as high as 100%.

6. Conclusions

To overcome the weakness of previous anomaly detection approaches, especially to solve the data labeling and sequential behavior prediction problem, this paper suggests the TD_SAD algorithm, a sequential anomaly detection method based on temporal-difference (TD) learning, where intrusion detection of multi-stage cyber attacks is studied as an application case. In the proposed approach, a novel Markov reward model is established for anomaly detection of data sequences and it is proved that under certain assumptions, the learning prediction of value functions for the Markov reward process is equivalent to estimate the anomaly probabilities of the data sequences. Therefore, TD learning algorithms from the RL literature can be used to construct detection models and improve the performance of sequential anomaly detection only by simplified labeling schemes using evaluative signals or feedbacks. From the experiments in anomaly detection of multi-stage cyber attacks, it is illustrated that compared with previous anomaly detection approaches using machine learning, the TD_SAD method can obtain comparable or even better detection accuracies for complex sequential attacks. More importantly, the proposed approach provides a new RL-based anomaly detection technique with a simplified labeling procedure and reduced computational complexity for sequential data. Future work may need to be focused on the extension of the proposed method to more general anomaly detection problems.

Acknowledgments

The author would like to thank the anonymous reviewers for their valuable comments and suggestions, which greatly improved the quality of this paper.

References

[1] V. Chandola, Arindam Banerjee, Vipin Kumar, Anomaly detection: a survey, ACM Computing Surveys (2009) 1–72.
[2] M.V. Joshi, R.C. Agarwal, V. Kumar, Predicting rare classes: can boosting make any weak learner strong? in: Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, New York, NY, USA, (2002), pp. 297–306.
[3] N.V. Chawla, N. Japkowicz, A. Kotcz, Editorial: special issue on learning from imbalanced data sets, SIGKDD Explorations 6 (1) (2004) 1–6.
[4] I. Steinwart, D. Hush, C. Scovel, A classification framework for anomaly detection, Journal of Machine Learning Research 6 (2005) 211–232.
[5] R. Sekar, A. Gupta, J. Frullo, T. Shanbhag, A. Tiwari, H. Yang, S. Zhou, Specification-based anomaly detection: a new approach for detecting network intrusions, in: Proceedings of the 9th ACM Conference on Computer and Communications Security, ACM Press, 2002, pp. 265–274.
[6] W. Lee, S.J. Stolfo, K.W. Mok, Adaptive intrusion detection: a data mining approach, Artificial Intelligence Review 14 (6) (2000) 533–567.
[7] P. Laskov, P. Düssel, C. Schäfer, K. Rieck, Learning intrusion detection: supervised or unsupervised? Proc. ICIAP 2005, September. Lecture Notes in Computer Science, LNCS 3617 (2005) 50–57.
[8] M. Mahoney, P. Chan:, Learning nonstationary models of normal network traffic for detecting novel attacks, in: Proceedings of 8th International Conference on Knowledge Discovery and Data Mining, 2002, pp. 376–385.
[9] H. Shah, J. Undercoffer, A. Joshi:, Fuzzy clustering for intrusion detection, in: Proceedings of the 12th IEEE International Conference on Fuzzy Systems, 2003, pp. 1274–1278.
[10] X. Xu, X.N. Wang, Adaptive network intrusion detection method based on PCA and support vector machines, ADMA 2005, Lecture Notes in Artificial Intelligence, LNAI 3584 (2005) 696–703.
[11] S. Jha, K. Tan, R. Maxion, Markov chains, classifiers, and intrusion detection, in: Proceedings of the Computer Security Foundations Workshop (CSFW), June, 2001.
[12] D.Y. Yeung, Y.X. Ding, Host-based intrusion detection using dynamic and static behavioral models, Pattern Recognition 36 (2003) 229–243.
[13] N. Ye, Y. Zhang, C.M. Borror, Robustness of the Markov–Chain model for cyber-attack detection, IEEE Transactions on Reliability 53 (1) (2004) 116–123.
[14] T. Lane, A decision-theoretic, semi-supervised model for intrusion detection, in: Machine Learning & Data Mining for Computer Security: Methods & Applications, Springer, 2006, pp. 157–178.
[15] L.P. Kaelbling, M.L. Littman, A.W. Moore, Reinforcement learning: a survey, Journal of Artificial Intelligence Research 4 (1996) 237–285.
[16] J. Cannady, Next generation intrusion detection: autonomous reinforcement learning of network attacks, in: 23th National Information Systems Security Conference, 2000.
[17] http://www.cs.unm.edu/~immsec/data-sets.htm.
[18] D. Ourston, S. Matzner, et al., Applications of hidden Markov models to detecting multi-stage network attacks, in: Proc. of the 36th Hawaii International Conference on System Science, 2002, 334–343.
[19] R. Sutton, Learning to predict by the method of temporal differences, Machine Learning 3 (1) (1988) 9–44.
[20] S. Hofmeyr, et al., Intrusion detection using sequences of systems call, Journal of Computer Security 6 (1998) 151–180.
[21] Y.H. Liao, V.R. Vemuri, Using text categorization techniques for intrusion detection, in: Proceedings of the 11th USENIX Security Symposium, August, (2002), pp. 51–59.
[22] J.N. Tsitsiklis, B.V. Roy, An analysis of temporal difference learning with function approximation, IEEE Transactions on Automatic Control 42 (5) (1997) 674–690.
[23] J.A. Boyan, Technical update: least-squares temporal difference learning, Machine Learning 49 (2002) 233–246.
[24] J.P. Egan, Signal Detection Theory and ROC Analysis, Academic Press, New York, 1975 (Series in Cognition and Perception).
[25] D.K. Kang, D. Fuller, V. Honavar, Learning classifiers for misuse detection using a bag of system calls representation, in: P. Kantor et al. (Eds.), ISI 2005, Lecture Notes in Computer Science, 3495 (2005) 511–516.
[26] X. Xu, T. Tao, X.C. Lu, Kernel least-squares temporal difference learning, International Journal of Information Technology 11 (9) (2005) 54–63.
[27] W. Lee, D. Xiang, Information-theoretic measures for anomaly detection, IEEE Symposium on Security and Privacy (2001).
[28] A. Tajbakhsh, Mohammad Rahmati, and Abdolreza Mirzaei, intrusion detection using fuzzy association rules, Applied Soft Computing 9 (2) (2009) 462–469.
[29] A. Lazarevic, L. Ertoz, V. Kumar, A. Ozgur, J. Srivastava, A comparative study of anomaly detection schemes in network intrusion detection, Proceedings of the Third SIAM International Conference on Data Mining, (2003), pp. 25–36.